Paper Title\* **(**use style: ***paper title***)

Subtitle as needed ***(paper subtitle)***

Chi Feng Lin

Information Management. National Central University

National Central University

Taoyuan, Taiwan

j8888888871@yahoo.com.tw

Authors Name/s per 2nd Affiliation (*Author*)

line 1 (of *Affiliation*): dept. name of organization

line 2-name of organization, acronyms acceptable

line 3-City, Country

line 4-e-mail address if desired

*Abstract*—Time series forecasting is a wide and important topic. We propose a sphere complex neuro-fuzzy system (SCNFS) for time series forecasting. If we use complex neuro-fuzzy system (CNFS), we can get only one complex-valued output. The imagine part and real part of output can direct to different targets. We improve the original complex neuro-fuzzy system in this study. The membership degrees are still complex-valued, but the model outputs can be more than one. It means that we can forecast two or more targets simultaneously. Related to the design of model, premises adapt Gaussian type sphere complex fuzzy set (SCFSs) and consequences adapt Takagi-Sugeno linear function, finally they are combined by aim object. Aim object makes model different from traditional IF-THEN rules, because premises and consequences can be different quantity. Otherwise, for optimizing the output of model. We use Particle Swarm Optimization (PSO) to optimize the parameters of premises, and the consequences use Recursive Least Squares Estimator (RLSE) for parameters optimizing. Finally, we make performance comparison with different methods by three experimentations.

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

# Introduction

Artificial Intelligence (AI) has many applications in real world, like financial forecasting [1], emergency treatment [2]- [3], enroll prediction [4], etc. Some of these applications are time series, time series is an important topic. It is because that time series application field is very wide in real world. In the past, many scholars promoted different methods for time series forecasting problems, like ARIMA [5], [6], fuzzy theory [11], neural network computing, neural fuzzy hybrid system [7]- [10], etc. Among them, neuro fuzzy hybrid system [7]- [10] is most often promoted.

Neuro-fuzzy hybrid systems have been widely investigated. Neural network systems have IF-THEN rules, these rules are similar to human experience rules. IF-THEN Rules can combine with fuzzy theory, it makes entire model structure much flexible, and we call it neuro-fuzzy hybrid system. Neuro-fuzzy hybrid system’s feature also make time series forecasting have not bad performance. So studies about time series forecasting almost adapt neuro network to implement model.

In this study, we adapt neuro network hybrid system method, we use IF-THEN rules for reference, and construct multi layers neural structure. For the flexibility of model, unlike traditional IF-THEN rules, we combine the premises with the consequences by Aim Object Layer, it means that we can have different quantity of the premises and the consequences. In model implementation respect, we combine fuzzy theory [11] with neural network system to construct neuro hybrid system. Premises adapt Gaussian type sphere complex fuzzy set (SCFSs), consequences adapt Takagi-Sugeno linear function [17], finally they combined by aim object. We expect that time series forecasting can be more precise through model and machine learning.

In 1965, Zadeh promoted fuzzy set concept [11], it makes data can be transferred to a membership degree which is between 0 to 1 by a function. In 2002, another scholar promoted a complex fuzzy set concept [12], membership degree can be a complex value, this concept makes membership degree can present in a unit disk of the complex plane (UDCP). Complex-valued membership degree is more abundant than real-valued membership degree. In complex neuro-fuzzy set (CNFS) [13], [14], we can get only one complex-valued output. The imagine part and real part of output can direct to different targets, so the limitation is two targets. Today there are many two targets forecasting papers have been introduced [6], [13], [14]. For forecasting more targets simultaneously, in this study, we improve original neuro-fuzzy system. We replace original complex fuzzy sets (CFSs) with SCFSs. In the model, membership degrees are also complex-valued, but it can present in multi-dimension space. Because SCFSs have more complex-valued outputs, we can forecast two or more targets simultaneously.

In this study, for using data efficiently, we preprocess the data set. We use 30 days up and downs to be the features and calculate the information contribute index of each target through Shannon Entropy [15]. Otherwise, we adapt concept of multi-target feature selection [16], then we can get some features to be our training data. Extracting the most important data not only decrease the model computing, but also improve performance efficiently. Finally, in the part of machine learning, we use well-known Particle Swarm Optimization (PSO) [18] and popular Recursive Least Square Estimator (RLSE) [19] to optimize the parameters, they are integrated into a method which we called PSO-RLSE method [20]. We use different algorithm to train the parameters of premises and consequences, we expect that using divide-and-conquer concept can decrease dimensions of searching, and it will make model much easier to find the best solution and improve entire performance.

# 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 number of data amounts; is number of data groups. Then make original data first difference, express as follows:

(1)

where is number of data amounts; is 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 variables, 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.

(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.

According to above influence information formulas, we can get influence information of each feature variable to per target variables. Then we can start to make multi-target feature selection, steps are given as follows:

Step 1: Computing 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 number of data. We combine the data in each input dimension, and use a function called «subclust()» which is offered by MATLAB to get the number of the clusters. After get the number of the clusters, we use fuzzy c-mean algorithm to cluster , and we can get Q cluster centers and standards, the cluster center in the output is denoted as }, the cluster standard in the output is denoted as {, }. Use the cluster centers and standards to make aim object, an aim object is connected with a consequence, consequence is Takagi-Sugeno type, which is defined as follows:

(21)

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

Aim object is for catching premises outputs, in this paper, we use SCFSs, so the outputs of the premises are complex-valued, it means that aim object has to be converted for make sure the outputs of aim object are complex-valued too. Covert formula is given as follows:

(22a)

(22b)

(22c)

where is the value of the output in the premise which shoot on the aim object; ; is converted center of the aim object; is converted width of the aim object, as follows:

(23a)

(23b)

(23c)

(24a)

(24b)

(24c)

where is variable of ; is variable of ; is the average of the target; is the standard of the target.

## Model Structure and I/O Relationship

There are 7 layers in neural network in this study. The training data set is denoted as , is number of data, is a input vector, is number of input dimensions; is a target vector, is 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:

(25)

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:

(26)

(27)

(28)

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

(29)

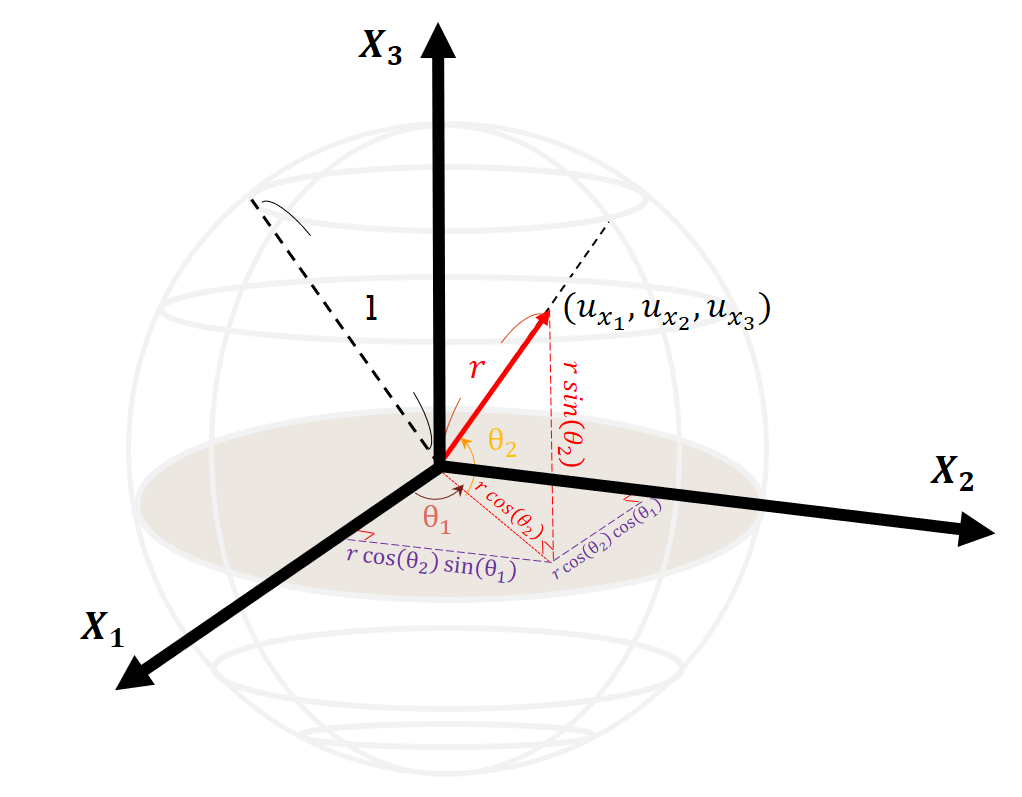
and the method of reducing dimensions, as follows:

(30)

(31)

(32)

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*  (33)

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:

(34a)

(34b)

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:

(35)

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

Layer 5: This layer is called aim object. Through the connection of the aim object, premises and consequences can be different quantity. In this paper, we use SCFSs, so the outputs of the premises are complex-valued, it means that aim object has to be converted (25)- (30) for make sure the outputs of aim object are complex-valued too. The outputs of this layer are given as follows:

(35)

where is the value of the output in the premise which shoot on the aim object; . Finally, we can get the Q outputs for the consequences.

Layer 6: This layer is called consequence layer. There are three inputs of this layer, they are outputs of the layer 1, the layer 4 and the layer 5. Through computing in this layer, we can get outputs , as follows:

(36)

(37)

where {} are the parameters of the consequence.

(38a)

R (38b)

(38c)

(38d)

(38e)

(38f)

Layer 7: This layer is called output layer. Combine the outputs and we can get model outputs.

(39)

## 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. 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:

(40)

(41)

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

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:

(42)

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:

(43a)

where:

(43b)

(43c)

(43d)

(43e)

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:

(44a)

(44b)

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:

(45)

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

Step 1: Prepare the training data.

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

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

(46a)

(46b)

(46c)

where .

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

Step 5: Calculate RMSE and update the pbest and the gbest of PSO.

Step 6: Repeat Step 2- 5 for all particle swarms until the iterations is over.

# Experimentation

## Example 1—Quadruple Time Series of Daily National Association of Securities Dealers Automated Quotation Composite Index

In this experimentation, we use the real-world time series data to testify the model performance, the data used is the daily opening price and the daily closing price of National Association of Securities Dealers Automated Quotation (NASDAQ) and the daily opening price and the daily closing price of Standard and Poor’s (S&P 500). The period of stock prices is from 3 Jan. 2007 to 20 Dec. 2010; data volume comes to 998. In order to compare with other papers, we use data which period is from 3 Jan. 2007 to 26 Dec. 2008 to be training data, data volume comes to 500, the rest of the data assumed as testing data, the testing data volume comes to 500. Note that proposed model has many complex-valued outputs, so it can forecast multiple targets simultaneously. In this experimentation, we use the daily opening price of NASDAQ to be real part of the first target, and use the daily closing price of NASDAQ to be imagine part of the first target; the daily opening price of S&P500 used to be real part of the second target and the daily closing price of S&P500 used to be imagine part of the second target.

1. Setting of SCNFS (Experimentation 1)

|  |  |
| --- | --- |
| **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 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 8 |
| Number of premise parameters | 44 |
| Number of aim object | 5 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 5 |
| Number of consequence parameters | 25 |

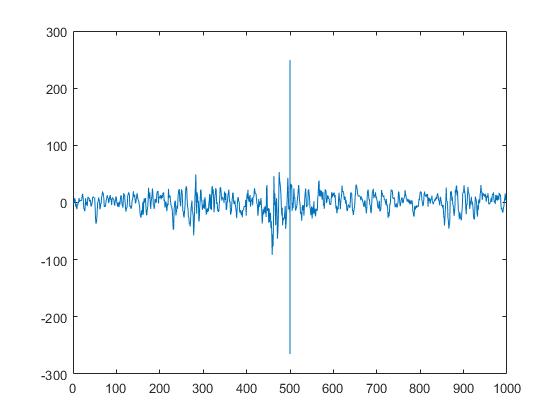
In the part of structure learning, we use the function “subclust()” which is offered from MATLAB to cluster the selected feature variables; parameter of the function is 0.2. Through premise selection, we extract 4 premises from original premises. The parameters after structure learning is shown in table I; the machine learning parameters of PSO-RLSE hybrid method are shown in table II.

1. Parameter setting of Machine Learning

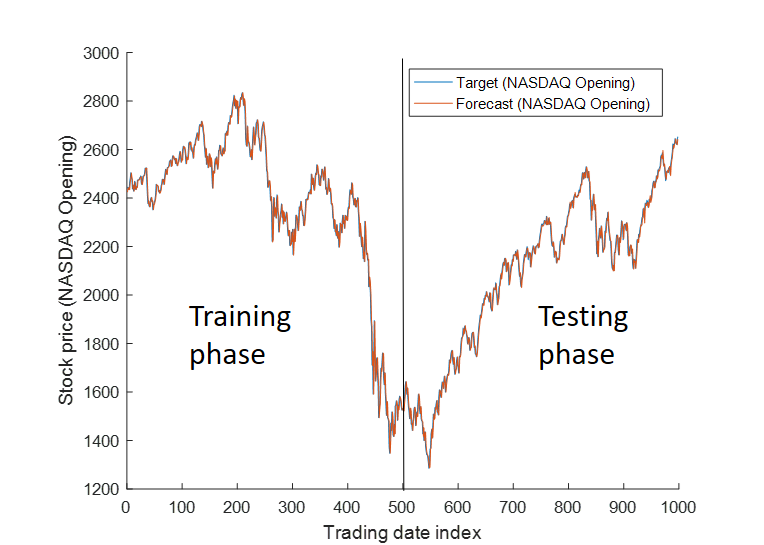
|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **RLSE** | |
|  |  |
|  | 25x1 zero vector |
|  | **I** |
| **I** | 25x25 identify matrix |

The result of this experimentation is compared with the methods of other papers, like ANFIS [21], CNFS-ARIMA [6], RBF network [22] and SVR [23] [24]. Except SVR, all the methods can forecast two real-valued targets simultaneously. So we use the first complex-valued output to compare with other papers, results are shown in table III. The learning curve of machine learning is shown in Fig. 2. The errors figure is shown in Fig. 3. The targets and the model outputs are shown in Fig. 4.

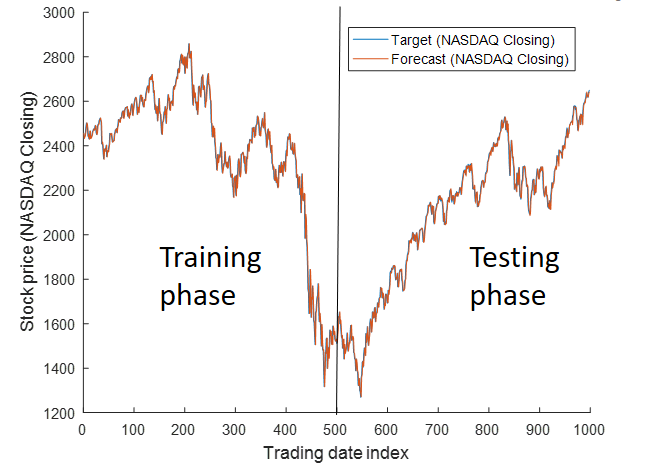


1. Learning curve (Experimentation 1)
2. Errors figure (Experimentation 1)
3. Performance Comparison in RMSE (NASDAQ Dual. Time Series)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| Opening index | Closing index |  | Opening index | Closing index |
| SVR (two models, each with single output) | 35.18 | 35.24 |  | 37.23 | 40.24 |
| ANFIS (two models, each with single output) | 37.83 | 38.66 |  | 38.80 | 42.36 |
| ANFIS (one model with two outputs) | 62.75 | 71.51 |  | 72.52 | 85.08 |
| RBF (two models, each with single output) | 37.59 | 33.89 |  | 37.52 | 44.08 |
| RBF (one model with two outputs) | 178.57 | 179.87 |  | 261.37 | 258.89 |
| CNFS(5)-ARIMA (one model with two outputs) | 21.56 | 20.81 |  | 32.52 | 33.70 |
| SCNFS(proposed) | 37.83 | 37.86 |  | 27.67 | 27.75 |



(a)



(b)

1. NASDAQ real values and model outputs (a) opening price (b) closing price
2. Ten Trials Performance (Experimentation 1)

|  |  |  |
| --- | --- | --- |
|  | Performance(RMSE) | |
| Triers | Opening index | Closing index |
| 1 | 27.67 | 27.75 |
| 2 | 29.62 | 28.04 |
| 3 | 33.85 | 33.99 |
| 4 | 28.09 | 28.78 |
| 5 | 27.73 | 27.44 |
| 6 | 28.09 | 28.77 |
| 7 | 27.68 | 27.61 |
| 8 | 29.35 | 27.93 |
| 9 | 27.67 | 27.91 |
| 10 | 27.72 | 27.76 |

## Example 2—Quadruple Time Series of Daily Dow Jones Industrial Average Index

In this experimentation, we also use real-world time series data to testify the model performance. But it is different from experimentation 1, the relationship of the targets is not between closing price and opening price, it means that the curve of the targets looks not similar. Used data of this experimentation are the closing stock price of The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX), Dow Jones Industrial Average Index (DJIA), National Association of Securities Dealers Automated Quotation (NASDAQ), Standard & Poor's 500 (S&P500). In order to compare with other papers, we use the closing price from 2001 to 2004 to forecast, the data volume of 2001 year comes to 245; the data volume of 2002 year comes to 248; the data volume of 2003 year comes to 249; the data volume of 2004 year comes to 250. We forecast four targets for every year. In each year, we use the data of earlier ten months to be training data, and the rest of the data is testing data. About the training data, 2001 year comes to 181, 2002 year comes to 184, 2003 year comes to 185, 2004 year comes to 185. Note that proposed model has lots of complex-valued outputs simultaneously, so it can forecast multiple targets. In this experimentation, real part of the first complex-valued target is the daily closing price of TAIEX; imagine part of the first complex-valued target is the daily closing price of DJIA; real part of the second complex-valued target is the daily closing price of NASDAQ; imagine part of the second complex-valued target is the daily closing price of S&P500.

In the part of structure learning, we use a function «subclust()» which is offered by MATLAB to cluster the selected feature variables; parameter of the function is 0.2. The parameters after structure learning is shown in table IV to table VII; the machine learning parameters of PSO-RLSE hybrid method are shown in table VIII.

1. Setting of SCFNS (2001 year)

|  |  |
| --- | --- |
| **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 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises | 10 |
| Number of premise parameters | 48 |
| Number of aim object | 3 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 3 |
| Number of consequence parameters | 15 |

1. Setting of SCNFS (2002 Year)

|  |  |
| --- | --- |
| **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 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises | 13 |
| Number of premise parameters | 48 |
| Number of aim object | 3 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 3 |
| Number of consequence parameters | 15 |

1. Setting of SCNFS (2003 Year)

|  |  |
| --- | --- |
| **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 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises | 11 |
| Number of premise parameters | 48 |
| Number of aim object | 4 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 4 |
| Number of consequence parameters | 20 |

1. Setting of SCNFS (2004 Year)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | { } |
| Number of fuzzy sets of each input | 1, 3, 3, 3 |
| Number of targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises | 15 |
| Number of premise parameters | 40 |
| Number of aim object | 3 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 3 |
| Number of consequence parameters | 15 |

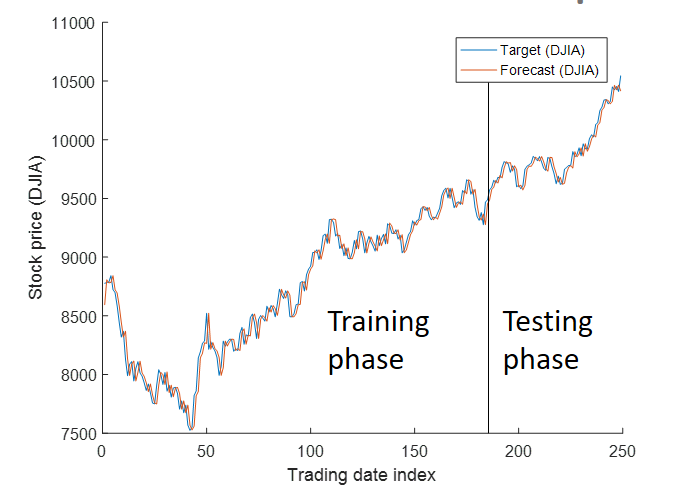
1. Parameters Setting of Machine Learning (2003 year)

|  |  |
| --- | --- |
| **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 |
| **RLSE** | |
|  |  |
|  | Given by Number of consequence parameters |
|  | x1 zero vector |
|  | **I** |
| **I** | identify matrix |

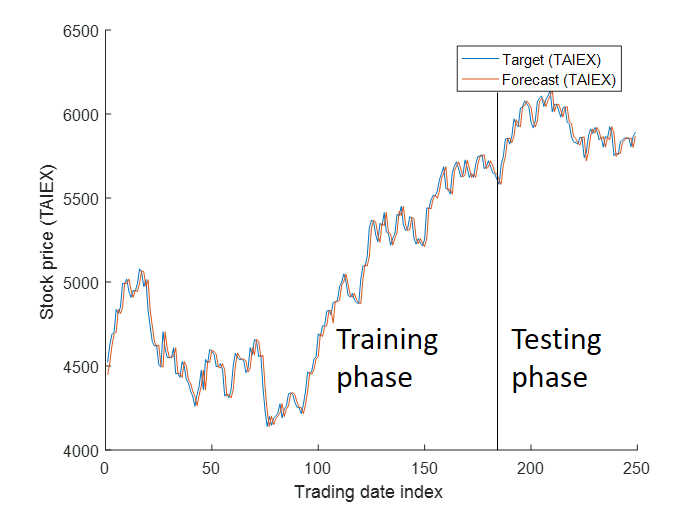
The result of this experimentation is compared with the methods of other papers, like ANFIS [21], CNFS-ARIMA [6], RBF network [22] and SVR [23] [24]. Except SVR, all the methods can forecast two real-valued targets simultaneously. So we use the first complex-valued output to compare with other papers, results are shown in table IX and table X. The targets and the model outputs are shown in Fig. 5. The errors figure is shown in Fig. 6.

1. Ten Trials Performance (Experimentation 2)

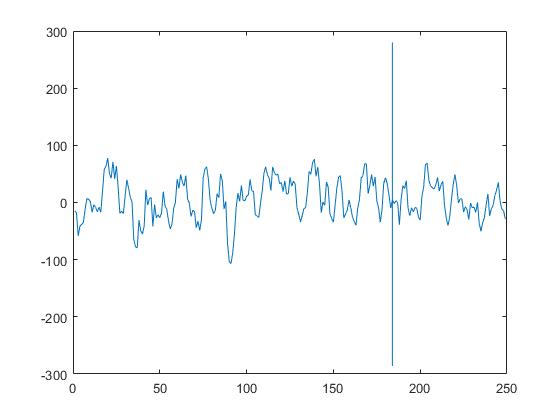
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Performance(RMSE) | | | |
| triers | 2001 | 2002 | 2003 | 2004 |
| 1 | 259.65 | 279.40 | 197.69 | 279.28 |
| 2 | 278.12 | 279.35 | 197.27 | 276.41 |
| 3 | 260.49 | 279.60 | 197.23 | 278.93 |
| 4 | 263.14 | 280.96 | 198.83 | 279.94 |
| 5 | 259.93 | 279.61 | 196.77 | 280.14 |
| 6 | 259.99 | 280.91 | 197.18 | 276.07 |
| 7 | 259.54 | 280.50 | 195.96 | 279.47 |
| 8 | 261.27 | 280.10 | 197.03 | 280.19 |
| 9 | 286.01 | 280.76 | 198.36 | 276.35 |
| 10 | 259.44 | 282.01 | 195.45 | 278.65 |



(a)



(b)

1. Real values and model outputs (a) DJIA (2003 year) (b) TAIEX (2003 year)
2. Errors figure (Experimentation 2)
3. Performance Comparison in RMSE (DJIA Time Series)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Year** | **2001** | **2002** | **2003** | **2004** |
| SVR (two models, each with single output) [23] [24] | 101.44 | 117.95 | 82.76 | 71.49 |
| ANFIS (two models, each with single output) [21] | 105.56 | 111.69 | 72.09 | 68.00 |
| ANFIS (one model with two outputs) [21] | 128.20 | 142.05 | 90.37 | 83.69 |
| RBF (two models, each with single output) [22] | 106.33 | 131.24 | 97.58 | 81.79 |
| RBF (one model with two outputs) [22] | 181.79 | 136.28 | 154.14 | 148.11 |
| CNFS(5)-ARIMA (one model with two outputs) [6] | 103.06 | 103.42 | 70.70 | 66.55 |
| SCNFS(proposed) training phase | 91.95 | 98.41 | 70.23 | 100.94 |
| SCNFS(proposed) testing phase | 99.86 | 89.52 | 57.21 | 63.48 |

1. Performance Comparison in RMSE (TAIEX Time Series)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Year** | **2001** | **2002** | **2003** | **2004** |
| SVR (two models, each with single output) [23] [24] | 162.46 | 67.72 | 59.47 | 58.81 |
| ANFIS (two models, each with single output) [21] | 147.36 | 70.17 | 72.61 | 65.33 |
| ANFIS (one model with two outputs) [21] | 151.62 | 78.27 | 81.69 | 70.54 |
| RBF (two models, each with single output) [22] | 134.32 | 65.15 | 60.41 | 102.86 |
| RBF (one model with two outputs) [22] | 137.58 | 78.54 | 115.92 | 126.48 |
| CNFS(5)-ARIMA (one model with two outputs) [6] | 115.82 | 64.34 | 57.69 | 55.56 |
| SCNFS(proposed) training phase | 90.58 | 100.94 | 68.15 | 98.90 |
| SCNFS(proposed) testing phase | 87.92 | 87.25 | 55.50 | 59.95 |

## 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 forecast multiple targets simultaneously. In this experimentation, we use the daily closing price of IBM to be real part of the first target, and use the daily closing price of APPLE to be imagine part of the first target; the daily closing price of Dell used to be real part of the second target, and the daily closing price of Microsoft used to be imagine part of the second target.

In the part of structure learning, we use the function «subclust()» which is offered by MATLAB to cluster the selected feature variables; parameter of the function is 0.3. Through premise selection, we extract 15 premises from original premises. The parameters after structure learning is shown in table XI; the machine learning parameters of PSO-RLSE hybrid method are shown in table XII.

1. Setting of SCNFS

|  |  |
| --- | --- |
| **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 |
| Number of aim object | 3 |
| Type of consequences | Takagi-Sugeno |
| Number of consequences | 3 |
| Number of consequence parameters | 15 |

Table XIV

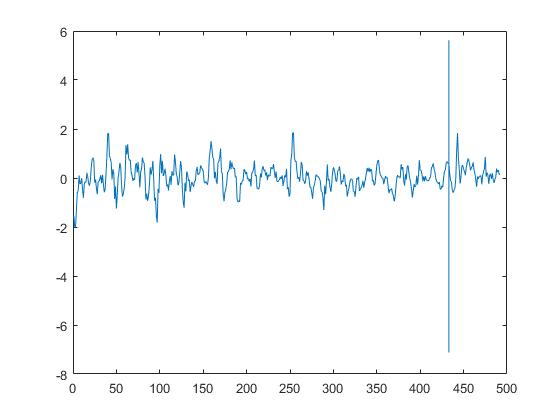
Parameters setting of machine learning

|  |  |
| --- | --- |
| **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 |
| **RLSE** | |
|  |  |
|  | 15x1 zero vector |
|  | **I** |
| **I** | 15x15 identify matrix |

The proposed models are compared with many other approaches in the literature, like HiMMI [27], ANN-GA-HMM-Interpolation [27], ANN-GA-HMM-WA [27], ARIMA [5], Bayesian ANN [28]. So we use the first model output and the real-valued part of the second model output to compare with other studies, the result is shown in Table XIII. The learning curve of model is shown in Fig. 7, targets and model outputs are shown in Fig. 9. The errors figure is shown in Fig. 8.

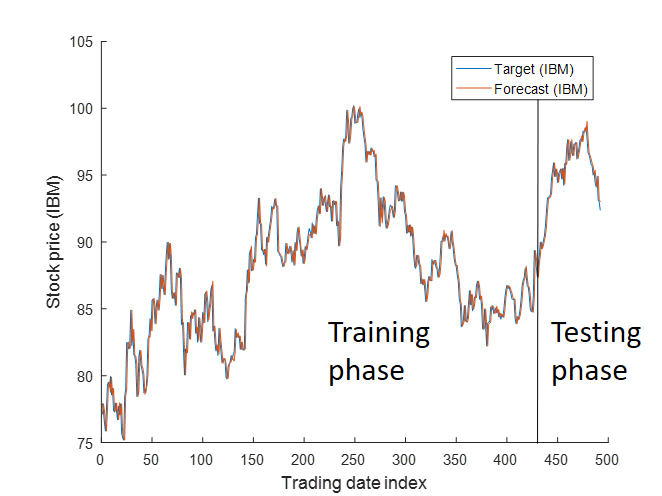
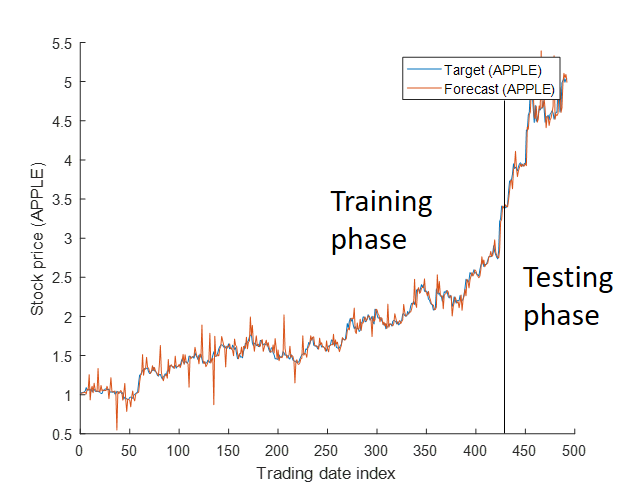
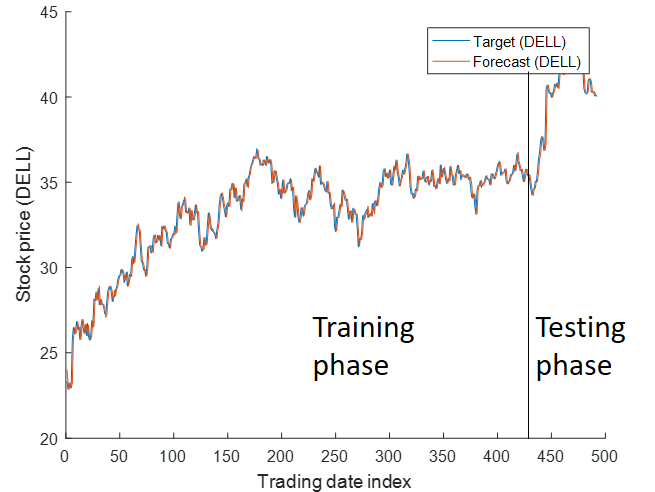


1. Learning curve (experimentation3)



1. Errors figure (Experimentation 3)
2. Ten Trials Performance (Experimentation 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Performance | | | |
| triers | APPLE | IBM | DELL | Microsoft |
| 1 | 0.6627 | 0.6627 | 0.6654 | 0.6522 |
| 2 | 0.6718 | 0.6638 | 0.6603 | 0.6516 |
| 3 | 0.6655 | 0.6589 | 0.6764 | 0.6718 |
| 4 | 0.6629 | 0.6604 | 0.6646 | 0.6526 |
| 5 | 0.6668 | 0.6618 | 0.6638 | 0.6511 |
| 6 | 0.6627 | 0.6631 | 0.6670 | 0.6629 |
| 7 | 0.6655 | 0.6617 | 0.6693 | 0.6553 |
| 8 | 0.7070 | 0.6834 | 0.6494 | 0.6504 |
| 9 | 0.6706 | 0.66057 | 0.6511 | 0.6558 |
| 10 | 0.6587 | 0.6528 | 0.6628 | 0.6530 |

 (a)  (b)  (c)

1. Real value and model output (a) IBM closing price (b) APPLE closing price (c) DELL closing price
2. Performance Comparison in MAPE

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Company** | **APPLE** | **IBM** | **DELL** |
| HiMMI [27] | 2.8373 | 1.2186 | 1.0117 |
| ANN-GA-HMM-Interpolation [27] | 2.1649 | 1.0555 | 0.8446 |
| ANN-GA-HMM-WA [27] | 1.9247 | 0.8487 | 0.6992 |
| Bayesian ANN [28] | 1.9688 | 0.7441 | - |
| ARIMA [5] | 1.8009 | 0.9723 | 0.6604 |
| SCNFS (proposed) training phase | 0.7828 | 0.7883 | 0.7833 |
| SCNFS (proposed) testing phase | 0.6718 | 0.6638 | 0.6603 |

# Conclusion

Through three experimentations, we can find that proposed model SCFNS indeed has ability of multi-target forecasting. After multi-target feature selection, we can extract useful data from original data, even if data is different in every experimentation. Otherwise we can also control the input size of the model via multi-target feature selection. In the part of structure learning, model can adjust the parameters automatically by input data, it means that it can construct different structure when it encounters different data. Through experimentation, we can find that time series forecasting of four targets is not bad, its performance of each target is even better than other studies. This result fully proves that different data can be forecasted effectively in this model, it means that PSO-RLSE hybrid method is up to par. Up to now model for multi-target time series forecasting can work, in the part of machine learning may be limited by feature of PSO, it converges prematurely and it is not easy to escape local optima. It is not difficult to find that RMSE of earlier iterations is almost the same as the best RMSE in figure 2. If input data volume is huge, the searching dimension will increase, it makes the result of PSO be ineffective, this situation may restrict the entire performance of model. In the future, for overcoming the problems of PSO, we can adapt different machine learning to combine SCFNS, like Random Search [25], CPSO [26], etc. Finally, proposed model in this study is different from traditional IF-THEN rules. Through connection of aim object, IF-part and THEN-part can’t be one-to-one relationship, despite this, performance of model still achieves the desired result. This change makes model more flexible, and we can increase more hidden layers 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. S. S. Jones, R. S. Evans, T. L. Allen, A. Thomas, P. J. Haug, S. J. Welch and G. L. Snow, “A multivariate time series approach to modeling and forecasting demand in the emergency department,” Journal of Biomedical Informatics, vol. 42, pp. 123-139, 2009.
3. P. Aboagye-Sarfo, Q. Mai, F. M. Sanfilippo, D. B. Preen, L. M. Stewart and D. M. Fatovich, “A comparison of multivariate and univariate time series approaches to modeling and forcasting emergency department demand in Western Australia,” Journal of Biomedical Informatics, vol. 57, pp. 62-73, 2015.
4. S.-T. Li and T.-C. Cheng, “Deterministic fuzzy time series model for foercasting enrollments,” Computers and Mathematics with Application, vol. 53, pp. 1904-1920, 2017.
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. 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.
28. L. Ticknor, “A Bayesian regularized artificial neural network for stock market forecasting,” ELSEVIER. Expert Systems with Applications, April, 2013.