Investment Strategy Be Applied With Spherical Complex Neural Fuzzy System Model For Stock Time Series Forecasting

Chi Feng Lin

Information Management. National Central University

National Central University

Taoyuan, Taiwan

j8888888871@yahoo.com.tw

Chun Shien Li

Information Management. National Central University

National Central University

Taoyuan, Taiwan

jamesli@mgt.ncu.edu.tw

*Abstract*— This paper mainly discusses efficiency of different algorithms in the time series prediction, and through the simulation of investment strategy to do performance testing. The prediction of time series is a very broad and important issue. We propose a spherical complex neural fuzzy system (SCNFS) to predict the time series. Through the general complex neural fuzzy system (CNFS), we can obtain the complex outputs, and the real part and the imaginary parts can be predicted for different targets. In this model, the complex fuzzy sets (CFSs) in the original CNFS is improved, the value of membership degree is still complex-valued, but it can have multiple sets of outputs, and it is possible to predict more than two targets at the same time. In the model design, the premise part uses the spherical complex neural fuzzy sets (SCFSs) of the Gaussian type, while the consequence part uses the linear function of Takagi-Sugeno, and the two are linked by the IF-THEN rule. In addition, in order to optimize the prediction results of the model, we use different algorithms like Particle Swarm Optimization (PSO), Artificial Bee Colony Optimization (ABCO) to optimize the parameters, the consequence part uses Recursive Least Squares Estimator (RLSE) for parameter optimization. Finally, we will use three experiments to test the performance difference under different learning algorithms.

Keywords—Spherical complex neurofuzzy system (SCNFS); complex fuzzy set (CFS); complex neurofuzzy system (CNFS), time series forecasting.

# INTRODUCTION

Artificial Intelligence has many fields in reality, from 18th century, there are many related research, but problem of hardware equipment leads to that the results can’t be unable to apply in general life. Now with the development of hardware equipment, artificial intelligence has brought a lot of help to mankind, research is more numerous, such as financial forecasting [1], image identification [2][3], games [4] and so on. These are part of the prediction of time series, and the prediction of time series is an important issue because of the wide range of applications in the real world. In the past, many scholars have put forward different methods to predict the time series, such as ARIMA [5][6], fuzzy theory, neural network operation, neuro-fuzzy hybrid system and so on. Among them, the most commonly proposed is the neuro-fuzzy hybrid system [7]- [10].

Neural fuzzy system (NFS) is based on the extension of neural network, the so-called neural network is a network made of multilayer neurons, through the data transmission between layer and layer to get output, now the popular issue ‘deep learning’ is a kind of neural network, there are many studies in this field [2][3][27].

The extension of neural network, neuro-fuzzy system (NFS) has been a widely studied model, neural network systems have IF-THEN rules, these rules are like our human experience rules. IF-THEN rules can often be combined with fuzzy theory to make the overall architecture more resilient, which we call neuro-fuzzy systems. As mentioned above, the neural-fuzzy system has a good effect on the prediction of time series. Therefore, most of the research on time series prediction is based on the neural network model.

In this experiment, the neuro-fuzzy system adopts the IF-THEN rule to construct the multilayer neurons structure. In the model implementation, we combine the fuzzy theory with the neural network system to form a neuro-fuzzy system, in which the premise part uses a spherical complex neural fuzzy sets (SCFSs) of Gaussian type, the consequence part uses a linear function of Takagi-Sugeno, and the two are combined by IF-THEN rules. Through this model and machine learning, we expect to be able to predict time series more accurately.

With regard to fuzzy sets, the concept of fuzzy sets was first proposed by Zadeh in 1965 [11], so that data can be attributed to a degree between 0 and 1 through a function. Then in 2002, another study presented the concept of complex fuzzy sets (CFSs) [12], in which the complex-valued membership degree can be obtained by the function, which allows the membership degree to present in a unit disk of the complex plane (UDCP) with a radius of 1. This concept makes it possible to express a more substantial membership degree. In general, we can obtain a set of outputs of a complex-valued through a complex neural fuzzy system (CNFS) [13][14], and the real part and the imaginary part can be predicted for different targets, so a set of output can be used for two targets. At present, the prediction of the two targets has a lot of research [6][13][14]. In order to predict more targets at the same time, this paper improves the original complex neural fuzzy system and changes the original complex fuzzy sets (CFSs) to spherical complex fuzzy sets (SCFSs). The membership degree is still complex-valued, but can be presented in 3-D stereo space and have more sets of complex-valued output, meaning that multiple targets can be predicted at a time.

In this study, in order to enable the data to be used effectively, in the part of the data preprocessing, we use the up and down values of the original data as features, and calculate their individual contributions to the target through Shannon Entropy [15], we use the concept of multi-objective feature selection [16] to calculate the effective information for each feature on the target as a basis for selecting training data. Extracting the most effective data from the data not only reduces the computational burden of the model, but also improves the effectiveness of the prediction. Finally, in the machine learning section, we individually use the popular Particle Swarm Optimization (PSO) [18], Artificial Bee Colony Optimization (ABCO) [29] to combine with the well-known Recursive Least Square Estimator (RLSE) method to optimize the parameters, and these two hybrid method called PSO-RLSE and ACO-RLSE. We train the parameters of the premise part and the consequence part by different algorithms, and try to reduce the searching dimension through the Divide-and-conquer principle for making the model easier to find the best solution and improving the overall efficiency.

# METHODOLOGY

## Multi-Target Feature Selection

In order to use data efficiently, we preprocess the data for decreasing computing cost and improving model precise. Original data set denoted as , where, is the number of data amounts, is the number of data groups. Then make original data first difference, express as follows:

(1)

where is the number of data amounts; is the data variable. In each data group, use the first difference data to be 30 feature variables. Features in first data group denoted as to , and features in second data group denoted as to , and so on. Through multi-target feature selection method [16], we can get training data from features. Multi-target feature selection regards entropy which means chaos degree of information. If the random of information is high, and it means that entropy is high too. Entropy definition is given as follows:

(2)

where is expectation of ;  is the probability density of , but if bigger than 1, the part is negative, it will effect entire expectation, so we revise formula as follows:

(3)

(4)

where, .

Because our feature selection is for multiple targets, we have to calculate influence information of every feature variable to per target variable, as follows:

(5)

where is influence information of the feature variable to the target variable ; is mutual information of the feature variable x to the target variable y, when values in the feature variable x are positive; is mutual information of the feature variable x to the target variable y, when values in the feature variable x are negative. As follows:

(6)

(7)

where is expectation of target; is expectation of the target variable y, when values in the feature variable x are positive; is expectation of the target variable y, when values in the feature variable x are negative. As follow:

(8)

(9)

(10)

(11)

where ; is probability density of the feature variable x, when values in the feature variable are positive; is probability density of the target variable , when values in the feature variable are positive; is probability density of the feature variable x, when values in the feature variable are negative; is probability density of the target variable , when values in the feature variable are negative.

Through the above influence information formula, we can get the influence information of each feature to each target, and then we can choose the features by these steps, which are as follows:

Step 1: Compute selection gain which is the feature variable to the target variable, denoted as . Selection gain is given as follows:

(12)

where is the feature variable; is the target variable; is influence information of to ; is redundancy information of to the feature variables which have been selected in ; is selected features pool. Redundancy is given as follows:

(13)

where is number of features in ; is influence information of to the feature variable in ; is influence information of the feature variable in to . If is bigger than 0, select into

Step 2: Record all feature variables in the selected feature pools , either overlapping or non-overlapping, denoted as , where is the feature variable in . For , calculate overlap count in each SP, denoted as .

Step 3: Through , we can calculate covering rate , which is given as follows:

(14)

calculate the mean of , denoted as .

Step 4: For , calculate sum of the selection gains, as follows:

(15)

calculate the mean of , denoted as .

Step 5: For , according to the sum of selection gains and the covering rate , calculate effective contribution index :

(16)

Step 6: Test all feature variables in , if , then accumulate .

Step 7: Set lower and upper limits, which are denoted as and . Through lower and upper limits, find the which is number of the final selected feature variables. In this study,  is 4; is 2. If is between and , set to ; if is smaller than , set to ; if is bigger than , set to

Step 8: Sort , select top feature variables into final selected pool as the result of multi-target featrue selection.

## Structure Learning

Structure learning is for putting the training data into model more logically. Through the multi-target feature selection, we can get some selected feature variables which are assumed as the training data to put into the model.The training data denoted as , is the feature variable; is number of the selected feature variables, it is same as number of input dimesions. In this study, we use a function called «subclust()» which is offered by MATLAB to cluster each input dimension data. After clustering, use the centers and standards of each cluster to construct Gaussian-type fuzzy set, which is given as follows:

(17)

where is input variable; and are cluster center and standard. According to the combination of the different input dimension fuzzy set, we can get premises which are defined as below:

*Premise :*

*IF*  (18)

where is the linguistic variable; is input linguistic variable fuzzy set of the premise; is input variable, for .

For model computing efficiency, we do not use all premises in model. So we use the firing strength to make premises selection for decreasing the number of premises. Steps are given as follows:

Step 1: The firing strength of the premise can got from fuzzy set of each input dimension, denoted as follows:

(19)

where is the firing strength of the input variable in the premise; is the data in the input dimension, for , where n is number of the data in an input dimension; is the fuzzy set of the input dimension in premise.

Step 2: The sum of the firing strength in each premise is denoted as , as follows:

(20)

where is number of the data. Calculate the mean of which is denoted as , standard is denoted as .

Step 3: Identify each premise, if , accumulate . Set lower and upper limit which are denoted as and , through lower and upper limit, find the which is the number of the final selected premises. In this study, is 15, is 4. If is between and , set to ; If smaller than , set to ; if is bigger than , set to

Step 4: Sort , select top premises as the neuros of the premise layer in model.

Training data set is denoted as , is the number of data. This model uses IF-THEN rules to linked premise part and consequence part, the number of the consequence part is same as premise’s, and the consequence part is Takagi-Sugeno type, which is defined as follows:

(21)

where {} are parameters of the consequence; is the input variable.

## Model Structure and I/O Relationship

There are 6 layers in neural network in this study. The training data set is denoted as , is the number of data, is a input vector, is the number of input dimensions; is a target vector, is the number of complex-valued targets. The outputs of the model are .

Layer 1: This layer is called the input layer, inputs are the feature variables which are selected by multi-target feature selection, inputs sent to next layer directly. The input layer vector at time is given as follows:

(22)

Layer 2: This layer is called the sphere-complex-fuzzy-set layer, through the clustering of the structure learning, we can construct some fuzzy sets in each dimension and get the membership degrees from them via putting the input variables of different input dimensions. In general, Gaussian-type fuzzy set can only get one real-valued membership degree. Through sphere complex fuzzy set (SCFS) can get lots of complex-valued membership degrees, and they are in a sphere which radius is 1, like Fig. 1. The different membership degree is for different model output to achieve multiple target forecasting. Through concept of the SCFSs, we can get lots of values, as follows:

(23)

(24)

(25)

where is Gaussian function (17);;. Through concept of the SCFSs, we can get at least four membership degree, they include the method of keeping dimensions, as follows:

(26)

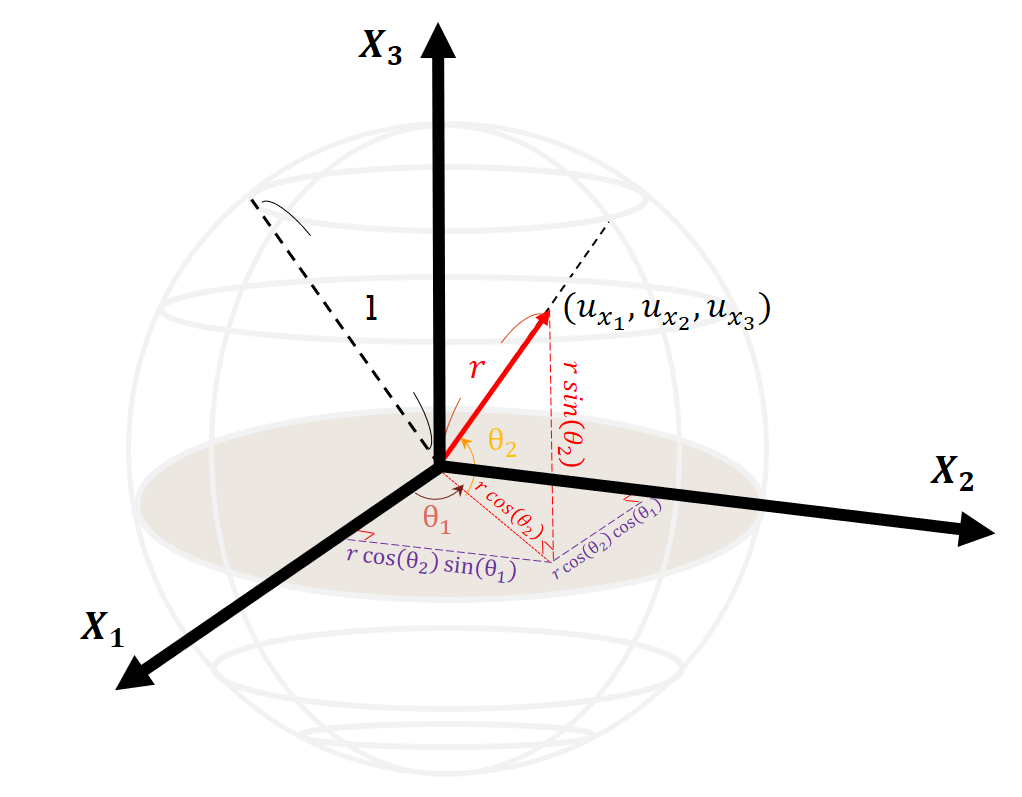
and the method of reducing dimensions, as follows:

(27)

(28)

(29)

where . Finally, we can get a vector



1. Unit complex sphere

Layer 3: This layer is called premise layer. After structure learning, we have useful premises, which are denoted as follows:

*Premise :*

*IF*  (30)

where is the  input linguistic variable; is the fuzzy set of the input linguistic variable in the premise; is the input variable, for . Multiplied by each membership degree which from the different input dimension, we can get the firing strength of each premise, as follows:

(31a)

(31b)

where is the membership degree of the input variable in the premise; is the firing strength of the premise. Finally, we can get a vector

Layer 4: This layer is called normalization layer. It is for normalizing the firing strength, the neuro input is , for Normalization formula is given as follows:

(32)

where is the firing strength of the premise. Finally, we can get a vector

Layer 5: This layer is called consequence layer, we can get model outputs after calculating, is same as number ， output formula as follows:

(33)

where， {} are the parameters of the consequence; is the number of consequences.

Layer 6: This layer is called output layer. Combine the outputs and we can get model outputs, as follows:

(34)

## Parameter Learning

According to divide-and-conquer concept, we use different machine learning algorithms to optimize parameters of each layer for searching best solution much easier. In this study, we individually use PSO and ABCO to optimize the parameters of premise part, and RLSE is used in the consequence part. These machine learning algorithms are as follows:

1. Particle Swarm Optimization (PSO)

We adapt well-kwon PSO [18] to optimize the parameters of the premises, its principle is like social behavior of bird flocking or fishing schooling, its features are adjusting step size automatically, certainty and randomness, algorithm is given as follows:

(35)

(36)

where is the position of the particle at iteration; is the velocity of the particle at iteration; is the best position of the particle at iteration; is the best position of all particle at iteration; are parameters of PSO; are random values between 0 to 1. In this experimentation, position of the particles is the premises parameters, it contains cluster centers, cluster standards, and

1. Artificial Bee Colony Optimization (ABCO)

There are three kinds of bees in this algorithm, including employed bees, onlooker bees and scout bees, but they are not directly related to real bees looking for food. The steps are as follows:

Step 1: Randomly form position of employed bees and update their position, as follows:

(37)

where, is the dimension value of the employed bee after moving; is the dimension value of the employed bee before moving; is the dimension value of the another random employed bee’s position.

Step 2: Use roulette method to select a position, the formula of roulette probability is as follows:

(38)

where, is the roulette probability of the bee; is the benefit of the employed bee, in this study, we use RMSE’s reciprocal as benefit; is the number of employed bees.

Step 3: All onlooker bees search around the selected bee, as follows:

(39)

where， is the dimension value of the onlooker bee after moving; is the dimension value of the employed bee before moving; is the dimension value of the selected employed bee after moving.

Step 4: If the employed bee did not update the position until reach the limit, then assign the scout bee to replace it, the position of the scout bee as follows:

(40)

where, is the dimension initial value of the employed bee; is the maximum of the dimension of all employed bees; is the minimum of the dimension of all employed bees.

Step 5: Repeat step 2- 4 until iteration is over.

1. Recursive Least Square Estimator (RLSE)

In this experimentation, RLSE [19] is used to update the parameters of the consequences, a least square estimation problem is specified by a linearly expression, which is given as follows:

(41)

where is target; is model output; {} are known function of ; {, =1,2,…,m} are the parameters of the model to be estimated; is the error of the entire model. Least square estimation problem also can be expressed as matrix, which is given as follows:

(42a)

where:

(42b)

(42c)

(42d)

(42e)

whereis an input matrix; is an parameters matrix which is unknown; is an target matrix; is an error vector. To optimize can compute through RLSE [19] equation:

(43a)

(43b)

where ; is the row of In the beginning of RLSE algorithm, we set to 0, and set to **,** whereis anidentity matrix. We use root-mean-square errors (RMSE) to be our cost function, which is defined as follows:

(44)

where is conjugate compute; , where is target vector; is model outputs vector. Entire process of the PSO-RLSE method and ABCO-RLSE is given as follows:

Step 1: Prepare the training data.

Step 2: Use position of the particle to calculate the firing strength.

Step 3: Use RLSE to update the parameters of consequences, and vector are given as follows:

(45a)

(45b)

(45c)

where 。

Step 4: After updating all parameters, calculate the model outputs.

Step 5: Calculate RMSE.

Step 6: Repeat step 2- 5 for all particles until the iteration is over.

# METHODOLOGY FOR FINANCIAL APPLICATION

In order to assess whether the model makes money, it is not enough to use RMSE, because there is no way to tell us whether model is profitable, we only can understand fit rate from RMSE, and the better fit rate does not mean higher profits. Therefore, this experiment will apply predicted values with the investment strategy, further decide to buy or sell, and then calculate the profits which model can make, buy and sell formula is as follows:

buy: (46a)

sell: (46b)

where， is the output of model, which means the predicted closing price of the day; is actual opening price of the day. If the predicted closing price is higher than the actual opening price, means buy is the best choice because it can make money until the end of the day; If the predicted closing price is lower than the actual opening price, means sell is the best choice.

The method of estimating the profits is calculated by the actual closing price and the opening price, and the formula is as follows:

(47)

where, is the profit value, is the number of days which strategy is buy; is the number of days which strategy is sell; is the actual closing price of day.

Through the above formula, we can obtain the profit value of the whole model and roughly simulate the effect of the model on the real world. In this study, the profit value and other parameters will be shown in each experiment.

# EXPERIMENTATION

## Example 1— Quadruple Time Series of Daily Dow Jones Industrial Average Index

In this experimentation, we use the real-world time series data to testify the model performance, four targets are closing price of The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX), Dow Jones Industrial Average Index (DJI), National Association of Securities Dealers Automated Quotation (NASDAQ) and Standard & Poor's 500 (S&P500). To compare performance with other paper, we forecast the closing price for 2001, with data for the first ten months of 2001 treated as training data and the rest as test data, with a total of 181 training data and 66 test data. The model presented in this paper can have multiple complex-valued outputs at once, so it can predict multiple targets. In this experiment, the real part of the first complex-valued target is the closing price of TAIEX in 2001, the imaginary part of the first complex-valued target is the closing price of DJI in 2001, the real part of the second complex-valued target is the closing price of NASDAQ in 2001, and the imaginary part of the second complex-valued target is the closing price of S&P500 in 2001.

1. Setting of SCNFS (Experimentation 1)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | 3, 3, 3, 3 |
| Number of targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Number of premises (before selection) | 81 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 9 |
| Number of premise parameters | 48 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 9 |
| Number of consequence parameters | 45 |

In the part of structure learning, we use the function “subclust()” which is offered from MATLAB to cluster the selected feature variables; parameter of the function is 0.2. Through premise selection, we extract 9 premises from 81 original premises. The parameters after structure learning is shown in table I; the machine learning parameters are shown in table II.

1. Setting of Algorithm (Experimentation 1)

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 45x1 zero vector |
|  | **I** |
| **I** | 45x45 identify matrix |

In this experiment, besides compare performance between PSO-RLSE and ABCO-RLSE, we also compare with the methods proposed by other papers, like ANFIS [21], CNFS-ARIMA [6], RBF network [22] and SVR [23][24]. Except SVR, all methods can predict two targets at same time. And we will use the first complex-valued output to compare, the result is shown in Table III. The learning curve of machine learning is shown in Fig. 2.; targets and model outputs are shown in Fig. 3.; the RMSE of algorithms is shown in Table IV, where and mean they only run 1 iteration; the profits from the simulation of investment strategy and the number of transactions are shown in Table V.



(a)



(b)

1. Learning Curve (Experimentation 1) (a) PSO-RLSE (b) ABCO-RLSE
2. Performancecomparison of Multi-target (RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | DJI |  | TAIEX | DJI |
| SVR (two models, each with single output) [23] | - | - |  | 162.46 | 101.44 |
| ANFIS (two models, each with single output) [21] | - | - |  | 147.36 | 105.56 |
| ANFIS (one model with two outputs) [21] | - | - |  | 151.62 | 128.20 |
| RBF (two models, each with single output) [22] | - | - |  | 134.32 | 106.33 |
| RBF (one model with two outputs) [22] | - | - |  | 137.58 | 181.79 |
| CNFS(4)-ARIMA (one model with two outputs) [6] | - | - |  | 115.82 | 103.06 |
| PSO-RLSE (proposed) | **94.43** | **140.71** |  | **101.45** | **101.83** |
| ABCO-RLSE (proposed) | **95.06** | **133.74** |  | **101.38** | **102.65** |



(a)



(b)



(c)



(d)

1. Targets and model outputs (a) PSO-RLSE (TAIEX) (b) PSO-RLSE (DJI) (c) ABCO-RLSE (TAIEX) (d) ABCO-RLSE (DJI)
2. Performance comparison in 10 trials (RMSE)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |  | |  | |
| TAIEX | DJI | TAIEX | DJI | TAIEX | DJI | TAIEX | DJI |
| 1 | 104.76 | 104.16 | 103.96 | 102.87 | 101.85 | 102.73 | **102.81** | 115.99 |
| 2 | 118.44 | 133.07 | 102.74 | 103.15 | 103.00 | 102.45 | 128.57 | 194.34 |
| 3 | 102.38 | 103.20 | 101.38 | 102.65 | 118.44 | 133.06 | 113.29 | 486.25 |
| 4 | 102.14 | 104.13 | 103.02 | 104.39 | 102.73 | 104.67 | 115.79 | **103.46** |
| 5 | 103.38 | 103.72 | 103.96 | 102.87 | 102.84 | 101.87 | 112.13 | 111.17 |
| 6 | 102.21 | 103.32 | 103.96 | 102.87 | **101.68** | 103.56 | 198.62 | 308.72 |
| 7 | **101.45** | 103.87 | 102.74 | 103.15 | 102.83 | 102.92 | 110.51 | 126.32 |
| 8 | 102.75 | 102.42 | **101.38** | **102.65** | 103.30 | **101.24** | 104.66 | 104.76 |
| 9 | 102.06 | 102.44 | 103.96 | 102.87 | 103.03 | 102.82 | 114.03 | 126.89 |
| 10 | 102.87 | **101.83** | 102.74 | 103.15 | 107.92 | 114.24 | 171.89 | 102.87 |

1. simulation of investment (Experimentation 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |  | |  | |
| Profits | Sell/buy | Profits | Sell/buy | Profits | Sell/buy | Profits | Sell/buy |
| 1 | 1827 | 127/119 | 1745 | 125/121 | 1774 | 129/117 | **2623** | 127/119 |
| 2 | 2018 | 130/116 | 1866 | 126/120 | 1658 | 127/119 | 702 | 141/105 |
| 3 | 1871 | 128/118 | 1745 | 125/121 | **2019** | 130/116 | 903 | 124/122 |
| 4 | 2072 | 125/121 | 1817 | 126/120 | 1909 | 127/119 | 2087 | 126/120 |
| 5 | 1933 | 127/119 | 1745 | 125/121 | 1929 | 128/118 | 1353 | 124/122 |
| 6 | **2059** | 131/115 | 1745 | 125/121 | 1655 | 128/118 | 2072 | 143/103 |
| 7 | 1753 | 127/119 | 1866 | 126/120 | 1745 | 125/121 | 1815 | 131/115 |
| 8 | 1983 | 126/120 | 1745 | 125/121 | 1980 | 128/118 | 2474 | 128/118 |
| 9 | 1730 | 128/118 | 1745 | 125/121 | 1638 | 126/120 | 2007 | 126/120 |
| 10 | 1785 | 129/117 | **1866** | 126/120 | 1438 | 127/119 | 1783 | 128/118 |
| Mean | 1903 | - | 1788 | - | 1774 | - | 1782 | - |

## Example 2—Quadruple Time Series of Daily Taiwan Stock Exchange Capitalization Weighted Stock Index

In this experimentation, we use the real-world time series data to testify the model performance, two targets are closing price of The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) and Hang Seng Index (HSI). To compare performance with other paper, we forecast the closing price for 2000, with data for the first ten months of 2000 treated as training data and the rest as test data, with a total of 205 training data and 43 test data. The model presented in this paper can have multiple complex-valued outputs at once, so it can predict multiple targets. In this experiment, the real part of the complex-valued target is the closing price of TAIEX in 2000, the imaginary part of the complex-valued target is the closing price of HSI in 2000.

In the part of structure learning, we use the function “subclust()” which is offered from MATLAB to cluster the selected feature variables; parameter of the function is 0.05. Through premise selection, we extract 15 premises from 625 original premises. The parameters after structure learning is shown in table VI; the machine learning parameters are shown in table VII.

1. Setting of SCNFS (Experimentation 2)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | 3, 3, 3, 3 |
| Number of targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Number of premises (before selection) | 625 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 15 |
| Number of premise parameters | 48 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 15 |
| Number of consequence parameters | 75 |

1. Setting of algorithm (Experimentation 2)

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 75x1 zero vector |
|  | **I** |
| **I** | 75x75 identify matrix |

In this experiment, besides compare performance between PSO-RLSE and ABCO-RLSE, we also compare with the methods proposed by other papers, like Chen [31], Yu [32], AR (1) [33], SVR [23], ANFIS [21] and ANFIS (EMD) [34]. We will use the complex-valued output to compare not only the model performance but profits, and the results are shown in Table X and Table XI. The learning curve of machine learning is shown in Fig. 4.; targets and model outputs are shown in Fig. 5.; the RMSE of algorithms is shown in Table VIII, where and mean they only run 1 iteration; the profits from the simulation of investment strategy and the number of transactions are shown in Table IX.

1. Performance comparison in 10 trials (RMSE)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |  | |  | |
| TAIEX | HSI | TAIEX | HSI | TAIEX | HSI | TAIEX | HSI |
| 1 | 153.49 | 256.82 | 150.07 | 255.98 | 151.23 | 254.96 | 154.69 | **250.99** |
| 2 | 156.72 | 257.53 | 151.83 | 275.88 | 150.12 | 254.79 | 151.62 | 260.86 |
| 3 | 157.85 | 257.65 | **149.54** | 258.22 | 151.68 | 255.61 | **147.38** | 362.98 |
| 4 | 150.88 | 255.63 | 149.69 | 255.48 | **149.60** | **254.42** | 149.10 | 255.22 |
| 5 | 150.05 | **252.93** | 244.96 | 260.05 | 149.76 | 254.84 | 155.48 | 258.03 |
| 6 | 150.68 | 263.73 | 151.32 | 257.73 | 149.74 | 256.06 | 150.59 | 254.50 |
| 7 | 160.21 | 256.23 | 150.37 | 256.56 | 153.71 | 260.39 | 150.56 | 254.56 |
| 8 | **149.59** | 254.32 | 151.19 | **249.83** | 150.73 | 254.86 | 150.85 | 253.80 |
| 9 | 151.24 | 255.89 | 150.09 | 256.37 | 151.33 | 255.95 | 221.64 | 1210 |
| 10 | 154.54 | 266.08 | 172.72 | 463.91 | 159.98 | 255.96 | 394.39 | 15429 |

1. Simulation of investment (Experimentation 2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |  | |  | |
| Profits | Sell/buy | Profits | Sell/buy | Profits | Sell/buy | Profits | Sell/buy |
| 1 | 4338 | 114/131 | 3597 | 111/134 | 3802 | 114/131 | 3874 | 113/132 |
| 2 | 2957 | 121/124 | 3813 | 111/134 | **4096** | 113/132 | 2724 | 113/132 |
| 3 | 2799 | 115/130 | 3246 | 113/132 | 2486 | 117/128 | **4219** | 119/126 |
| 4 | 2845 | 111/134 | 4069 | 114/131 | 3803 | 114/131 | 3439 | 113/132 |
| 5 | 3822 | 112/133 | **4853** | 114/131 | 3820 | 112/133 | 3297 | 112/133 |
| 6 | **4530** | 117/128 | 3940 | 115/130 | 3859 | 113/132 | 3820 | 112/133 |
| 7 | 3273 | 118/127 | 4049 | 114/131 | 3410 | 118/127 | 2928 | 114/131 |
| 8 | 3945 | 111/134 | 3233 | 110/135 | 3974 | 113/132 | 3991 | 114/131 |
| 9 | 4202 | 111/134 | 4273 | 109/136 | 3832 | 118/127 | 3603 | 116/129 |
| 10 | 3019 | 111/134 | 4410 | 112/133 | 1317 | 118/127 | 3195 | 111/134 |
| Mean | 3573 | - | 3948 | - | 3440 | - | 3509 | - |

1. Performance comparison of Multi-target (RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | HSI |  | TAIEX | HSI |
| Chen [31] | - | - |  | 191 | 403 |
| Yu [32] | - | - |  | 176 | 435 |
| AR(1) [33] | - | - |  | 130 | 256 |
| SVR [23] | - | - |  | 136 | 250 |
| ANFIS [21] | - | - |  | 130 | 242 |
| ANFIS (EMD) [34] | - | - |  | 129 | 230 |
| PSO-RLSE (proposed) | **175.02** | **318.90** |  | **149.59** | **252.93** |
| ABCO-RLSE (proposed) | **172.86** | **304.93** |  | **149.54** | **249.83** |

1. Comparison of profits using various models (Experimentation 2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Year | Models | | | | | | | |
| Chen [31] | Yu [32] | AR(1) [33] | SVR [23] | ANFIS [21] | ANFIS (EMD) [34] | PSO-RLSE (proposed) | ABCO-RLSE (proposed) |
| 2000 | -92 | -73 | 671 | 202 | 686 | 795 | 4530 | 4853 |



(a)



(b)

1. Learning Curve (a) PSO-RLSE (b) ABCO-RLSE



(a)



(b)



(c)



(d)

1. Targets and the model outputs (a) PSO-RLSE (TAIEX) (b) PSO-RLSE (HSI) (c) ABCO-RLSE (TAIEX) (d) ABCO-RLSE (HSI)

## Example 3—Quadruple Time Series of Daily Taiwan Stock Exchange Capitalization Weighted Stock Index

In this experimentation, we also use the real-world time series data to testify this model performance, targets are closing price of APPLE Computer Inc., International Business Machines Corporation (IBM), Dell Inc. and Microsoft Inc. The period of stock prices is from 10 Feb. 2003 to 21 Jan. 2005; data volume comes to 492. In order to compare with other papers, we use data which period is from 10 Feb. 2003 to 10 Sep. 2004 be training data, data volume comes to 400, the rest of the data assumed as testing data, the testing data comes to 92. Note that proposed model has many complex-valued outputs, so it can predict multiple targets simultaneously. In this experiment, the real part of the first complex-valued target is the closing price of IBM, the imaginary part of the first complex-valued target is the closing price of APPLE, the real part of the second complex-valued target is the closing price of DELL, and the imaginary part of the second complex-valued target is the closing price of Microsoft.

In the part of structure learning, we use the function “subclust()” which is offered from MATLAB to cluster the selected feature variables; parameter of the function is 0.15. Through premise selection, we extract 8 premises from 81 original premises. The parameters after structure learning is shown in table XII; the machine learning parameters are shown in table XIII.

1. Setting of SCNFS (Experimentation 3)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | 2, 3, 3, 3 |
| Number of targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Number of premises (before selection) | 81 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 8 |
| Number of premise parameters | 44 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 8 |
| Number of consequence parameters | 40 |

1. Setting of algorithm (Experimentation 3)

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 40x1 zero vector |
|  | **I** |
| **I** | 40x40 identify matrix |

In this experiment, besides compare performance between PSO-RLSE and ABCO-RLSE, we also compare with the methods proposed by other papers, like HiMMI [35], ANN-GA-HMM-Interpolation [35], ANN-GA-HMM-WA [35] and ARIMA [5]. We use the complex-valued output to compare, the result is shown in Table XVI. The learning curve of machine learning is shown in Fig. 5.; targets and model outputs are shown in Fig. 6.; the MAPE of algorithms is shown in Table XIV, where and mean they only run 1 iteration; the profits from the simulation of investment strategy and the number of transactions are shown in Table XV.

(a)



(b)

1. Learning Curve (a) PSO-RLSE (b) ABCO-RLSE
2. Performance comparison in 10 trials (MAPE)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |  | |  | |
| APPLE | IBM | APPLE | IBM | APPLE | IBM | APPLE | IBM |
| 1 | 1.7973 | **0.8037** | 1.8350 | **0.7862** | 1.7901 | 0.8038 | 1.8661 | 0.8036 |
| 2 | 1.8238 | 0.8557 | **1.7927** | 0.8039 | 1.7936 | 0.8040 | 1.8147 | 0.8049 |
| 3 | 4.0664 | 0.8959 | 1.8690 | 0.8093 | 1.8113 | 0.8112 | 1.8786 | 0.8046 |
| 4 | **1.7970** | 0.8044 | 1.9528 | 0.8092 | 1.7981 | 0.8028 | 1.7937 | 0.8040 |
| 5 | 1.8541 | 0.8054 | 1.8690 | 0.8093 | 1.8124 | **0.7988** | 1.8316 | 0.8046 |
| 6 | 2.1120 | 0.8965 | 1.9528 | 0.8092 | 1.7918 | 0.8044 | 1.7941 | 0.8039 |
| 7 | 2.0104 | 0.7990 | 1.7937 | 0.8040 | 2.1383 | 0.7999 | 1.7937 | 0.8040 |
| 8 | 1.8238 | 0.8557 | 1.8690 | 0.8093 | **1.7837** | 0.8085 | 1.8170 | 0.8101 |
| 9 | 4.0664 | 0.8959 | 1.9528 | 0.8092 | 1.7900 | 0.8039 | **1.7830** | **0.8032** |
| 10 | 1.7970 | 0.8044 | 1.7937 | 0.8040 | 1.8466 | 0.8048 | 3.3550 | 0.9991 |

1. Simulation of investment (MAPE)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |  | |  | |
| Profits | Sell/buy | Profits | Sell/buy | Profits | Sell/buy | Profits | Sell/buy |
| 1 | 0.5314 | 268/223 | 0.3600 | 271/220 | 0.8300 | 268/223 | 0.1243 | 309/182 |
| 2 | 0.6843 | 265/226 | **0.6171** | 267/224 | 0.6886 | 267/224 | 0.6800 | 272/219 |
| 3 | **1.2971** | 266/225 | 0.4857 | 252/239 | 0.5314 | 268/223 | 0.3014 | 268/223 |
| 4 | 0.7257 | 266/225 | 0.5143 | 265/226 | **1.0629** | 267/224 | 0.4943 | 267/224 |
| 5 | 0.5971 | 265/226 | 0.4857 | 252/239 | 0.5657 | 267/224 | -0.1514 | 272/219 |
| 6 | 0.3200 | 269/222 | 0.5143 | 265/226 | 0.7743 | 266/225 | 0.7429 | 266/225 |
| 7 | 0.3929 | 264/227 | 0.6171 | 267/224 | 0.6713 | 268/223 | 0.4971 | 266/225 |
| 8 | 0.6845 | 265/226 | 0.4857 | 252/239 | 0.5657 | 267/224 | 0.4229 | 269/222 |
| 9 | 1.2971 | 266/225 | 0.5143 | 265/226 | 0.2814 | 269/222 | **0.9628** | 246/245 |
| 10 | 0.7257 | 266/225 | 0.6171 | 267/224 | 0.8600 | 263/228 | 0.3471 | 256/235 |
| Mean | 0.7256 | - | 0.5211 | - | 0.6834 | - | 0.4421 | - |



(a)

(b)



(c)



(d)

1. Targets and the model outputs (a) PSO-RLSE (APPLE) (b) PSO-RLSE (IBM) (c) ABCO-RLSE (APPLE) (d) ABCO-RLSE (IBM)
2. Performance comparison of multi-target (Experimentation 3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **MAPE** | | | | |
| Training phase | |  | Testing phase | |
| APPLE | IBM |  | APPLE | IBM |
| HiMMI [35] | - | - |  | 2.8373 | 1.2186 |
| ANN-GA-HMM-Interpolation [35] | - | - |  | 2.1649 | 1.0555 |
| ANN-GA-HMM-WA [35] | - | - |  | 1.9247 | 0.8487 |
| Bayesian ANN [36] | - | - |  | 1.9688 | 0.7441 |
| ARIMA [5] | - | - |  | 1.8009 | 0.9723 |
| PSO-RLSE (proposed) | **1.7840** | **0.8122** |  | **1.7970** | **0.8037** |
| ABCO-RLSE (proposed) | **1.7654** | **0.8043** |  | **1.7927** | **0.7862** |

# CONCLUSION

Through three experiments, it can be found that the model SCFNS proposed in this paper has the ability of multi-target prediction. After make multi-target feature selection, we can extract useful data from the original data according to different data and control the size of the data which will input into the model. As part of structure learning, input data can be automatically adjusted to the data, and different structures can be generated for different types of data. From the experiments, it can be found that by making time series prediction on four targets at a time, the effect of each target is as good as that proposed by other papers, or even better. It is proved that different data can be predicted effectively in this model. It also represents the algorithms in this study, which have a certain level and can work for multi-target prediction.

The part of machine learning may be limited by the characteristics of the algorithm itself, such as Particle Swarm Optimization (PSO) convergence fast, easy to fall into the local minimum, PSO algorithm in the first few rounds has been close to the last round of RMSE, or the large amount of data in the case, the need to search for more dimensions, the efficiency of the PSO will be poor, may restrict the overall performance of the model. For another algorithm Artificial Bee Colony Optimization (ABCO), if it reaches the setting limit, it will search new position to avoid falling into the local minimum, in contrast, is has higher probability to find the best solution, but the new position is calculated by the maximum and the minimum, the initial position of all bees is important, it will deeply affect the training of the model, it means that this algorithm is less stable. We can find that ABCO-RLSE is better than PSO-RLSE in ten trials, but the performance of PSO-RLSE method is not bad to other methods. In the future, we can use different algorithm to combine SCFNS to predict multiple targets, like Random Search [25], Ant Colony Optimization [28][30] and so on. Maybe we can overcome the current problem, we think the algorithm which has the concept of divide-conquer is a good idea, like CPSO [26].

In this study, neural network uses IF-THEN rules， the number of premises is same as the number of consequences， in the future, we can use different two number for them to increase the flexibility of model. Finally, in the part of simulation of investment strategy, we can understand that every algorithm combined with the model can make money, and we find two interesting things, first, we can observe the number of sell and buy has no relationship with profits, second, the lower RMSE not means the higher profits, but if the machine learning algorithm lacks of training, the profits will be apparently lower, we can sure that after compare the mean of the profits in ten trials. Through simulation of investment strategy, we know the investment event is variable, so we have to invest carefully on anything about money, maybe use different portfolio with machine learning in the future.

##### ACKNOWLEDGMENT

##### REFERENCES

1. A. J. Patton, “A review of copula models for economic time series,” Journal of Multivariate Analysis, vol. 110, pp. 4-18, 2012.
2. J. Zbontar and Y. LeCun, “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches,” Journal of Machine Learning Research, vol. 17, pp. 1-32, April 2016.
3. Z. Feng, L. Jin, D. Tao and S. Huang “DLANet A Manifold Learning based Discriminative Feature Learning Network for Scene Classification,” Neurocomputing, vol. 157, pp. 11-21, 2015.
4. D. Perez et al., “The 2014 General Video Game Playing Competition,” IEEE Transactions on computational intelligence and AI in games, in press, 2014.
5. C. Li and J.-W. Hu, “A new ARIMA-based neuro-fuzzy approach and swarm intelligence for time series forecasting,” Engineering Applications of Artificial Intelligence, vol. 25, pp. 295-308, 2012.
6. C. Li and T.-W. Chiang, ”Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets,” IEEE Transtraction on fuzzy systems. vol. 21.no. 3, pp.567-584, June 2013.
7. L. J. Herrera, H. Pomares, I. Rojas, A. Guillen, J. Gonzalez, M. Awad and A. Herrera, ”Multigrid-based fuzzy systems for time series prediction:CATS competition,” Neurocomputing, vol.70, pp. 2410-2425,2007.
8. I. Sugiarto and S. Natarajan, “Parameter estimation using least square method for MIMO Takagi-Sugeno neuro-fuzzy in time series forecasting,” J. Tek. Elektro, vol. 7(2), pp. 82-87, 2007.
9. M. Z.-Kermani and M. Teshnehlab, ”Using adaptive neuro-fuzzy inference system for hydrological time series prediction.” Appl. Soft Comput, vol. 8,pp. 928-936, 2008.
10. H. J. Rong, N. Sundararajan,G. B. Huang and P. Saratchandran, “Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction,” Fuzzy Set Syst, vol.157, pp. 1260-1275, 2006.
11. Zadeh, “Fuzzy sets”, Inf. Control , vol. 8, pp. 338-353, 1965.
12. D. Ramot, R. Milo, M. Friedman, and A, Kandel, “Complex fuzzy sets,” IEEE Trans. Fuzzy syst., vol . 10, no.2, pp. 171-186, Apr. 2002
13. C. Li, T.-W. Chiang. J.-W. Hu, and T. Wu, “Complex neuro-fuzzy intelligent approach to function approximation,” in Proc. 2010 3rd Int.Workshop Adv. Comput. Intell., pp. 151-156, 2010.
14. C. Li and T.-W. Chiang, “Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach,” Lect. Notes Artif. Intell., vol. 6592, pp. 242-251, 2011.
15. E. C. Shannon, ”A mathematical theory of communication,” Bell System Technical Journal, vol. 27, pp. 379-423, 1948.
16. C. Li , ”Multi-target feature selection,” unpublished, 2017.
17. T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applocations to modeling and control,” IEEE Trans. Systems, Man, and Cybernetics, vol. SMC-15, no. 1, pp. 116-132, 1985.
18. J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.
19. J. S. R. Jang, C. T. Sun and E. Mizutani, “Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence prentice hall,” Upper Saddle River, 1997.
20. C. Li and T.-W. Chiang, “Complex neuro-fuzzy self-learning approach to function approximation,” Lect. Notes Artif. Intell., vol. 5991, pp. 289-299, 2010.
21. J. S. R. Jang, “ANFIS: Adaptive-network-based fuzzy inference system,” IEEE Trans. Syst., Man, Cybern., vol. 23, no. 3, pp. 665-685, May/Jun, 1993.
22. D. S. Broomhead and D. Lowe, “Multivariable functional interpolation and adaptive networks,” Complex Syst., vol. 2, pp. 321-355, 1998.
23. A. J. Smola and B. Scholkopf , “A tutorial on support vector regression,” Static. Comput., vol. 14, no. 3, pp. 199-222, 2004.
24. N. I. Sapankevychand and R. Sankar, “Time series prediction using support vector machines: A survey,” IEEE Comput. Intell. Mag., vol.4, no. 2, pp. 24-38, May 2009.
25. Z. B. Zabinsky, ”Random search algorithms.” Technical report, University of Washington, Seattle, 2009.
26. F. van den Bergh and A.P. Engelbrecht, ”A cooperative approach to particle swarm optimization,” IEEE. Trans. On Evolutionary Computation, vol. 8, pp. 225-239, 2004.
27. H. Greenspan, B. van Ginneken and R. M. Summers, “Guest Editorial Deep Learning in Medical Imaging Overview and Future Promise of an Exciting New Technique,” IEEE Transactions on medical imaging, vol. 35, no. 5, May 2016.
28. A. Colomi, M. Dorigo and V.Maniezzo, “Distributed Optimization by Ant Colonies,” Proceedings of ECAL91 –European Conference on artificial life, pp. 134-142.
29. D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization artificial bee colony algorithm,” J Glob. Optim., vol. 39, pp. 459-471, 2007.
30. K. Socha and M. Dorigo, “Ant colony optimization for continuous domains,” European Journal of Operational Reaserch, vol. 185, pp. 1155-1173, 2008.
31. S.M. Chen, “Forecasting enrollments based on fuzzy time-series,” Fuzzy Sets Syst, vol. 81, pp. 311–319, 1996.
32. H. K. Yu, “Weighted fuzzy time-series models for TAIEX forecasting,” Phys. A, vol. 349, pp. 609–624, 2005.
33. R. F. Engle, “Autoregressive conditional heteroscedasticity with estimator of thevariance of United Kingdom inflation,” Econometrica, vol. 50, no. 4, pp. 987–1008, 1982.
34. L. Y. Wei, “A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting,” Applied Soft Computing, in press, 2016.
35. R. Hassan, B. Nath and M. Kirley, “A fusion model of HMM, ANN and GA for stock market forecasting,” ELSEVIER. Expert Systems with Applications, vol. 33, pp. 171-180, 2007.
36. L. Ticknor, “A Bayesian regularized artificial neural network for stock market forecasting,” ELSEVIER. Expert Systems with Applications, April, 2013.