

Forecasting Vietnamese stock index: A comparison of hierarchical ANFIS and LSTM

Quang Hung Do^{a*} and Tran Van Trang^b

^aFaculty of Information Technology, University of Transport Technology, Vietnam

^bFaculty of Business Administration, Thuongmai University, Hanoi, Vietnam

CHRONICLE

Article history:

Received October 10, 2019

Received in revised format:

October 28, 2019

Accepted November 30, 2019

Available online

November 30, 2019

Keywords:

Vietnamese stock index

Forecasting

Adaptive network based fuzzy

inference system (ANFIS)

Long short-term memory (LSTM)

ABSTRACT

Forecasting stock index has been received great interest because an accurate prediction of stock index may yield benefits and profits for investors, economists and practitioners. The objective of this study is to develop two efficient forecasting models and compare their performances in one day-ahead forecasting the daily Vietnamese stock index. The model development used the data across 9 years of the trading days. The developed models are based on two artificial intelligence techniques, including adaptive network based fuzzy inference system (ANFIS) and long short-term memory (LSTM). The performance indexes including RMSE, MAPE, MAE and R were used to make comparison of the models. The experimental results reveal that both models successfully forecasted the daily Vietnamese stock index with a high accuracy rate. The comparative results of the two models were then discussed and analyzed. It was found that the LSTM model outperformed the hierarchical ANFIS model in forecasting stock index of the Vietnamese stock market.

© 2020 by the authors; licensee Growing Science, Canada.

1. Introduction

Stock market has been considered as an indicator of the economy. Considerable amount of studies have shown that the stock market development is positively related with the level of economic development both in short run and long run (Ake, 2010; Rahman & Salahuddin, 2009; Touny, 2012). Stock price index which is a statistical measure reflects the level and changing situation of various stock prices in stock market. Forecasting stock index has been getting more attention from practitioners and academia since it provides information for national macroscopic decision and affects brokers' investment strategy (Jian & Song, 2016). However, accurate prediction of the trends of the stock price index has long been regarded as one of the most challenging tasks since the stock market is complex, complicated, dynamic and chaotic (Kara et al., 2011).

The techniques used in forecasting stock price index can be grouped into two broad categories including statistical-based models and artificial intelligence-based models. Several popular methods in the statistical-based models are Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and Seasonal Autoregressive Integrated Moving Average (SARIMA). Naturally, the financial time series (i.e., stock index and exchange rate) include both linear and nonlinear patterns. One of the main limitations of the statistical-based models is that they can only effectively solve linear problems. Many studies concluded that the artificial

* Corresponding author.

E-mail address: hungdq@utt.edu.vn (Q.H. Do)

intelligence-based models outperform the statistical-based models in forecasting stock price index. Kyungjoo et al. (2007) applied SARIMA and artificial neural network (ANN) to forecast Korean Stock Index and then stated that the performance of ANN is more reliable. It was also shown that the combination of ANN and ARIMA gives much better results than those of ARIMA and ANN models in forecasting Vietnam index (Bao & My, 2019). Şenol and Özturan (2008) applied several models in prediction of the direction of the stock market index in Turkey, confirming that ANN-based model is one of the most robust methods for forecasting. In order to predict the direction of next day's Nikkei 225 index movement, Qui and Song (2016) applied ANN and genetic algorithm in constructing the forecasting model. The results indicated that their proposed method was more effective and obtained a high accuracy. Liu et. al. (2016) applied four supervised learning models, including Logistic Regression, Gaussian Discriminant Analysis (GDA), Naive Bayes and Support Vector Machine (SVM) to the prediction of S&P 500 index. It was found out that all the models could provide predictability to a certain degree. Besides, among developed models, the SVM model with a Radial Basis Function (RBF) kernel can achieve the highest accuracy rate. The above mentioned studies reveal the great success of artificial intelligence-based models in forecasting stock index.

The adaptive network based fuzzy inference system (ANFIS), a hybrid intelligent system, is a combination of ANNs and fuzzy systems; therefore, it has the advantages of both techniques (Azadeh et al., 2011; Buragohain & Mahanta, 2008; Metin Ertunc & Hosoz, 2008). Fuzzy systems are appropriate if sufficient expert knowledge about the process is available, while neural networks are useful if sufficient process data is available or measurable. The advantage of neural networks and fuzzy systems can be integrated in a neuron-fuzzy approach. Fundamentally, a neuron-fuzzy system is a fuzzy network that has its function as a fuzzy inference system. The system can overcome some limitations of neural networks, as well as the limits of fuzzy systems (Nauck et al., 1997; Singh et al., 2005), when it has the capacity to represent knowledge in an interpretable manner and the ability to learn. The details of the neuron-fuzzy system were proposed by Takagi and Hayashi (1991). Among the neuron-fuzzy systems, ANFIS, introduced by Jang (1993), is the most common tools. In the FIS, the fuzzy if-then rules are determined by experts, whereas in the ANFIS, it automatically produces adequate rules with respect to input and output data, and facilitates the learning capabilities of neural networks. ANFIS is suitable for the forecast of chaotic time series in financial markets. Esfahanipour and Mardani (2011) showed the superiority of the ANFIS model against ANN model and ANFIS model in forecasting Tehran Stock Exchange Price Index (TEPIX) using the time series data from 25 March 2001 until 25 September 2010. Boyacioglu and Avci (2010) developed a model based on ANFIS to predict the return on stock price index of the Istanbul Stock Exchange (ISE). The findings revealed that the model successfully forecasts the monthly return of ISE National 100 Index with an accuracy rate of 98.3%. Then, it can be concluded that ANFIS provides a promising alternative for stock market prediction.

Currently, recurrent neural network (RNN), a deep learning model, has been considered as one of the most affective models in processing sequential data. Long short-term memory (LSTM) is the most successful RNN architectures. LSTM has a unit of computation, a memory cell, that replaces traditional artificial neurons in the hidden layer of the neural network. Thus, LSTMs can grasp the structure of data dynamically over time with high prediction capacity. Jeenanunta et. al. (2018) used a RNN with LSTM to investigate the prediction of daily stock prices of the top five companies in the Thai SET50 index. In comparison with different techniques, the result showed that LSTM give the best performance with three index with less than 2% error. These studies showed that that deep learning models based on LSTM for forecasting stock price time series data are consistent.

The objective of this study is to investigate the effectiveness of LSTM networks, and hierarchical ANFIS in forecasting stock market price movements. Additionally, Vietnam stock market is cautious and unpredictable. It is absolutely necessary to develop an efficient forecasting model for stock price index. Therefore, the main contributions of this study are the followings: (1) evaluating two forecasting models based on LSTM and hierarchical ANFIS by comparing and analyzing through several

performance indexes; (2) developing the model using real data from Vietnam stock exchange market; and (3) proposing a suitable prediction model for Vietnam stock market.

2. Hierarchical adaptive neuro-fuzzy inference system (HANFIS)

Artificial Neural Network (ANN) and fuzzy theory, which are soft computing techniques, are used in establishing intelligent systems. A fuzzy inference system (FIS) employs fuzzy if-then rules when acquiring knowledge from human experts to deal with imprecise and uncertain problems (Yusof et al., 2012). FISs have been widely used to solve different classification problems (Fakhrhmad et al., 2012). However, fuzzy systems cannot learn from or adjust themselves (Ata & Kocyigit, 2010). An ANN has the capacity to learn from its environment, self-organize, and adapt in an interactive way. For these reasons, a neuron-fuzzy system, which is the combination of a fuzzy inference system and neuron network, has been introduced to produce a complete fuzzy rule base system. The advantage of neural networks and fuzzy systems can be integrated in a neuron-fuzzy approach. Fundamentally, a neuron-fuzzy system is a fuzzy network that has its function as a fuzzy inference system. The system can overcome some limitations of neural networks, as well as the limits of fuzzy systems (Nauck et al., 1997; Singh et al., 2005), when it has the capacity to represent knowledge in an interpretable manner and the ability to learn. The details of the neuron-fuzzy system were proposed by Takagi and Hayashi (1991). Among the neuron-fuzzy systems, ANFIS, introduced by Jang (1993), has been one of the most common tools. In the FIS, the fuzzy if-then rules are determined by experts, whereas in the ANFIS, it automatically produces adequate rules with respect to input and output data, and facilitates the learning capabilities of neural networks.

An ANFIS is a multilayer feed-forward neural network, which employs neural network learning algorithms and fuzzy reasoning to map from input space to output space. The architecture of ANFIS includes five layers, namely the fuzzification layer, the rule layer, the normalization layer, the defuzzification layer, and a single summation node. To present the ANFIS architecture and simplify the explanations, assume that the FIS has two inputs, x_1 and x_2 , two rules, and one output, y , as shown in Fig. 1. Each node within the same layer performs the same function. The circles are used to indicate fixed nodes while the squares are used to denote adaptive nodes. A FIS has two inputs and two fuzzy if-then rules that can be expressed as

$$\begin{aligned} \text{Rule 1: If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } y_1 &= p_1x_1 + q_1x_2 + r_1, \\ \text{Rule 2: If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } y_2 &= p_2x_1 + q_2x_2 + r_2, \end{aligned} \quad (1)$$

where x_1 and x_2 are the inputs; A_1, A_2, B_1, B_2 are the linguistic labels; p_i, q_i , and r_i , ($i=1$ or 2) are the consequent parameters (Jang, 1993) that are identified in the training process; and y_1 and y_2 are the outputs within the fuzzy region. Eq. (1) represents the first type of fuzzy if-then rules, in which the output part is linear. The output part can also be constants (Sugeno, 1985), represented as:

$$\begin{aligned} \text{Rule 1: If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } y_1 &= C_1 \\ \text{Rule 2: If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } y_2 &= C_2 \end{aligned} \quad (2)$$

where C_i ($i=1$ or 2) are constant values. Eq. (2) represents the second type of fuzzy if-then rules. For complicated problems, the first type of if-then rules is widely utilized to model the relationships of inputs and outputs (Wei et al., 2007). In this research, we also used a linear function for the output.

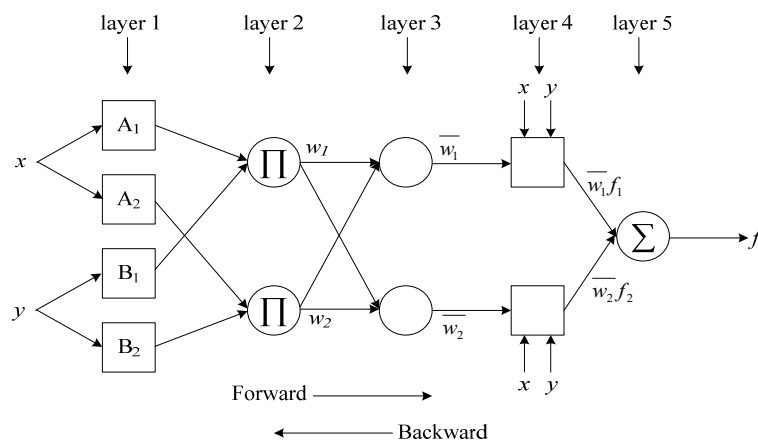


Fig. 1. An ANFIS architecture of two inputs and two rules

The brief description of the functions of each layer is as follows:

Layer 1 - fuzzification layer: Each node in this layer is a square node. The nodes produce the membership values. Outputs obtained from these nodes are calculated as follows:

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x_1) \text{ for } i=1, 2 \text{ or} & (3) \\ O_{1,i} &= \mu_{B_{i-2}}(x_2) \text{ for } i=3, 4, \end{aligned}$$

where $O_{1,i}$ denotes the output of node i in layer 1, and $\mu_{A_i}(x_1)$ and $\mu_{B_{i-2}}(x_2)$ are the fuzzy membership functions of A_i and B_{i-2} . The fuzzy membership functions can be in any form, such as triangular, trapezoidal, or Gaussian functions.

Layer 2 - rule layer: Every node in this layer is a circular node. The output is the product of all incoming inputs.

$$O_{2,i} = w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2) \text{ for } i=1, 2, \quad (4)$$

where $O_{2,i}$ denotes the output of node i in layer 2, and w_i represents the firing strength of a rule.

Layer 3 - normalization: Every node in this layer is a circular node. Outputs of this layer are called normalized firing strengths. The i th node is calculated by the i th node firing strength to the sum of all rules' firing strengths.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \text{ for } i=1, 2, \quad (5)$$

where $O_{3,i}$ denotes the output of node i in layer 3, and \bar{w}_i is the normalized firing strength.

Layer 4 - defuzzification layer: Every node in this layer is an adaptive node with a node function.

$$O_{4,i} = \bar{w}_i y_i \text{ for } i=1, 2, \quad (6)$$

where $O_{4,i}$ denotes the output of node i in layer 4, \bar{w}_i is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are consequent parameters of the Sugeno fuzzy model.

Layer 5 - a single summation node: The node is a fixed node. This node computes the overall output by summing all the incoming signals from the previous layer:

$$O_{5,i} = \sum_i \bar{w}_i y_i = \frac{\sum_i w_i y_i}{\sum_i w_i} \text{ for } i=1, 2, \quad (7)$$

where $O_{5,i}$ denotes the output of node i in layer 5. The results are then defuzzified using a weighted average procedure.

It can be seen that the ANFIS architecture has two adaptive layers: layer 1 and layer 4. Layer 1 has parameters related to the fuzzy membership functions and layer 4 has parameters $\{p_i, q_i, r_i\}$ related to the polynomial. The aim of the hybrid learning algorithm in the ANFIS architecture is to adjust all these parameters in order to make the output match the training data. Adjusting the parameters includes two steps. In the forward pass of the learning algorithm, the premise parameters are fixed, functional signals go forward until layer 4, and the consequent parameters are identified by the least squares method to minimize the measured error. In the backward pass, the consequent parameters are fixed, the error signals go backward, and the premise parameters are updated by the gradient descent method (Jang et al., 1997). This hybrid learning algorithm is able to decrease the complexity of the algorithm and increase learning efficiency (Singh et al., 2005). Due to this advantage of the hybrid learning algorithm, it was utilized in this study.

According to Güneri et al. (2011), too many inputs in the ANFIS structure makes the system complicated and limits its applicability. In addition, many studies pointed out that ANFIS gives better solutions with a simple structure. To deal with this issue, several low-dimensional rule bases should be arranged in a hierarchical structure (Brown et al., 1995). To model a hierarchical ANFIS, it is necessary to identify: a suitable hierarchical structure, the number of inputs for each sub-ANFIS model, and a rule base for each sub-ANFIS model.

When identifying the rule base for ANFIS, the problems under consideration are: (1) there are no standard methods for transforming human knowledge or experience into the rule base; and (2) it is necessary to tune the membership functions to maximize the performance and minimize the errors (Jang, 1993). There are several methods to identify FIS. In this paper, the grid partition method was utilized. This method divides the data space into rectangular sub-spaces using an axis-parallel partition based on the number of membership functions and their types in each dimension. An example of a grid partition with two input variables and two membership functions for each input variable is illustrated in Fig. 2. The combination of a grid partition and ANFIS has been mentioned by Kennedy et al. (2003). The grid partition is suitable for problems with a small number of inputs (Wei et al., 2007).

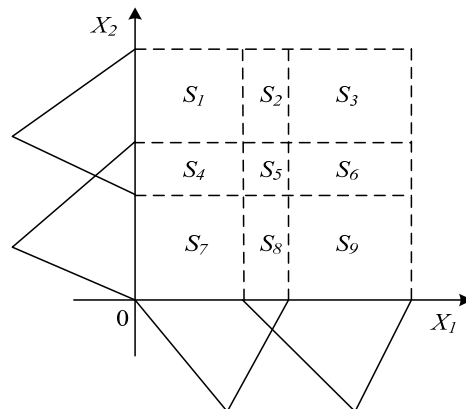


Fig. 2. An example of grid partition method with two input variables

3. Long short-term memory (LSTM)

An ANN is a mathematical model to simulate the network of biological neurons that mimic a human brain so that the computer would be able to learn things and make decisions in a humanlike manner. A deep neural network (DNN) is an ANN with more than the three layers. With more hidden layers, DNNs have the ability to capture highly abstracted feature from training dataset. Fig. 3 shows a deep neural network with three hidden layers. In comparison with conventional shallow learning architectures, DNN has capability to model deep complex non-linear relationship by using distributed and hierarchical feature representation. Various deep learning architectures such as convolution neural network (CNN), recurrent neural network (RNN) have been applied to the domain of computer vision, speech recognition, and natural language processing. RNN is an artificial neural network which solved the problem of traditional neural network. It is powerful to handle sequential data. As shown in Figure 4a, RNNs are networks with inner loops at the hidden layers, allowing information to persist (Schmidhuber, 2015). In a traditional ANN, it is assumed that all inputs (and outputs) are independent of each other. Whereas, RNNs perform the same task for every element of a sequence, with the output being depended on the previous computations.

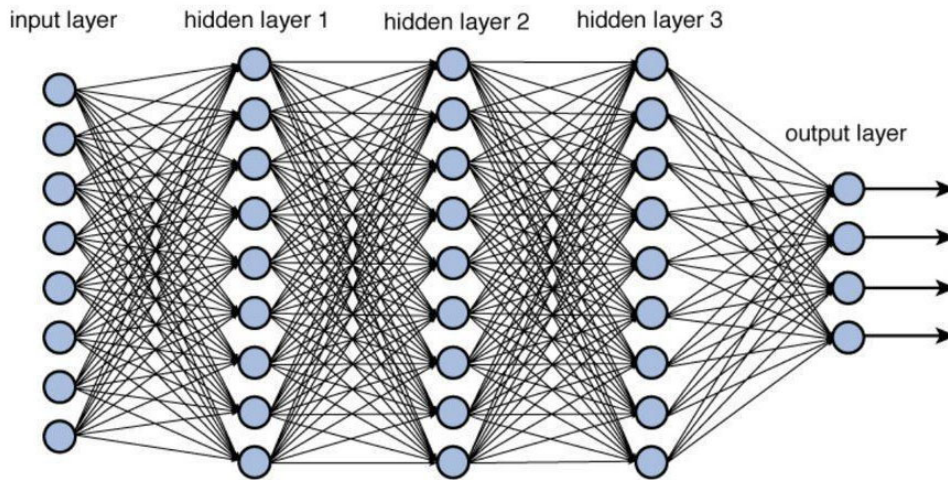


Fig. 3. A deep neural network with multiple layers

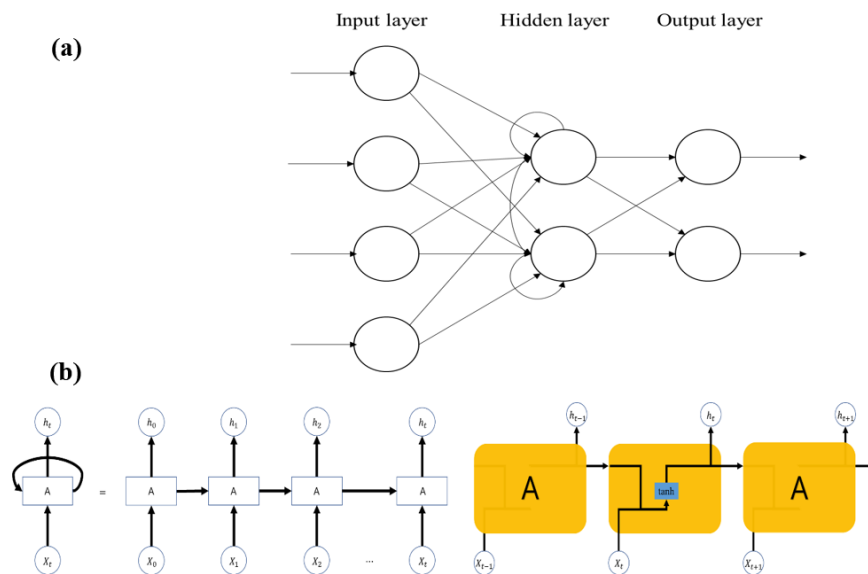


Fig. 4. The structure of RNN 2

Fig. 4b shows the unfolding in time of the computation involved in its forward computation. Also presented in Fig. 4, the output h_t is produced from input x_t through neural network A. The loop transfers the data to the next step. Via the loop, each independent data becomes dependent on each other. RNN can be seen as multiple copies of the same network.

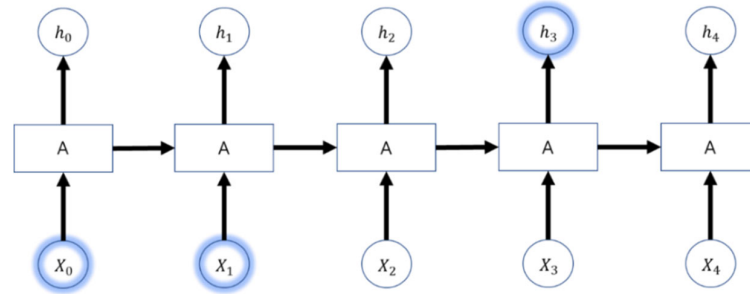


Fig. 5. Short-term memory of RNN

Fig. 5 indicates that RNN excels in short-term memory. The output h_3 contains the information of input x_0, x_1 . However, Fig. 6 shows that RNN is not good at long-term memory. The output h_{t+1} cannot consider the information of input x_0, x_1 . RNN processes the next data by memorizing the recent data but it loses the information of previous data as time elapsed. This problem is called the problem of long term dependencies. As the distance between output and input increases, RNN cannot learn the information of input data.

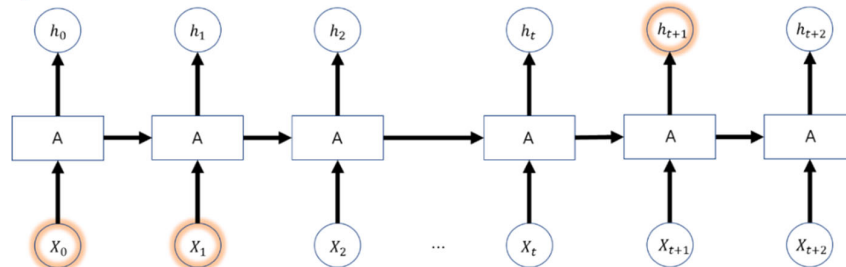


Fig. 6. Long-term memory of RNN

LSTM is a specific version of RNN. LSTM outperforms other RNN-based models (Hochreiter & Schmidhuber, 1997). It is useful because both the long term dependency problem and gradient vanishing problem which occurs during backpropagation are solved. LSTM sums the weights instead of multiplication to solve the vanishing gradient problem. Also, the model continuously transfers the information of historical data to solve long term dependency problem. The structure of LSTM is given in Fig. 7.

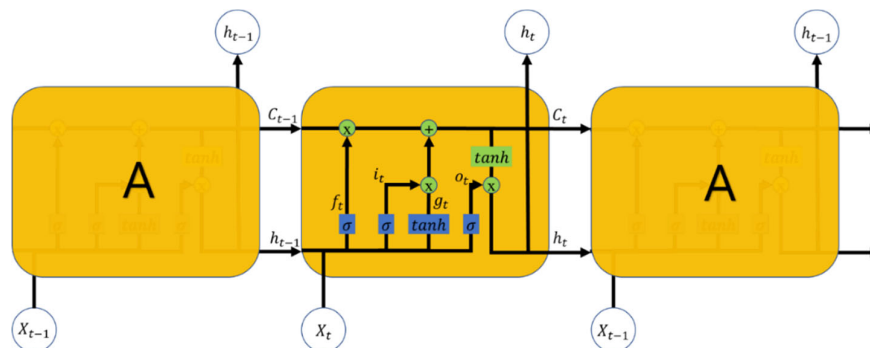


Fig. 7. The structure of LSTM

LSTM has four network layers for each module. It calculates the hidden layer using memory cell, instead of neural. The yellow box represents the trained network layer (hidden layer). The green circle indicates arithmetic operation such as vectored sum. The arrow is the flow of vector, which transfers the entire single vector from the output of a node to the input of another node.

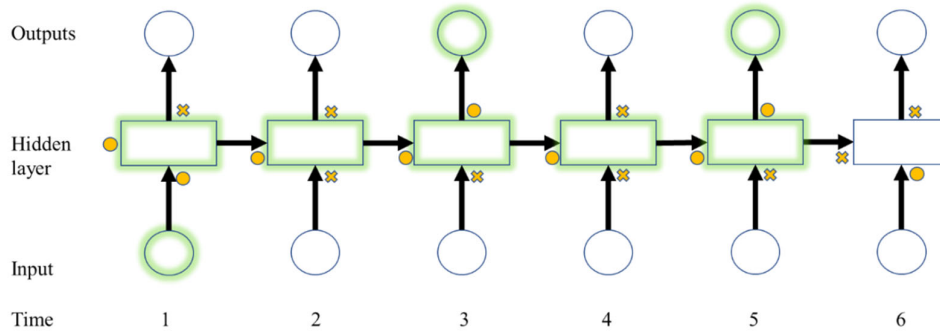


Fig. 8. LSTM neural network consisting of the unrolled LSTM cells

LSTM is able to add or remove the information to cell state via the gate. It carefully controls this procedure in the gate. As shown in Fig. 8, LSTM updates the information, selectively. The gate is responsible to add or remove information selectively, and LSTM controls the gate to discard of memory the previous information. In addition, the gate adds or eliminates new information. The gate is composed of multiplication for each factor and the sigmoid network layer. The output of sigmoid layer is between 0 and 1, which indicates the number of factors to be passed. The gate discards or eliminates the information for output 0 whereas memorizes or adds the information for output 1. Fig. 9 represents the LSTM network cell at time step t .

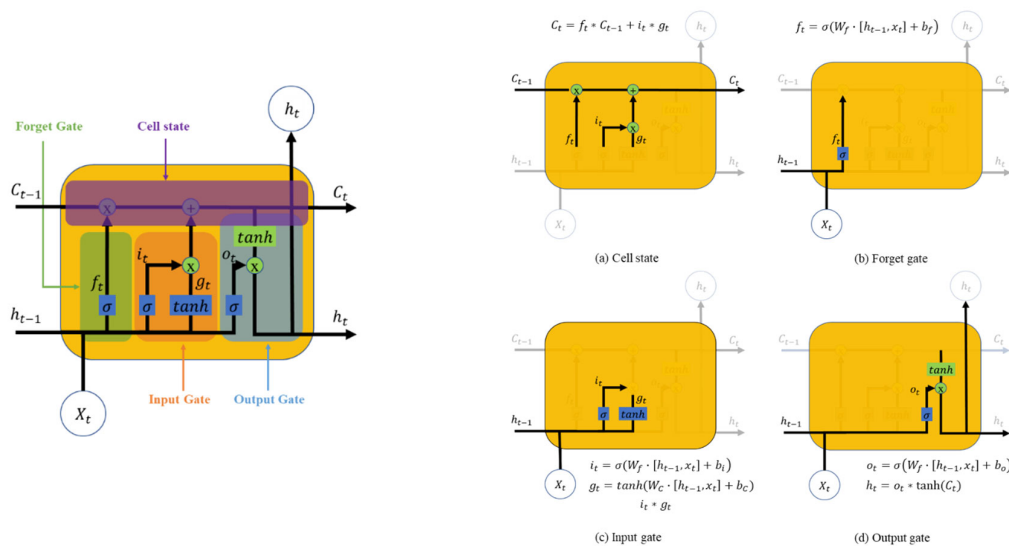


Fig. 9. LSTM network cell at time step t

Fig. 10. The internal operation process of LSTM

Fig. 10 depicts the operation process of LSTM’s memory cell. Forget gate determines which information of previous cell state to be eliminated. It is composed of sigmoid functions and behaves depending on the output of forget gate. Output 1 and 0 indicates retention and elimination, respectively. Input gate determines whether to store the new data to cell state or not. In the input gate, the value to be updated is determined by sigmoid function and the vector to be added to cell state is generated by tanh function. Cell state updates the previous cell state to a new state. Output gate decides the final output. It outputs the filtered value based on cell state.

4. Research design

This section applies the hierarchical ANFIS and LSTM models to the prediction of stock index. In this study, an application related to the context of Vietnam (Vietnam index) is used as an illustration.

4.1. Data description

In this study, the research data are daily opening prices of VNINDEX from January 3, 2001 to August 30, 2019 (Fig. 11). There are total 4538 trading days in this period. The data is divided into two datasets: the first dataset with 70% of the source data are used for the model development (training) and the other portion (30%) is for testing and evaluating the model.



Fig. 11. Daily opening prices of VNINDEX from January 3, 2001 to August 30, 2019

4.2. The proposed framework

The overall research process is shown in Fig. 12. First, VNINDEX data are collected. Then, the pre-process is performed for analysis. In this step, NaN data and abnormal data are removed and normalized is performed after extracting the necessary data and converting them to time series data. The pre-processed data is separated as training and testing datasets. The model development is done by the use of training dataset. Through the validation step with testing dataset, the optimal model is obtained. Using the optimal model, the VNINDEX forecasting is conducted.

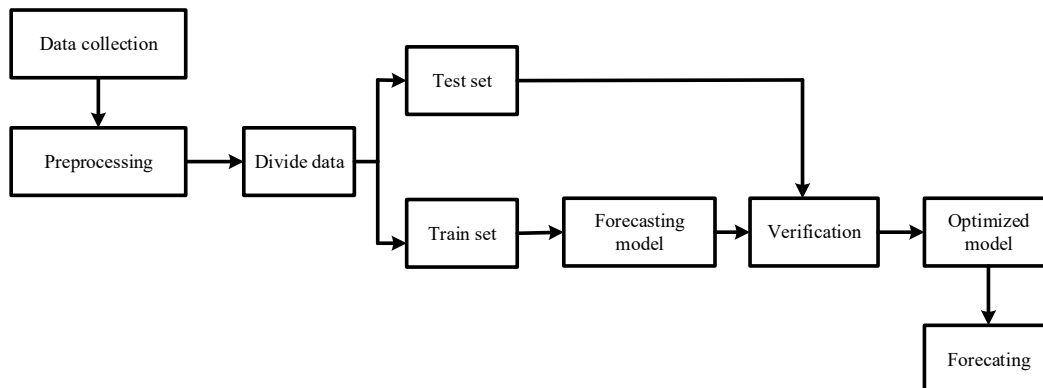


Fig. 12. The development steps of forecasting model

4.3. Development of the hierarchical ANFIS-based model

The grid partition method was used for FIS (Fuzzy Inference System) generation. In our study, two membership functions were chosen for each input in the model. The Gaussian membership function was used in ANFIS model. The membership function output is linear. We have used lagged versions of the variable, five input variables are given $[(x - 5), (x - 4), (x - 3), (x - 2), (x - 1)]$. As discussed earlier, a two-layer ANFIS structure was introduced to decrease the dimension of the rule

base. Layers 1 have three input variables and layer 2 has two input variables; each layer has one output. Fig. 13 represents the hierarchical ANFIS model, where x_1 - x_5 are input variables and y represents one output.

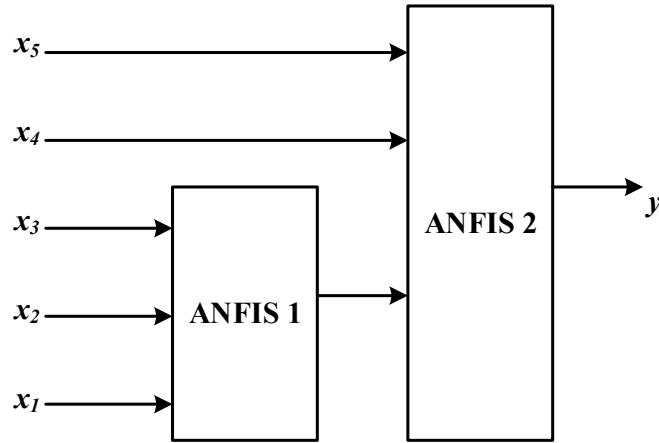


Fig. 13. The HANFIS structure for proposed model

4.4. Development of the LSTM-based model

The number of layer was set equal to 5. The sequential structure with a linear stack of layers was applied to the model development. The main parameters were set as follows: the dimensionality of the output space is 50; the rectified linear unit was used as the activation function. The cost function was mean squared error (MSE).

4.5. Model evaluation

To evaluate the performance of the forecasting model, several performance indexes were used. These criteria are applied to the developed model to know how well it works. The criteria were used to compare predicted values and actual values. They are as follows:

Root mean squared error ($RMSE$): This index estimates the residual between the actual value and desired value. A model has better performance if it has a smaller $RMSE$. An $RMSE$ equal to zero represents a perfect fit.

$$RMSE = \sqrt{\frac{1}{m} \sum_{k=1}^m (t_k - y_k)^2}, \quad (8)$$

where t_k is the actual (desired) value, y_k is the predicted value produced by the model, and m is the total number of samples. Mean absolute percentage error ($MAPE$): This index indicates an average of the absolute percentage errors; a model has better performance if it has a smaller $MAPE$.

$$MAPE = \frac{1}{m} \sum_{k=1}^m \left| \frac{t_k - y_k}{t_k} \right|. \quad (9)$$

Mean absolute error (MAE): This index indicates how close predicted values are to the actual values. A model with a lower MAE means it has better performance.

$$MAE = \frac{1}{m} \sum_{k=1}^m |t_k - y_k| \quad (10)$$

Correlation coefficient (R): This criterion reveals the strength of relationships between actual values and predicted values. The correlation coefficient has a range from 0 to 1, and a model with a higher R means it has better performance.

$$R = \frac{\sum_{k=1}^m (t_k - \bar{t})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^m (t_k - \bar{t})^2 \cdot \sum_{k=1}^m (y_k - \bar{y})^2}}, \quad (11)$$

where $\bar{t} = \frac{1}{m} \sum_{k=1}^m t_k$ and $\bar{y} = \frac{1}{m} \sum_{k=1}^m y_k$ are the average values of t_k and y_k , respectively.

5. Results and discussions

Validation of HANFIS and LSTM models was performed with the testing data and the results of actual and forecasting values of both studied models are shown in Fig. 14. Results indicated the the forecasting values of VN index obtained from the testing dataset for HANFIS and LSTM models are in excellent correlation with actual experimental values.

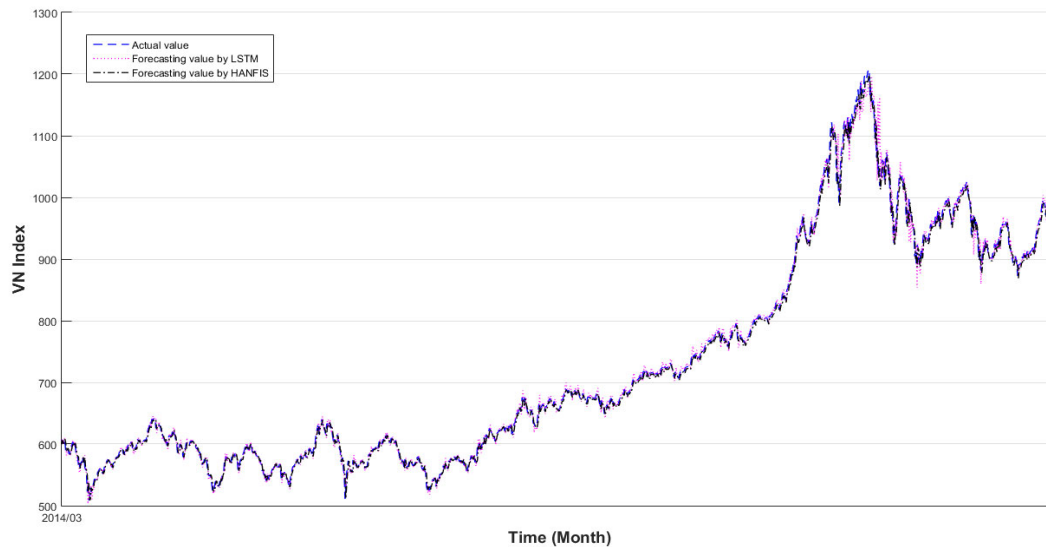


Fig. 14. Comparison between VNIndex predicted by HANFIS and LSTM models and actual

The average performance criteria for each model were calculated and are presented in Table 1. The performance criteria $RMSE$, $MAPE$, MAE , and R obtained by HANFIS model were calculated as 12.5689, 0.0102, 7.9529, and 0.9976. Performance indexes calculated by LSTM are 9.9566, 0.0091, 7.0809, and 0.9987 for $RMSE$, $MAPE$, MAE , and R , respectively. Theoretically, a forecasting model is regarded as good when $RMSE$, $MAPE$, and MAE are small, and R is close to 1. It can be seen from Table 1 that the LSTM model has smaller errors as well as a bigger R than those of the HANFIS model. The performance criteria indicate that the assessed result is highly correlated and precise.

Table 1

Performance statistics of HANFIS and LSTM models

Model	RMSE	MAPE	MAE	R
HANFIS	12.5689	0.0102	7.9529	0.9976
LSTM	9.9566	0.0091	7.0809	0.9987

The simulation results by HANFIS and LSTM models are presented in Figs. 15-16. It can be clearly observed that the actual values and the forecasting values obtained from the testing data of LSTM

model is in excellence agreement with the actual data (Fig. 15a). The corresponding errors between actual and forecasting values are plotted in Fig. 15b, along with the histogram of error (Fig. 15c). The values of errors were calculated as: $MSE = 99.1341$, $RMSE = 9.9566$, $mean\ error = 2.9313$ and standard deviation $St.D = 9.5188$. The results obtained by HANFIS model was also highlighted in Fig. 16. Similar observation is monitor for comparison between forecasting and actual values, with satisfactory error values, for instances, $MSE = 157.9774$, $RMSE = 12.5689$, $mean\ error = 0.62617$ and $St.D = 12.5579$. It can be concluded that both HANFIS and LSTM models have the effectiveness and good rate of success in forecasting. However, the results also indicate that the LSTM model achieves better performances than the HANFIS model.

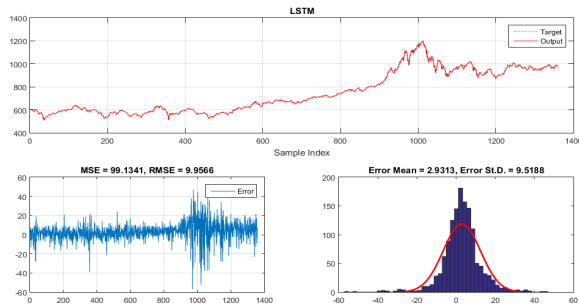


Fig. 15. VN Index forecasting performance by LSTM model: (a) forecasting and actual values, (b) error value and (c) standard deviation

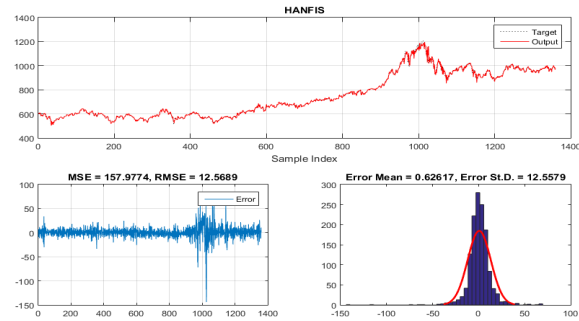


Fig. 16. VN Index forecasting performance by HANFIS model: (a) forecasting and actual values, (b) error value and (c) standard deviation

The comparison between actual values and corresponding output values obtained by the HANFIS and LSTM model are also shown in Fig. 17. The figure presents scatter diagrams that illustrates the degree of correlation between predicted values and actual values. In the figure, the 1:1 line was drawn as a reference. In a scatter diagram, the 1:1 line represents that the two sets of data are identical. The more the two data sets agree, the more the points tend to concentrate in the vicinity of the 1:1 line. It may be observed that most predicted values are close to the actual values in Fig. 17, and this indicates a good agreement between the forecasting values obtained by the HANFIS and LSTM models and the actual values.

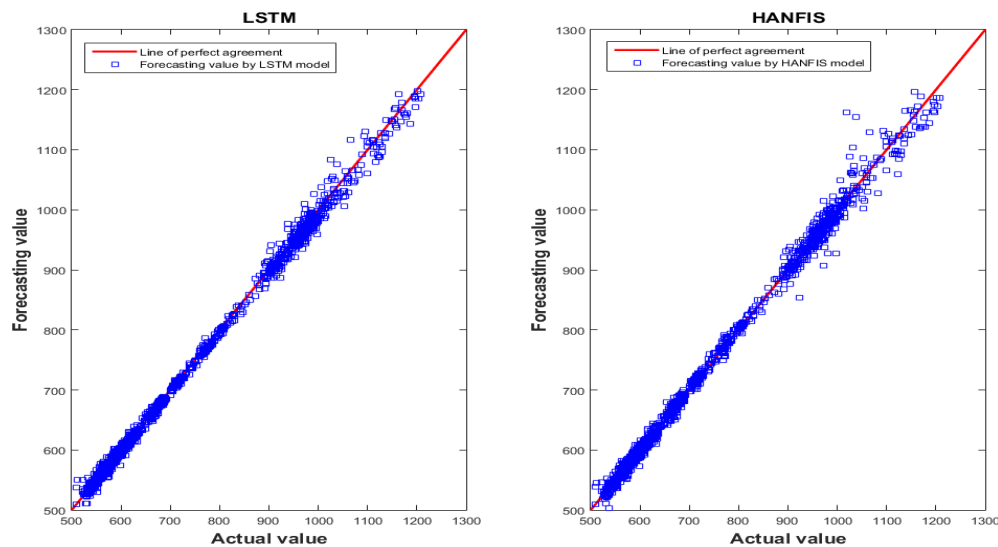


Fig. 17. The scatter plot of actual values and forecasts using HANFIS and LSTM models

Based on the obtained results, it can be concluded that both the HANFIS and ANN models can be used to predict VN Index. However, regarding forecasting accuracy, the LSTM model is highly appreciated. The LSTM model outperformed the HANFIS model, and the results show that its prediction outcome is more accurate and reliable. Hence, the LSTM may be acceptable and good enough to serve as a forecasting tool in forecasting VN stock index.

6. Conclusions

Financial time series, i.e., stock price index, has characteristics of classical nonlinearity and instability. In this study, we have analyzed and compared the ability of the ANFIS and LSTM models in forecasting VN Index. Several criteria namely *RMSE*, *MAPE*, *MAE* and *R* were used to evaluate the performance of the develop models. The results indicated that both ANFIS and LSTM can be promising tools for stock price prediction in emerging markets, like Vietnam. However, it was also showed that LSTM model was the most robust and powerful method, with respect to all performance criteria, for forecasting VN Index. The study findings show the forecasting potential of the artificial intelligence models in financial applications and are expected to provide an assistance and forecasting tool for managers and policy makers. For future research, the authors are exploring more techniques for stock index forecasting as ongoing research.

References

- Ata, R., & Koçyigit, Y. (2010). An adaptive neuro-fuzzy inference system approach for prediction of tip speed ratio in wind turbines. *Expert Systems with Applications*, 37(7), 5454-5460.
- Azadeh, A., Saberi, M., Anvari, M., Azaron, A., & Mohammadi, M. (2011). An adaptive network based fuzzy inference system–genetic algorithm clustering ensemble algorithm for performance assessment and improvement of conventional power plants. *Expert Systems with Applications*, 38(3), 2224-2234.
- Bao, T. Q., & My, B. T. T. (2019). Forecasting stock index based on hybrid artificial neural network models. *Science & Technology Development Journal-Economics-Law and Management*, 3(1), 52–57.
- Ake, B. (2010). The role of stock market development in economic growth: evidence from some Euronext countries. *International Journal of Financial Research*, 1(1), 14-20.
- Boyacioglu, M. A., & Avci, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange. *Expert Systems with Applications*, 37(12), 7908-7912.
- Brown, M., Bossley, K. M., Mills, D. J., & Harris, C. J. (1995, March). High dimensional neurofuzzy systems: overcoming the curse of dimensionality. In *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*. (Vol. 4, pp. 2139-2146). IEEE.
- Buragohain, M., & Mahanta, C. (2008). A novel approach for ANFIS modelling based on full factorial design. *Applied Soft Computing*, 8(1), 609-625.
- Esfahanipour, A., & Mardani, P. (2011, June). An ANFIS model for stock price prediction: The case of Tehran stock exchange. In *2011 International Symposium on Innovations in Intelligent Systems and Applications* (pp. 44-49). IEEE.
- Fakhrahmad, S. M., Rezapour, A. R., Jahromi, M. Z., & Sadreddini, M. H. (2012). A new fuzzy rule-based classification system for word sense disambiguation. *Intelligent Data Analysis*, 16(4), 633-648.
- Güneri, A. F., Ertay, T., & Yücel, A. (2011). An approach based on ANFIS input selection and modeling for supplier selection problem. *Expert Systems with Applications*, 38(12), 14907-14917.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. In *Prentice Hall*.

- Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665-685.
- Jeenanunta, C., Chaysiri, R., & Thong, L. (2018). Stock Price Prediction With Long Short-Term Memory Recurrent Neural Network. *2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*, 1-7. IEEE.
- Jian, Z., & Song, L. (2016). Financial Time Series Analysis Model for Stock Index Forecasting. *International Journal of Simulation--Systems, Science & Technology*, 17(16).
- Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert systems with Applications*, 38(5), 5311-5319.
- Kennedy, E. P., Condon, M., & Dowling, J. (2003). Torque-ripple minimisation in switched reluctance motors using a neuro-fuzzy control strategy. *Proceedings of the IASTED International Conference on Modelling and Simulation*.
- Kyungjoo, L., Schwan, Y., & John, J. J. (2007). Neural network model vs. SARIMA model in forecasting Korean stock price index. *Issues in Information Systems*, 8(2), 372-378.
- Liu, C., Wang, J., Xiao, D., & Liang, Q. (2016). Forecasting s&p 500 stock index using statistical learning models. *Open Journal of Statistics*, 6(06), 1067.
- Ertunc, H. M., & Hosoz, M. (2008). Comparative analysis of an evaporative condenser using artificial neural network and adaptive neuro-fuzzy inference system. *International Journal of Refrigeration*, 31(8), 1426-1436.
- Nauck, D., Klawonn, F., & Kruse, R. (1997). *Foundations of neuro-fuzzy systems*. John Wiley & Sons, Inc.
- Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PloS one*, 11(5), e0155133.
- Rahman, M. M., & Salahuddin, M. (2009). The determinants of economic growth in Pakistan: does stock market development play a major role? *Proceedings of the 38th Australian Conference of Economists (ACE 2009)*, 1-22. Economic Society of Australia (South Australian Branch).
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117.
- Senol, D., & Ozturan, M. (2008). Stock price direction prediction using artificial neural network approach: The case of Turkey. *Journal of Artificial Intelligence*, 1(2), 70-77.
- Singh, T. N., Kanchan, R., Verma, A. K., & Saigal, K. (2005). A comparative study of ANN and neuro-fuzzy for the prediction of dynamic constant of rockmass. *Journal of Earth System Science*, 114(1), 75-86.
- Sugeno, M. (1985). An introductory survey of fuzzy control. *Information Sciences*, 36(1-2), 59-83.
- Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, 5(3), 191-212.
- Touny, M. A. (2012). Stock Market Development and Economic Growth: Empirical Evidence from Some Arab Countries. *Arab Journal of Administration*, 32(1).
- Wei, M., Bai, B., Sung, A. H., Liu, Q., Wang, J., & Cather, M. E. (2007). Predicting injection profiles using ANFIS. *Information Sciences*, 177(20), 4445-4461.
- Yusof, N., Ahmad, N. B., Othman, M. S., & Mohammad, F. A. (2012). A concise fuzzy rule base to reason student performance based on rough-fuzzy approach. In *Fuzzy inference system-theory and applications* (pp. 63-82). INTECH Open Access Publisher.

