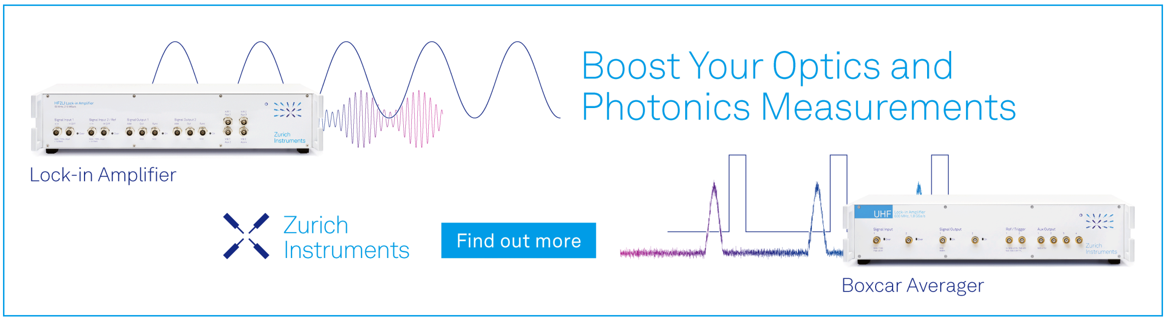
RESEARCH ARTICLE | OCTOBER 22 2018

Predicting the Jakarta composite index price using ANFIS and classifying prediction result based on relative error by fuzzy Kernel C-Means FREE

F. Fanita; Z. Rustam

AIP Conf. Proc. 2023, 020206 (2018)
https://doi.org/10.1063/1.5064203



Predicting the Jakarta Composite Index Price Using ANFIS and Classifying Prediction Result Based on Relative Error by Fuzzy Kernel C-Means

F. Fanita and Z. Rustam [a)]

Department of Mathematics, Faculty of Mathematics and Natural Sciences (FMIPA), Universitas Indonesia, Depok 16424, Indonesia

[a)] Corresponding author: rustam@ui.ac.id

Abstract. Stock index reflects the price movement a group of stock. There are many stock indices in the world. JKSE (Jakarta Composite Index) is one of the stock indices in IDX (Indonesia Stock Exchange). There are many benefits in knowing the movement of JKSE value, one of them is to reduce the risk in investing in the stock market. Therefore it is a need to predict JKSE value. The method that is used in this paper is ANFIS (Adaptive Neuro Fuzzy Inference System). ANFIS is a hybrid model which can give the better accuracy than other isolated technique of AI (Artificial intelligence) such as ANN, fuzzy logic, and GA. This paper gives two outputs, they are the prediction result and classification result based on some relative error values. Classification method that is used in this paper is Fuzzy Kernel C-Means. This two outputs will help the investors in decision making. The experimental results give an average of prediction accuracy 91 % and classification accuracy 80.2 %.

Keywords: predict, classify, Jakarta Composite Index, ANFIS, Fuzzy Kernel C-Means, Relative Error

INTRODUCTION

The stock is a type of investment that is quite popular in this era. The stock offers high profit, but the risk is also high. One way that can be used to minimize the loss when investing in the stock market is to follow the development of the stock price index value. Stock price index reflects the price movement a group of stock. By looking at the stock price index, it can be seen the trend in the market that is rising, falling, or stable. The trend information can be used by the stock investor to take the decision in buying, selling, or holding.

There are many stock indices in the world. JKSE (Jakarta Composite Index) is one of 11 stock indices in IDX (Indonesia Stock Exchange). JKSE is usually used as a benchmark to value the preformance of investor's stock portofolio. If the increment of JKSE value is higher than the investor's stock portofolio, it means that the investor's stock portofolio does not work well.

JKSE is usually called as the leading economic indicator. Economic indicator is the data that is used to know how the country economic condition. There are many economic indicators such as stock index, inflation, export, import, the number of unemployment, etc. JKSE has ability in describing the economic condition in the future preceding another economic indicators, as written in kompas news by A. H. Manurung.

Because of the benefits in following JKSE value, we want to predict the JKSE value here by applying machine learning concept. In this paper, ANFIS (Adaptive Neuro Fuzzy Inference System) is used to predict JKSE value. ANFIS is a hybrid model which can give the better accuracy than other isolated technique of AI (Artificial intelligence) such as ANN (Artificial Neural Network), fuzzy logic, and GA (Genetic Algorithm) [1]. The previous research that is written by Setianingrum and Rustam [2], ANFIS is used to predict the stock price by using the technical indicator. In this paper, we will predict the stock index JKSE using ANFIS not by using technical indicator

Proceedings of the 3rd International Symposium on Current Progress in Mathematics and Sciences 2017 (ISCPMS2017)
AIP Conf. Proc. 2023, 020206-1–020206-6; https://doi.org/10.1063/1.5064203
Published by AIP Publishing. 978-0-7354-1741-0/$30.00

but using the information the JKSE value one or a few days before. The data that is used in this paper is the daily adjusted close value of JKSE in 2013, 2014, 2015, and 2016. The dataset JKSE is taken from the yahoo finance.

After the ANFIS model is formed from training data, we can do the prediction. Because we have the real target value on testing data so we can calculate the relative error in every datum of training data. Then, we can classify the relative error of every datum of training data under some values, this paper using 1 %, 2 %, and 3 %. Classification method that is used in this paper is Fuzzy Kernel C-Means (FKCM). FKCM overcomes the FCM's drawback that is sensitive to the noise or outlier.

Using Matlab program to make the model, this paper gives two outputs, they are the prediction result and classification result based on some relative error values. This two outputs will help the investors in decision making. The rest of the paper provides the literatur survey, the experimental result and following by the conclusion.

LITERATUR SURVEY

Adaptive Neuro Fuzzy Inference System

Adaptive Neuro Fuzzy Inference System (ANFIS) [3] is equivalent to FIS (Fuzzy Inference Systems) functionally and belongs to a class of adaptive networks. In order to ease in explaining ANFIS, it is assumed that Fuzzy Inference System has only two input x and y. ANFIS has only one output f_{out}. In the first order Sugeno fuzzy model, two "if-then" rules are as follows.

Rule 1 : if x is A_1 and y is B_1 then $f_1 = p_1 x + q_1 y + r_1$

Rule 2 : if x is A_2 and y is B_2 then $f_1 = p_2 x + q_2 y + r_2$

p_1, p_2, q_1, q_2, r_1 dan r_2 are the parameters. There are five layers in ANFIS architechture and the nodes of the same layer have the similar function. We denote the output of the i^{th} node of layer l as $O_{l,i}$. The ANFIS architechture is shown in Fig. 1.

The node function in this layer is

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x) \quad for \;\; i = 1,2, \;\; or \\ O_{1,i} &= \mu_{B_{i-2}}(y) \;\; for \;\; i = 3,4 \end{aligned} \tag{1}$$

x(or y) is the inputs in node i and A_i(or B_{i-2}) is a linguistic labels for inputs. $O_{1,i}$ is a membership grade of A_i(or B_{i-2}). In general, membership function for $\mu_{A_i}(x)$ and $\mu_{B_{i-2}}(y)$ is generalized bell function :

$$\mu_{A_i}(x), \mu_{B_{i-2}}(y) = \frac{1}{1 + \left| \dfrac{x - c_i}{a_i} \right|^{2b_i}} \tag{2}$$

$\{a_i, b_i, c_i\}$ is parameter set of membership function.

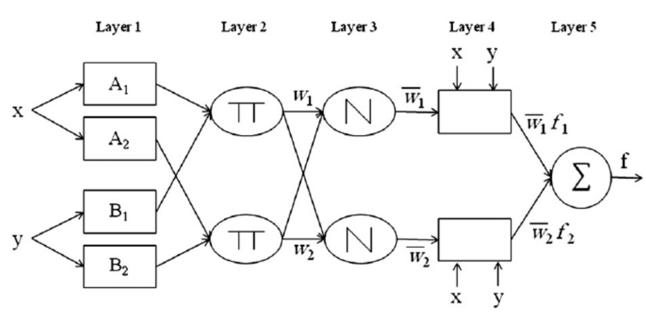


Figure 1. ANFIS architecture

Layer 2

The output of every node in this layer is the product of all the incoming signals. Each node output represent firing strength of the rules

$$O_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_{i-2}}(y), \quad i = 1,2 \tag{3}$$

Layer 3

The i^{th} node represents the ratio of the i^{th} rule's firing strength to the sum of all rule's firing strength.

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \tag{4}$$

Layer 4

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i x + r_i), \quad i = 1,2 \tag{5}$$

$\overline{w_i}$ is the output of layer 3.

Layer 5

The single node in this layer computes the overall output as the summation of all incoming signals.

$$O_{5,i} = \sum_i \overline{w_i}.f_i = \frac{\sum_i w_i.f_i}{\sum_i w_i} = f_{out} \tag{6}$$

Fuzzy C-Means

Fuzzy C-Means (FCM) [4] is one kind of many clustering methods that is found by Bezdek. For a set of data $X = \{x_1, x_2, \ldots, x_m\} \subseteq \Re^d$, we define $n \times c$ Membership Matrix $U = [u_{ij}],\ 1 \le i \le n,\ a \le j \le c,$ and Cluster Center $V = \{v_1, v_2, \ldots, v_c\}$. Each object in V is belong to d-dimensional Euclidean Space.

The objective in Fuzzy C-Means model is given by :

$$J(U,V) = \min \sum_{i=1}^{n} \sum_{j=1}^{c} (u_{ij})^m d^2(x_i, v_j) \tag{7}$$

with constraints:

$$\begin{aligned} &\sum_{j=1}^{c} u_{ij} = 1, \quad i = 1,2,\ldots,n \\ &\sum_{j=1}^{n} u_{ij} > 0, \quad i = 1,2,\ldots,c \\ &u_{ij} \in [0,1], \quad j = 1,2,\ldots,c \end{aligned} \tag{8}$$

In the model above d is dissimilarity or distance function and we called $m \in [1, \infty)$ as the degree of fuzziness for cluster partition.

We update the cluster center and membership values using :

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}, \quad j = 1,2,\ldots,c \tag{9}$$

$$u_{ij} = \left(\sum_{j=1}^{c} \left(\frac{d(x_i, v_i^t)}{d(x_i, v_j^t)} \right)^{\frac{2}{m-1}} \right)^{-1}, \quad 1 \le i \le n \tag{10}$$

Fuzzy Kernel C-Means

Fuzzy Kernel C-Means (FKCM) [4] is an algorithm that comes from FCM. FKCM modifies FCM by adding the kernel information. FKCM can handle the outlier, so its cluster center measurement is better than FCM.

FCM method is not quite good to classify non-linearly separable data. To recover this drawback, the set of data are transformed into the space which a higher dimention. We call it as the feature space. The transformed data behavior are expected close to linearly separable data, the classification accuracy can be better. While working in the high dimensional space, the cost is expensive (memory use and computational time). We can use kernel so we do not need to work directly at the feature space.

Set a nonlinear mapping φ from input data space into feature space F. The clustering process is at F other than $\Re^d$. Set φ as a nonlinear mapping from space of input data into space of feature F and x, y are objects at data space without knowing explicit form of φ. A way to measure distance between transformed data $\varphi(x)$ and $\varphi(y)$ is needed to be found. We use kernel function K to solve this problem. The Kernel function that is usually used are polynomial and RBF (Radial Basis Function), they are:

- Polynomial [5]

  $K(x, x') = \left(x^T x' + c\right)^M$, c is constant, $c > 0$, M is a parameter

- RBF [5]

  $K(x, x') = \exp\left\{ -\frac{\|x - x'\|^2}{2\sigma^2} \right\}$, σ is a parameter

The distance between $\varphi(x)$ and $\varphi(y)$ that use kernel function is measured by :

$$d^2(\varphi(x), \varphi(y)) = \|\varphi(x) = \varphi(y)\|^2 = \varphi(x)^t \varphi(x) - 2\varphi(x)^t \varphi(y) + \varphi(y)^t \varphi(y) = K(x,x) - 2K(x,y) + K(y,y)$$

The FKCM algorithm is showed in figure below.

Input : $X, c, m_i, m_f, \varepsilon, T$

Output : U and V

1. Initial condition: $V^0 = [v_1, v_2, \ldots, v_c]$, $v_j \in C_j$
2. For $t = 1$ to T
3. $m = m_i + \frac{t(m_j - m_i)}{T}$
4. $b = -\frac{1}{m-1}$
5. Calculate membership

   $U^1 = [u_{ij}]$, $1 \le i \le n$, $1 \le j \le c$, by using

   $u_{ij} = \frac{d^b(x_i, v_j)}{\sum_{k=1}^{c} d^b(x_i, v_k)}$, $1 \le i \le n$, $1 \le j \le c$,
6. Update cluster center

   $V^t = [v_1, v_2, \ldots, v_c]$, where $v_j = \frac{\sum_{i=1}^{n} u_{ij} x_i}{\sum_{i=1}^{n} u_{ij}^m}$,

   $j = 1, 2, \ldots, c$
7. If $E = \sum_{j=1}^{c} k^2(v_{jt} - v_{jt-1}) \le \varepsilon$,

   STOP
8. $t = t + 1$

Figure 2. FKCM algorithm [4]

EXPERIMENTAL RESULTS

ANFIS method belong to the supervised learning. The training data used to make model is equipped with the data target. The illustration of the training data is shown in Table 1. We want to predict JKSE value for tomorrow $(t+1)$ by knowing the JKSE value a few days before $(\dots, t-2, t-1, t)$. We call the JKSE value that will be predicted in $(t+1)$ as a target and JKSE value few days before $(\dots, t-2, t-1, t)$ as features. To get the optimal result, we can compare combination of using 1 untill 6 features and using 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 % training data.

In order to know how good the model formed in predicting JKSE value, we evaluate it by calculating the accuracy on testing data. The accuracy is calculated by knowing the persentage of relative error on each datum of the prediction results. The persentage of relative error is measured by $\frac{|p - p^*|}{|p|} \times 100\%$ as in [6] where p is the actual value and p^* is the predicted value. If the value of relative error below 2 %, we label it as T and others as F. After that, we count how many T, devided it by the total data testing, and multiplying it with 100 %. We can also label T or F for the prediction result that has the relative error under 1 % and 3 % (just to compare). The Table 2 below represents the maximum accuracy of model that the result of prediction has the relative error below 1 %, 2 %, and 3 % from comparing the combination of using 1 untill 6 features and the persentage of training data 10 % untill 90 %.

The optimal model is formed using 1 feature. It means that it is better to predict JKSE value tomorrow with knowing JKSE value today (one day before). The maximum accuracy 77.31 % (for the prediction result below 1 %), 96.90 % (for the prediction result below 2 %), and 98.80 % (for the prediction result below 1 %) is obtained if we use each 90 %, 90 %, 40 % training data.

Because of we have labeled every datum when we calculate the accuracy of prediction result, we can do classification using FKCM. This classification is good to know whether the result of prediction using ANFIS is below the relative error that we want or not. Here the relative error that we want is below 1 %, 2 % or 3 %.

In order to see how the training data and kernel parameters affect the accuracy, we can compare using 10 %, 20 %, untill 90 % training data with the fixed parameter kernel and using a certain parameter with fixed training data (shown in Table 3 and Table 4). The set of changing parameter of polinomial kernel used is {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and RBF kernel is {0.0001, 0.001, 0.05, 0.1, 1, 5, 10, 50, 100, 1000}.

TABLE 1. The Illustration of Training Data

...	t-2	t-1	t	t+1 (target)

TABLE 2. The maximum accuracy of prediction result using ANFIS that has the relative error below 1 %, 2 %, 3 % from comparing the combination of using 1 untill 6 features and the persentage of training data 10 % untill 90 %

	1 %	**2 %**	**3 %**
The Maximum Accuracy	77.31 %	96.90 %	98.80 %
The Number of Feature	1	1	1
The Persentage of Training Data	90 %	90 %	40 %

TABLE 3. The maximum accuracy of classifying the prediction result that has relative error under 1 %, 2 %, and 3 % using FKCM with polinomial kernel, using fixed kernel parameter with changing training data and fixed training data with changing kernel parameter. The fixed training data used is 10% and the fixed kernel parameter used is 2.

	1 %	**2 %**	**3 %**
Using Fixed Training Data (10 %)	83.72 %	55.17 %	73.56 %
Parameter Kernel	1, 2, 3	2, 3, 4, 5, 6, 7, 8, 9, 10	All parameter
Using Fixed Parameter Kernel (2)	83.72 %	83.33 %	73.56 %
Training Data	10 %	90 %	10 %

TABLE 4. The maximum accuracy of classifying the prediction result that has relative error under 1%, 2%, and 3% using FKCM with RBF kernel, using fixed kernel parameter with changing training data and fixed training data with changing kernel parameter. The fixed training data used is 10% and the fixed kernel parameter used is 0,001.

	1 %	2 %	3 %
Using Fixed Training Data (10 %)	83.72 %	55.17 %	73.56 %
Parameter Kernel	All parameter	0.0001, 0.001, 1, 5, 100	All parameter
Using Fixed Parameter Kernel (0.001)	83.72 %	83.33 %	73.56 %
Training Data	10 %	90 %	10 %

From Table 3, we know that we can get the maximum accuracy 83,72% (for classifying the prediction result that has relative error below 1%), 83,33% (for classifying the prediction result that has relative error below 2%), and 73,56% (for classifying the prediction result that has relative error below 3%) if we use the fixed polinomial kernel parameter 2 and each use 10%, 90%, and 10% training data.

The maximum accuracy in Table 4 is the same as Table 3 although the kernel function used is different. We can get the maximum accuracy 83.72 % (for classifying the prediction result that each has relative error below 1 %), 83.33 % (for classifying the prediction result that each has relative error below 2 %), and 73.56 % (for classifying the prediction result that each has relative error below 3 %) if we use the fixed RBF kernel parameter 0.001 and use each 10 %, 90 %, and 10 % training data.

CONCLUSIONS

We can predict JKSE value with ANFIS method using information of JKSE value one day before which give the optimal accuracy 77.31 % for the prediction result that has relative error below 1 % with using 90 % data training, 96.90 % for the prediction result that has relative error below 2 % with using 90 % data training, and 98.80 % for the prediction result that has relative error below 3 % with using 40 % data training.

After predicting JKSE value, we can do classification of prediction result that has relative error below 1 %, 2 %, and 3 %. We can get the optimal accuracy 83.72 % for classification of prediction result that has relative error below 1 % using 90 % training data, accuracy 83.33 % for classification of prediction result that has relative error below 2 % using 90 % training data, and accuracy 73.56 % for classification of prediction result that has relative error below 3 % using 40 % training data. The kernel that gives that optimal result is polynomial whose parameter is 2 or RBF whose parameter is 0.001.

So far, people are just focused on how to predict the value. In this paper, after we predict the value, we can check whether the prediction result has the relative error under 1 %, 2 %, or 3 %. By this result, it is expected that the investor can make a right decision and avoid losses.

ACKNOWLEDGMENTS

This research was funded by Hibah PITTA (Publikasi Internasional Terindeks untuk Tugas Akhir Mahasiswa) 2017 Universitas Indonesia with contract number 706/UN2.R3.1/HKP.05.00/2017.

REFERENCES

1. G. Kaur, J. Dhar, and R. K. Guha, Mathematics and Computers in Simulation **122**, 69 (2015).
2. A. Setianingrum and Z. Rustam, in *International Symposium on Current Progress in Mathematics and Sciences*, Depok, 2016 (FMIPA UI, Depok, 2016).
3. J. S. R. Jang, C. T. Sun, and E. Mizutani, ANFIS: Adaptive Neuro-Fuzzy Inference Systems, in *Neuro-Fuzzy and Soft Computing*, edited by Tom Robbins (Prentice-Hall Inc., Upper Saddle River, 1997), pp. 335-337.
4. Z. Rustam and A. S. Talita, Journal of Theoretical and Applied Information Technology (JATIT) **81**, 161 (2015).
5. C. M. Bishop, Kernel Methods, in Pattern Recognition and Machine Learning, edited by M. Jordan (Springer, Singapore, 2006), pp. 296.
6. R. L. Burden and J. D. Faires, Mathematical Preliminaries and Error Analysis, in *Numerical Analysis*, edited by M. Julet (Brooks/Cole, Boston, 2011), pp. 20.