A Novel Intelligent Time Series Forecasting Approach Using Sphere Complex Fuzzy Sets

Chi Feng Lina and Chunshien Lib

Information Management. National Central University, National Central University, Taoyuan, TAIWAN

aEmail: j8888888871@yahoo.tw

bEmail: [jamesli@mgt.ncu.edu.tw](mailto:jamesli@mgt.ncu.edu.tw) (corresponding author)

*Abstract*—The prediction of time series is a widely used and important research topic. The artificial intelligence models are currently being widely used in this topic, such as neuro-fuzzy systems. This paper proposes a new type of fuzzy sets, namely Sphere Complex Fuzzy Sets (SCFSs), which are applied to neuro-fuzzy system model and multi-objective time series prediction. Using SCFSs, the neuro-fuzzy model proposed in this paper has the capability of multiple complex-valued outputs. Every output can have real and imaginary parts for two different real-valued targets, respectively. With regard to feature selection, this study used multi-target feature selection to screen out favorable features for all objectives and use these features as a model input to reduce the overall computational loading on the model and improve data utilization efficiency. In terms of models, a multi-layer neural network architecture is used and includes an input layer, an SCFS neural layer, a premise neural layer, an aim object neural layer, a T-S neural layer (Takagi-Sugeno neural) and output layer, where, the aim object neural layer is an innovative model construction of this paper. Its purpose is to make the data driven model construction have structural adaptability and application flexibility. In machine learning, we use the divide-and-conquer principle when training the model. The parameters of the SCFS neural layer are optimized using Particle swarm optimization; the parameters of the T-S neural layer are optimized using recursive least squares estimation (RLSE); other neural layers not need to be optimized. In the multi-target time series forecasting, we designed three experiments to test the experimental results and compare performance with different methods. Although most of the literature methods aim at single target or a few methods for double-target time series prediction, the multi-target prediction method proposed in this paper shows excellent performance through performance comparison.

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

# Introduction

Time series data is a collection that uses time series as the sorting basis, such as stock market, exchange rate, energy consumption and so on. The time series can be regarded as data observed in a discrete time sequence and can be analyzed by mathematical methods or statistical models. With the development of present information technology, both the Internet of Things and the information brought by the Internet, these data have grown explosively, resulting in a quantitative change in the amount of data. These data can no longer be dealt with by human intelligence. We need to use machine learning or data mining to find out the hidden relationships or rules hidden in the data in order to discover the value.

Time series is a very important research topic. The prediction of time series data is widely used in many fields such as medicine, energy planning, financial forecast and so on. In the medical field, Wei et al. [28] predicted the number of daily patients through a mixed AR-EMD hybrid model for resource allocation, Osthus et al. [31] predicted seasonal infectious diseases and helped in public health planning and outbreak response, S. S. Jones et al. [16] used multivariate analysis to fit the time series model to predict the number of emergency patients so that hospitals could better allocate hospital resources, with regard to the number of emergency patients, Aboagye-Sarfo et al. [3] also analyzed and proposed a vector-autoregressive moving average (vector-ARMA) model to compare the differences between multivariate and univariate; in the area of energy prediction, Camelo et al. [9] used autoregressive integrated moving average with explanatory variable (ARIMEX) and Holt-Winters (HW) to combine with neural networks to forecast wind power generation, besides this national level application, Alobaidi et al. [1] developed a new framework specifically for forecasting household electricity consumption; in financial forecasting, in 2012, Li et al. [27] used the autoregressive integrated moving average (ARIMA) model and the neuro-fuzzy system (NFS) to predict the closing price of stocks, in the next year, Li et al. [23] used a neuro-fuzzy system to predict two targets simultaneously; in 2017, Koijen et al. [20] predicted each other's data through stocks and bonds; Pan et al. [32] compare short-term and long-term leverage by predicting the volatility of stocks.

In all fields, it is most difficult to predict financial time series, because it contains many factors, such as the company's operating conditions, international situation and so on. In financial predictions, many methods have been proposed [14][21][23][27][35][37], such as ARIMA[23][27], fuzzy theory, neural network computation, neuro-fuzzy hybrid systems and so on. Among them, neuro-fuzzy systems (NFSs) [14][21][35][37] are the most commonly proposed. Neuro-fuzzy systems are fuzzy systems that use learning algorithms derived from or inspired by the neural network theory [30] to determine their parameters (fuzzy sets and fuzzy rules) by processing data samples. The neural network was created by McCulloch et al. [30] in 1943 based on mathematics and an algorithm called threshold logic, a neural network model in which neurons can receive information to calculate whether or not to produce excitatory response. In 1956, Rochester et al. [33] combined this mathematical model with the Hebb's Law [15] to create a perceptron to simulate the human brain, but fewer perceptron neurons, and its transmitted signals with a weight of 0 or 1, This method still cannot deal with the XOR problem (non-linear problem), and at that time the computer did not have enough ability to calculate. Until 1975, Werbos [42] proposed a back-propagation (BP) algorithm. This algorithm effectively solved the XOR problem and the problem of training multi-layer neural networks. Afterwards, many other methods of artificial intelligence were proposed, such as support vector machines [10], Bayes classifier [38] and so on. Until recently, due to the rapid development of information technology, the computation speed has increased, it leads to artificial intelligence in certain areas has surpassed the human level, and neural networks have been heavily studied again.

Neuro-fuzzy systems are a combination of neural networks and fuzzy theory [44]. Neuro-fuzzy systems are usually represented as multi-layer feed-forward neural networks, such as ANFIS [18]. The characteristics of neuro-fuzzy systems make it have a good effect on the prediction of time series. Therefore, most of the current research in the area of time series prediction uses a neuro-fuzzy system as model structure.

This study adopts a neuro-fuzzy system and follows the IF-THEN rules of the expert system to construct a multi-layer neuron architecture. In order to increase the flexibility of the model, it is different from the traditional IF–THEN rule method, in this study, the aim object neural layer is used to connect with the neural network layer, which makes the number of neurons of each layer can be different. In terms of the model, we combine fuzzy theory with the neural network to form neuro-fuzzy system, the sphere complex fuzzy sets (SCFSs) and Takagi-Sugeno linear function are used, and the two are combined through the aim object layer. Through this model and machine learning we expect that the prediction of time series can be more accurate than other literature.

In 1965, the concept of fuzzy sets was proposed [44] by Zadeh. The purpose was to make human ideas become mathematic formula. Through the concept of fuzzy sets, elements can belong to a certain set and the membership degree is between 0 and 1. Afterwards, Buckley proposed the concept of fuzzy complex numbers [5] – [7], In 2002, Ramot et al. [34] proposed complex fuzzy sets (CFSs), 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 membership degree more abundant. We can obtain the complex-valued output through the complex neural fuzzy set system (CNFS) [24][26]. The real part and the imaginary part can be used to predict different targets respectively, so we can predict two targets simultaneously. At present, there are a lot of research of dual-target prediction [23][24][26]. In order to forecast for more targets at the same time, this paper improves the original neuro-fuzzy set system and changes the complex fuzzy sets (CFSs) to SCFSs. The membership degree is presented in the three–dimensional space. Through projecting numbers in different dimensions and combining multiple complex-valued, the membership degree is more abundant, and it is also possible to predict multiple targets at one time.

In order to make the data more effective for use, many scholars have proposed methods to deal with the data in data preprocessing. In time series data processing, Mikalsen et al. [29] deal with missing values in data through clustering and machine learning. Zhang et al. [43] proposed refined composite multiscale weighted-permutation entropy (RCMWPE) to improve the original multiscale weighted-permutation entropy (MWPE) to enable more accurate use of data. Alves et al. [2] proposed a completely new chaos description through theories in periodic, chaotic, and stochastic systems. In this study, we use the concept of Shannon entropy [39] to calculate feature’s individual contribution to targets. In addition, we adopted the concept of multi-target feature selection [22] 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 raw data can not only reduce the model computational burden, but also effectively increase the performance of the prediction. In the machine learning section, we use the popular particle swarm optimization (PSO) [19] and the well-known recursive least square estimation (RLSE) [17] to optimize parameters, and integrate the two algorithm called the PSO–RLSE method [25]. In this study, different parts of the parameters are trained through different algorithms. The divide-and-conquer principle is used to reduce the search dimension of the algorithm, which makes it easier for the model to find the best solution and improve overall performance.

The rest of this paper is as follows. In Section II, a series of research methods will be described in detail: SCFSs, multi-targets feature selection, structure learning, model input and output, and machine learning. In Section III, three examples for time series forecasting are given to test the proposed approach. Experimental results are given and compared with other papers. In Section IV, we discuss the contents of the data and the experimental results. Finally, in Section V, the contributions of the research and future development will be summarized.

# Methodology

## Sphere Complex Fuzzy Sets

In the past, the concept of fuzzy sets can derive the one-to-one membership degree of elements to a set. The complex fuzzy sets [34] can have a complex-valued membership degree, but in order to make the application more widespread, we hope to make the membership degree more abundant through one idea.

The SCFS is an innovative concept proposed in this paper. Through this concept, one piece of data can be converted into multiple complex-valued membership degrees, so that the model can predict multiple targets at a time. First, place the membership degree of Gaussian function in a SCFS with a radius of 1 (Fig. 1), and a set of spatial vectors can be obtained , whose components are expressed as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |

where, is the membership degree of Gaussian function (25); ; . With the dismantling of , at least four groups of complex-valued membership degrees are available, including the following membership degrees.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |

where, 。

1

1. Sphere complex fuzzy set. is membership degree of Gaussian function. The projection in each dimension can be calculated by the angles and . A SCFS and its attribution information is carried by a spherical space vector, which will change with the input.

## Multi–Target Feature Selection

Feature selection can not only eliminate negative information, but also help reduce the computational burden on the model. Therefore, it is an important part of data preprocessing. When faced with multiple targets, feature selection requires more careful handling to bring positive results. This paper predicts multiple targets at the same time. The concept of the Shannon information entropy [39] is used, and the multi-target feature selection method [22] is used to select the suitable features. Finally, training data are obtained from the selected features.

The term entropy was first proposed by the German physicist Clausius in 1854 [12]. It is a measure of the disorder of physical systems. In 1948, Shannon proposed the concept of information entropy [39]. Entropy is defined as the amount of uncertainty of information content. If the randomness of information is higher, the information entropy will be higher. For a random variable , the information entropy [39] is defined as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is entropy of random variable is the probability density of occurrence of event is entropy of .

If bigger than 1, the part will be negative and effect the overall expected value, so we made some changes to the formula as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

where, is a very small positive value.

The selection of our features is aimed at the target. Therefore, we use the concept of information entropy to calculate the influence information between each feature and the target [22]. The formula is as follows.

|  |  |
| --- | --- |
| , |  |

where, is the influence information of the random variable to the random variable is the random variable when event value is positiveis the random variable when event value is negative; is the mutual information of the random variable and the random variable ; is the mutual information of the random variable and the random variable , mutual information is defined as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

where, is the entropy of the target; is the entropy of the random variable when event is positive; is the entropy of the random variable when event is negative; The conditional information entropy formula is as follows.

|  |  |
| --- | --- |
| , |  |
| , |  |
| , |  |
| , |  |

where, is the variable when event is positive; is the variable when event is negative; is the probability density when event is positive; is the probability density of when event is positive; is the probability density when event is negative; is the probability density of when event is negative.

Through the above formulas, we can obtain the influence information of each feature variable for each target. To facilitate the calculation of the influence information, we can combine all the feature variables and targets into a data matrix (DM), as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the candidate feature, , is the total number of data for a feature, is the number of features; is the target variable, , is the number of targets.

We use the feature data of each row in the data matrix to calculate the influence information to other rows. The influence information matrix (IIM) is sorted through the influence information between the feature, another feature and the target. is expressed as follows.

|  |  |
| --- | --- |
| , |  |

where, is the feature variable; is the target variable; is the number of features; is the influence information of the feature variable to the feature variable , and .

Multi-target feature selection is based on the influence information in the IIM. The steps are as follows.

Step1: Calculate the selection gain of the feature to the target denoted as , where, is the feature variable; is the target. Selection gain formula is as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the influence information of the feature variable to the target variable ; is the selected pool, ; is the element in the selected pool; is the redundancy information of the to the existed features in . Redundancy information formula is as follows.

|  |  |
| --- | --- |
| , |  |

where, is the number of features in the selected pool; is the influence information of to the feature variable in ; is the influence information of the feature variable to in . After the above calculation, if is bigger than 0, the feature variable will be added in to the selected pool .

Step2: Whether overlapped or not, the feature variables that exist in all selected feature pools are recorded and denoted as Ω. Where, is the number of targets; , is the feature variable in . Calculate the number of times each feature appears in all selected feature pools, denoted as .

Step3: The covering rate can be calculated by as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

calculate , the mean of .

Step4: To accumulate the selection gain of features in each selected pool.

|  |  |
| --- | --- |
| , |  |

calculate , the mean of .

Step5: The effective contribution ρ of the feature can be calculated by the accumulated selection gain and covering rate .

|  |  |  |
| --- | --- | --- |
|  | , |  |

Step6: Test all feature variables in , if , then .

Step7: Set upper and lower limits as and , respectively. Find through the upper and lower limits. is the number of the final selected features. If is between and , then set as ; if is smaller than , then set as ; if is bigger than , then set as .

Step8: After sorting of , select top feature variables into final pool (FP) as the result of multi-target feature selection.

## Structure Learning

Structural learning is to apply the training data more logically to model construction. In addition, the results of structural learning will also become part of the subsequent parameter learning. In this study, the training data of different input dimensions are clustered by the subtractive clustering algorithm [11]. The cluster center and standard deviation after clustering are used to form a fuzzy set. The sum of the number of fuzzy sets in each dimension is the number of the first layer of neurons. This study uses a Gaussian type fuzzy set, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the input variable, and are the parameters of cluster center and standard deviation, respectively.

We can form areas based on the fuzzy sets of each input dimension.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the number of clusters in the input dimension, which is the number of fuzzy sets. For example, the composition of the area is as follows.

|  |  |
| --- | --- |
| , |  |

, where, is the input linguist variable; is the  input variable, ; is the fuzzy set of the input linguist variable in the area, which is constructed by Gaussian function (25).



1. Fuzzy sets input space (2-dimension). Two input dimensions, each divided into 3 groups will form a total of 9 areas, where the z-axis is the data density of the area.



1. Sum of data density. It can be seen that the sum of the data density in some areas are higher, which means that it is more advantageous to build models.

For the efficiency of the model, as well as reducing the computational burden, we filter out several more important areas, which will become the number of neurons in the second layer. If we take two input dimensions as an example, we can get the area shown in Fig. 2. Then, through the concept of data density, the data will be sprinkled into each area and the data density will be calculated. After accumulating the data density, the figure can be obtained (Fig. 3), from which the higher density data area can be selected as the Layer 2 neurons. Detail steps are as follows.

Step1: The data density of each area can be obtained from each input dimension and fuzzy set. Taking the block as an example, the data density formula is as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the data for the input dimension; is the fuzzy set of the input dimension in the area.

Step2: Accumulate data density in each area as , formula is as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the total number of the data. Compute the , mean of , standard deviation denoted as .

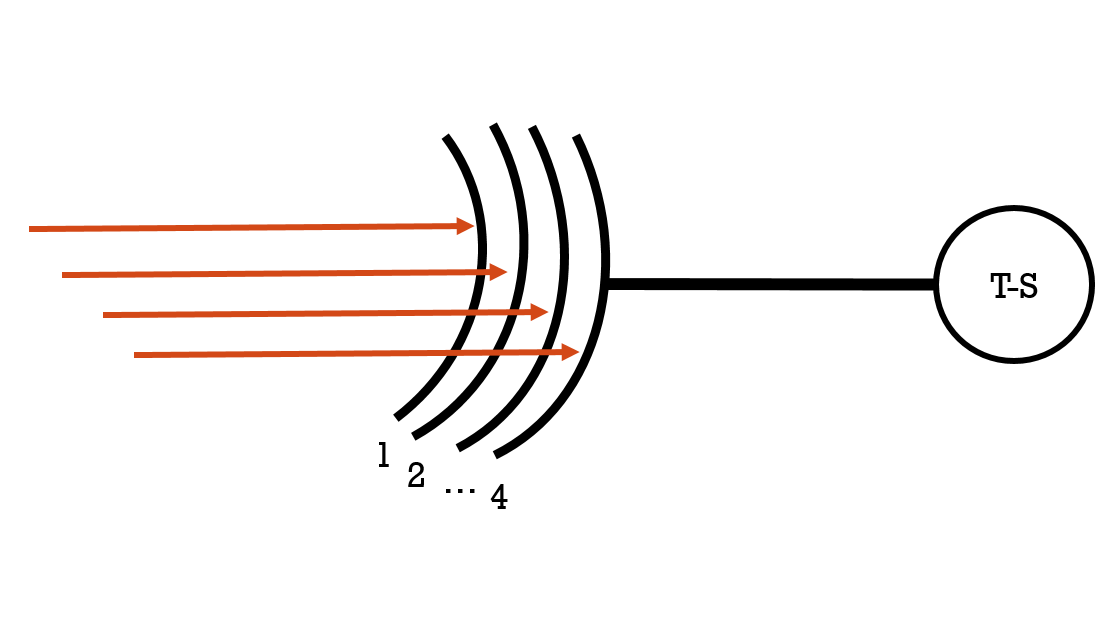
Step3: Check each area, if , then . Set lower and upper limits as and , respectively. Find through the upper and lower limits. is the number of the final selected areas. For all experiments in this study, was set to 15 and was set to 4. If is between lower and upper limits, then is set to ; if is smaller than the lower limit, then is set to ; if is bigger than the upper limit, then is set to .

Step4: After sorting , select top areas as the neurons of the second layer.

The number of neurons in the third layer is determined by the subtractive clustering [11]. After clustering, the relevant values of the neurons in the third layer (aim object layer) are determined by fuzzy c-mean clustering [36]. The steps are as follows.

Step1: Cluster input data set by subtractive clustering [11], then determine , which is the number of the neurons of the third layer (aim object layer).

Step2: After determining the number of clusters, cluster the target set by fuzzy c-mean algorithm to get centers and standard deviations.



1. Multi-layer aim object. An aim object can receive multiple inputs at one time, and each aim object connects with one T-S neuron.

Step3: Since this study uses SCFS, the aim object will receive the output of the previous layer, so each aim object has to own many layers to receive values of the input vector, as shown in Fig. 4. Where, the center of the cluster in the layer is denoted as }, the standard deviation of the cluster in the layer is denoted as {, }. The aim object is constructed using the center of the cluster and the standard deviation. The detail formula will be discussed in the next section. Each aim object neuron is connected to a T–S neuron. The T–S neurons are composed of T–S function, formula is as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, {} are the parameters of the T–S neuron, is the input.

After structure learning, we can confirm the number of neurons in the first layer by clustering, and get which is the number of the second layer by area selection. Finally, we can get third layer neurons and fourth layer neurons by clustering to create the model. The model detail will be discussed in the next section.

## Model Structure and I/O Relationship

In this study, the model is a six-layer neural network. The training data set is marked as , is the number of the data, is a *-*by*-*1 input vector, is the number of the input dimensions; is a -by-1 target vector, is the number of the complex-valued targets. Through the model computation, we can get the output .

**Layer0:** This layer is the input layer. Inputs data are the results of multi-target feature selection. The input vector of the time is marked as follows.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Layer1:** This layeris the SCFS layer. After structure learning, several fuzzy sets can be constructed in different dimensions, and the input of each dimension can obtain membership degree via the fuzzy sets. Multiple complex-valued membership degree can be obtained through the SCFS, and the different membership degree can be applied to different model output for multi-target prediction. Through the SCFS formulas (1)- (7), we can get membership degree vector, as follows.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Layer2:** This layer is the premise layer. After structure learning, we can filter out useful areas to construct neuron. Since the input of the neuron is the membership degree, and the output is the result of multiplication of the membership degree of each input dimension, it is called π neuron. Each neuron output is the firing strength. This study uses SCFS, so the output of each neuron will be a vector, and the output will be the same.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

, is the product of , is the membership degree of the dimension membership degree vector in the neuron.

**Layer3:** Aim object layer is used to undertake the output of the previous layer, and it is a vector. The output of this layer is also a vector, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the value of the layer which from the neuron to the aim object, . This study uses SCFS, the inputs are complex-valued and the outputs must also be in a unit disk complex plate. Therefore, aim object needs to be converted to ensure that the outputs are also complex-valued, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |

where, . is the converted center of the aim object, is the converted spread of the aim object, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |

where, uses as the input; uses as the input; is the mean of the target data; is the standard deviation of the target data.

**Layer4:** This layer is the T–S layer, we can get model outputs after computing in this layer, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

, where, is the output; {} are the parameters of the T–S neuron.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |
|  |  |  |

**Layer5:** This layer is the output layer, the result after accumulating model outputs from previous layer is the final output of the model.

|  |  |  |
| --- | --- | --- |
|  |  |  |

## Parameter Learning

According to the divide-and-conquer concept, we use different machine learning algorithms to optimize the parameters of each layer to make it easier to find the best solution. For the parameters optimization of the first layer (fuzzy set) layer, we use the popular particle swarm optimization (PSO)algorithm [19], which is based on the principle for the simulation of birds looking for food. In each iteration, particle adjusts the velocity via the personal best position and the group best position, the formulas are as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

where, is the position of the particle in the iteration; is the velocity of the particle in the iteration; is the best historic position of the particle in the iteration; is the group best position of all particles in the ; are the parameters of PSO; are random values between 0 and 1. In this paper, the position of the particles are the parameters of the fuzzy sets, includes the center, standard deviation, phase functions and for each dimension.

In this paper, the parameters of the T-S neuron are updated using recursive least square estimation (RLSE). The RLSE method uses the data of each row when updating, the recursive update is more effective than the LSE which receives the all data at one time. The LSE method can be regarded as a linear problem, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the target; are the model outputs; {} is the known function of u; {, =1,2,…,m} are estimated unknown parameters; is the model error. The LSE problem also can be expressed as a matrix, as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where,

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |

is the output matrix; is the matrix of the estimated unknown parameters; is the target matrix; is the errors vector. For optimization of , we can compute via RLSE formulas.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

where, is the iterations, {}; is the number of the data; is the row of . When the RLSE algorithm starts, is set to 0, is set to, is a very big positive integer, is an identity matrix.

The steps of the PSO–RLSE hybrid method are as follows.

Step1: Prepare the training data and the testing data.

Step2: The PSO particle position is used as the fuzzy set parameter. Bring training data into the model and calculate the firing strength of each neuron.

Step3: The parameters of the T-S neuron are updated by the RLSE method, and in the RLSE formulas are as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |

. For the multi-target prediction, the firing strength is a vector, the is a vector too. Therefore, the identity matrix is used to replace the constant 1 in the original formula (62), the new formula is as follows.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Step4: After updating all parameters, calculate the outputs of the model.

Step5: Calculate the cost and update the personal best position and the group best position in PSO. The cost function in this paper uses the root mean square error (RMSE), defined as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the error of the data in the model; is Hermitian transpose, means transposing the matrix and conjugating the elements in the matrix.

Step6: Repeat Step2-Step5 for each particle until the iteration end.

# Experimentation

There are 3 experiments in this study, and the targets in each experiment have high correlation, such as the relationship between the opening price and the closing price in experiment 1; the targets in experiment 2 are indicators, where, the Taiwan stock exchange capitalization weighted stock index (TAIEX) is a weighted indicator of stocks listed in Taiwan and represents the volatility of listed stock in Taiwan, the Dow Jones industrial average index (DJI) covers nine major industries such as finance, a weighted stock price index, the national association of securities dealers automated quotation (NASDAQ) is a market-valued weighted index of over 3,000 stocks, most of which are based on the science and technology industry, standard and Poor’s (S&P 500) is the weighted market capitalization of the top 500 companies in the United States, including 11 industries such as IT. All of the above are very famous stock indicators, so accurate forecasting can bring about a good effect for financial strategies. The experiment 3 uses mostly large-scale technology companies, such as IBM, APPLE, DELL and Microsoft. In order to compare the performance with other papers, we will evaluate the model through two indicators. The first two experiments use RMSE (68) as the evaluation index, and the third experiment, based on the data provided by other literatures, we will use mean average percent errors (MAPE) as the assessment index, MAPE formula is as follows.

|  |  |  |
| --- | --- | --- |
|  | , |  |

where, is the number of the data, is the actual data, is the data of the model output.

Structure learning in all experiments, the upper limit is set to 4 and the lower limit is set to 2, so the input dimensions will be between 2 and 4.

## Example 1—Quadruple Time Series of Daily NASDAQ and S&P 500

In this experiment, we use real-world time series data to verify the performance of the model. The data used are the opening and closing prices of the NASDAQ from January 3, 2007 to December 20, 2010 and the daily opening and closing prices of the S&P500 from January 3, 2007 to December 20, 2010. The raw data of this experiment is 1029 in every data set. After making first order difference, 1028 are obtained, and 30 features are extracted from each data set. There are 120 features in total, and 998 data are for each feature. The first 500 data are training data. The rest are the test data. The first to 30th features are NASDAQ opening prices, the 31st to 60th features are NASDAQ closing prices, the 61st to 90th features are S&P 500 opening prices, and the 91st to 120th features are S&P 500 closing prices. These 120 features and the targets form a data matrix. In the data matrix, the S&P500 features are the closest to the targets, and the target sorting is the NASDAQ opening price, the NASDAQ closing price, the S&P500 opening price, and the S&P500 closing price. After multi-target feature selection [22] from data matrix, the features and are selected as the model inputs. There are two complex-valued targets, the real part of the first target is the NASDAQ opening price, the imaginary part is the NASDAQ closing price, and the real part of the second target is the S&P500 opening price, and the imaginary part is S&P500 closing price. In the part of structure learning, each selected feature is clustered using subtractive clustering [11], and through the neurons selection method introduced in section 2, we can filter out 8 neurons from 54 neurons. The parameters of model after structure learning are shown in Table I. The parameters of machine learning of PSO–RLSE hybrid method is are shown in Table II. The model proposed in this paper can have four complex-valued outputs at one time, so the model can predict the four complex-valued targets. In this experiment, we only predict two complex-valued targets.

Model Setting (Experiment 1)

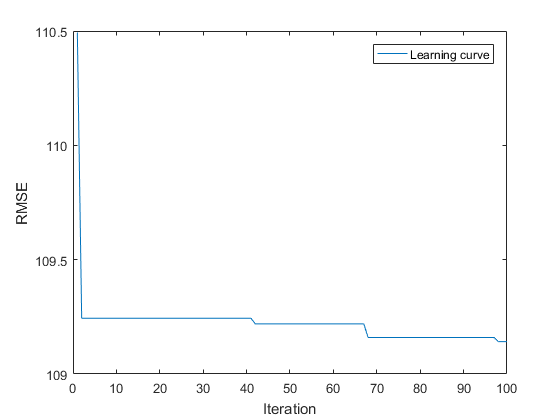
|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {2, 3, 3, 3} |
| Type of fuzzy sets | SCFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 8 |
| Number of parameters in the SCFS layer | 44 |
| Number of aim object neurons | 5 |
| Number of T–S neurons | 5 |
| Number of parameters in the T–S layer | 25 |

\* Each complex-valued target whose real and imaginary parts contain two real-valued targets, respectively.

Machine Learning Setting

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | {0.8, 2.0, 2.0} |
|  | Random in [0,1] |
| Initial particle positions | By SC algorithm in section II-C |
| Initial particle velocities | 0 |
| **RLSE** | |
|  |  |
|  | 25-by-1 zero vector |
|  | **I** |
| **I** | 25-by-25 identity matrix |

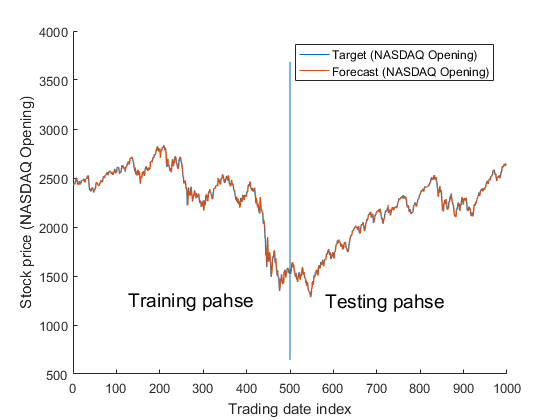
The results of this experiment will compare with the methods proposed in other papers [23]. Except the SVR models, other methods can predict two targets simultaneously. So we use the first complex-valued output to compare with other methods, the results are shown in Table III. The performance of 10 trials are shown as Table IV. Learning curve of the model is shown as Fig. 5, we can find learning is gradually stable in first 10 iterations. The model output and targets are shown as Fig. 6. The prediction errors are shown as Fig. 7. The errors are in a random number range which is from -50 to 50, indicating that the prediction ability of the model is stable.



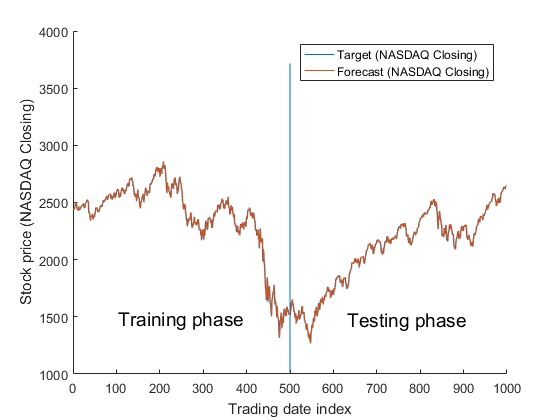
1. Learning curve. The RMSE variation can be seen from this curve. At the first 10 iterations, the learning is gradually stable. (Experiment 1)

Performance Comparison (NASDAQ, Experiment 1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase |  |  | Testing phase |  |
| Opening index | Closing index |  | Opening index | Closing index |
| SVR (two models, each with single output) [23] | 35.18 | 35.24 |  | 37.23 | 40.24 |
| ANFIS (two models, each with single output) [23] | 37.83 | 38.66 |  | 38.80 | 42.36 |
| ANFIS (one model with two outputs) [23] | 62.75 | 71.51 |  | 72.52 | 85.08 |
| RBF (two models, each with single output) [23] | 37.59 | 33.89 |  | 37.52 | 44.08 |
| RBF (one model with two outputs) [23] | 178.57 | 179.87 |  | 261.37 | 258.89 |
| CNFS(5)-ARIMA (one model with two outputs) [23] | 21.56 | 20.81 |  | 32.52 | 33.70 |
| SCNFS(proposed) | 38.58 | 38.59 |  | 27.73 | 27.44 |

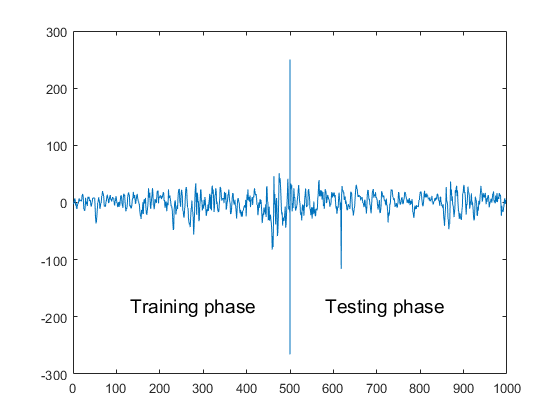


(a)



(b)

1. The actual values of NASDAQ and the model outputs, (a) daily opening price (b) daily closing price. The x-axis is the trading day, and the y-axis is the stock price. It can be seen that both the training and test phases have good results. (Experiment 1)



1. Prediction errors. The errors are in a random number range which is from -50 to 50, indicating that the prediction ability of the model is stable. (Experiment 1)

Ten Trials Performance (Experiment 1)

|  |  |  |
| --- | --- | --- |
|  | Performance (RMSE) | |
| Trials | Opening index | Closing index |
| 1 | 27.67 | 27.75 |
| 2 | 29.62 | 28.04 |
| 3 | 33.85 | 33.99 |
| 4 | 28.09 | 28.77 |
| **5** | **27.73** | **27.44** |
| 6 | 33.32 | 42.27 |
| 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 TAIEX, DJI, NASDQ and S&P 500

In this experiment, we also use real-world time series data to verify the performance of the model. The difference from experiment 1 is that the four targets are not the relationship between the closing price and the opening price, which means that the similarity of the target curves is not as high as the experiment 1. The data used are the closing prices of TAIEX, DJI, NASDAQ and S&P500 from 2001 to 2004. The closing prices per year are as follows. In 2001, there is 245, in 2002 there is 248, in 2003 there is 249, in 2004 there is 250. The model predicts the closing price for every year, the data for the first 10 months is used as training data and the rest is used as testing data. After making first order difference, 30 features are extracted from each data set. There are 120 features in total. The first to 30th features are TAIEX closing prices, the 31st to 60th features are DJI closing prices, the 61st to 90th features are NASDAQ closing prices, and the 91st to 120th features are S&P 500 closing prices. These 120 features and the targets form a data matrix. In the data matrix, the S&P500 features are the closest to the targets, and the target sorting is the TAIEX closing price, the DJI closing price, the S&P500 closing price, and the NASDAQ closing price. After multi-target feature selection [22] from data matrix, the selected features are used as the model inputs. There are two complex-valued targets, the real part of the first target is the TAIEX closing price, the imaginary part is the DJI closing price, and the real part of the second target is the S&P500 closing price, and the imaginary part is NASDAQ closing price. In the part of structure learning, each selected feature is clustered using subtractive clustering [11], and through the neurons selection method introduced in section 2. The parameters of model after structure learning are shown in Table V- Table VIII. The parameters of machine learning of PSO–RLSE hybrid method is are shown in Table IX. The model proposed in this paper can have four complex-valued outputs at one time, so the model can predict the four complex-valued targets. In this experiment, we only predict two complex-valued targets.

Model Setting (2001, Experiment 2)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 4, 3, 4} |
| Type of fuzzy sets | SCFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 15 |
| Number of parameters in the SCFS layer | 56 |
| Number of aim object neurons | 3 |
| Number of T–S neurons | 3 |
| Number of parameters in the T–S layer | 15 |

\* Each complex-valued target whose real and imaginary parts contain two real-valued targets, respectively.

Model Setting (2002, Experiment 2)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 3, 3, 3} |
| Type of fuzzy sets | SCFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 13 |
| Number of parameters in the SCFS layer | 48 |
| Number of aim object neurons | 5 |
| Number of T–S neurons | 5 |
| Number of parameters in the T–S layer | 25 |

\* Each complex-valued target whose real and imaginary parts contain two real-valued targets, respectively.

Model Setting (2003, Experiment 2)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 3, 3, 3} |
| Type of fuzzy sets | SCFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 11 |
| Number of parameters in the SCFS layer | 48 |
| Number of aim object neurons | 4 |
| Number of T–S neurons | 4 |
| Number of parameters in the T–S layer | 20 |

\* Each complex-valued target whose real and imaginary parts contain two real-valued targets, respectively.

Model Setting (2004, Experiment 2)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {5, 4, 4, 4} |
| Type of fuzzy sets | SCFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 15 |
| Number of parameters in the SCFS layer | 68 |
| Number of aim object neurons | 8 |
| Number of T–S neurons | 8 |
| Number of parameters in the T–S layer | 40 |

\* Each complex-valued target whose real and imaginary parts contain two real-valued targets, respectively.

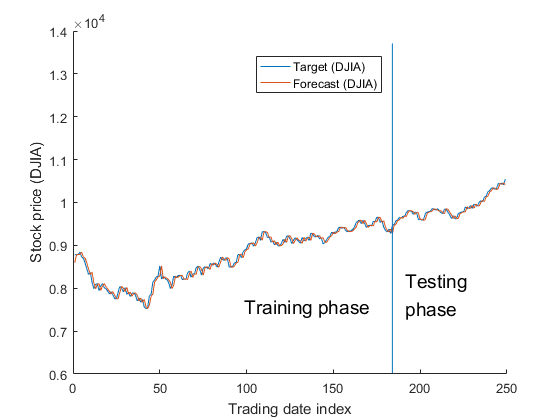
Machine Learning

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | {0.8, 2.0, 2.0} |
|  | Random in [0,1] |
| Initial particle positions | By SC algorithm in section II-C |
| Initial particle velocities | 0 |
| **RLSE** | |
|  |  |
|  | By number of parameters in the T–S layer |
|  | -by-1 zero vector |
|  | **I** |
| **I** | -by- identity matrix |

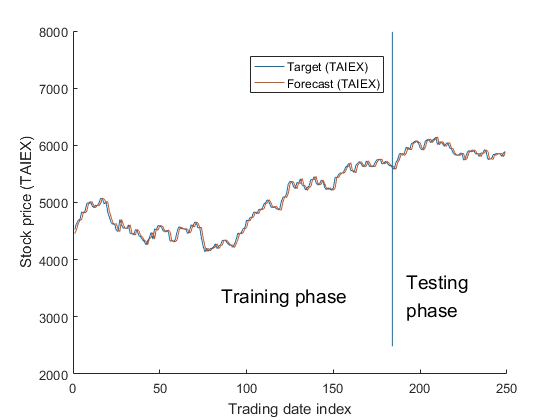
The results of this experiment will compare with the methods proposed in other papers [23]. Except the SVR models, other methods can predict two targets simultaneously. So we use the first complex-valued output to compare with other methods, the results are shown in Table XI and Table XII. The performance of 10 trials are shown as Table X. The model output and targets are shown as Fig. 8. The prediction errors are shown as Fig. 9. The errors are in a random number range which is from -10 to 10, indicating that the prediction ability of the model is stable.

Ten Trails Performance (Experiment 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Performance (RMSE) | | | |
| Trials | 2001 | 2002 | 2003 | 2004 |
| 1 | 259.74 | 279.93 | **196.91** | 282.82 |
| 2 | **259.27** | 286.02 | 198.10 | 283.94 |
| 3 | 260.99 | 280.34 | 199.06 | 286.83 |
| 4 | 260.44 | 280.56 | 199.54 | 284.57 |
| 5 | 263.29 | 281.85 | 196.94 | 278.60 |
| 6 | 263.57 | 282.00 | 197.85 | 288.52 |
| 7 | 259.86 | **279.46** | 200.05 | **275.31** |
| 8 | 262.02 | 282.35 | 198.33 | 285.69 |
| 9 | 261.23 | 282.11 | 197.53 | 288.15 |
| 10 | 260.59 | 280.93 | 197.62 | 286.68 |

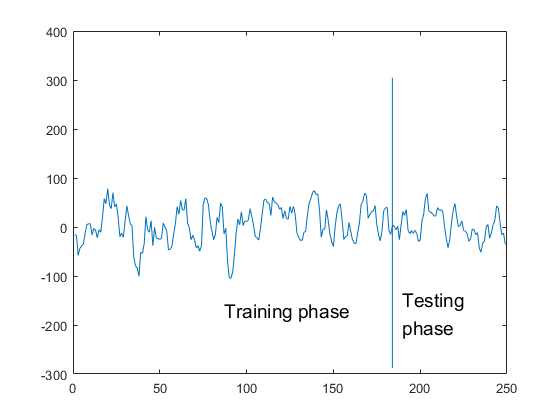


(a)



(b)

1. The closing prices and the model outputs, (a) DJIA (2003) (b) TAIEX (2003). The x-axis is the trading day, and the y-axis is the stock price. It can be seen that both the training and test phases have good results. (Experiment 2)



1. Prediction errors. The errors are in a random number range which is from -100 to 100, indicating that the prediction ability of the model is stable. (Experiment 2)

Performance Comparison (DJIA, Experiment 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **RMSE** | | | |
| **Method** | **2001** | **2002** | **2003** | **2004** |
| SVR (two models, each with single output) [23] | 101.44 | 117.95 | 82.76 | 71.49 |
| ANFIS (two models, each with single output) [23] | 105.56 | 111.69 | 72.09 | 68.00 |
| ANFIS (one model with two outputs) [23] | 128.20 | 142.05 | 90.37 | 83.69 |
| RBF (two models, each with single output) [23] | 106.33 | 131.24 | 97.58 | 81.79 |
| RBF (one model with two outputs) [23] | 181.79 | 136.28 | 154.14 | 148.11 |
| CNFS(5)-ARIMA (one model with two outputs) [23] | 103.06 | 103.42 | 70.70 | 66.55 |
| SCNFS(proposed) training phase | 91.95 | 98.69 | 69.66 | 97.99 |
| SCNFS(proposed) testing phase | 94.31 | 85.52 | 56.26 | 61.64 |

Performance Comparison (TAIEX, Experiment 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **RMSE** | | | |
| **Method** | **2001** | **2002** | **2003** | **2004** |
| SVR (two models, each with single output) [23] | 162.46 | 67.72 | 59.47 | 58.81 |
| ANFIS (two models, each with single output) [23] | 147.36 | 70.17 | 72.61 | 65.33 |
| ANFIS (one model with two outputs) [23] | 151.62 | 78.27 | 81.69 | 70.54 |
| RBF (two models, each with single output) [23] | 134.32 | 65.15 | 60.41 | 102.86 |
| RBF (one model with two outputs) [23] | 137.58 | 78.54 | 115.92 | 126.48 |
| CNFS(5)-ARIMA (one model with two outputs) [23] | 115.82 | 64.34 | 57.69 | 55.56 |
| SCNFS(proposed) training phase | 92.03 | 100.26 | 69.96 | 99.11 |
| SCNFS(proposed) testing phase | 89.59 | 86.81 | 55.34 | 60.30 |

## Example 3—Quadruple Time Series of Daily APPLE, IBM, DELL, and Microsoft

In this experiment, we use real-world time series data to verify the performance of the model. The data used are the closing prices of APPLE, IBM, DELL, and Microsoft from February 10, 2003 to January 21, 2005. The raw data of this experiment is 492 in every data set. After making first order difference, 491 are obtained, and 30 features are extracted from each data set. There are 120 features in total, and 460 data are for each feature. The first 433 data are training data. The rest are the test data. The first to 30th features are APPLE closing prices, the 31st to 60th features are IBM closing prices, the 61st to 90th features are DELL closing prices, and the 91st to 120th features are Microsoft closing prices. These 120 features and the targets form a data matrix. In the data matrix, the Microsoft features are the closest to the targets, and the target sorting is the APPLE closing price, the IBM closing price, the DELL closing price, and the Microsoft closing price. After multi-target feature selection [22] from data matrix, the features and are selected as the model inputs. There are two complex-valued targets, the real part of the first target is the APPLE closing price, the imaginary part is the IBM closing price, and the real part of the second target is the DELL closing price, and the imaginary part is Microsoft closing price. In the part of structure learning, each selected feature is clustered using subtractive clustering [11], and through the neurons selection method introduced in section 2, we can filter out 13 neurons from 81 neurons. The parameters of model after structure learning are shown in Table XIII. The parameters of machine learning of PSO–RLSE hybrid method is are shown in Table XIV. The model proposed in this paper can have four complex-valued outputs at one time, so the model can predict the four complex-valued targets. In this experiment, we only predict two complex-valued targets.

Model Setting (Experiment 3)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 3, 3, 3} |
| Type of fuzzy sets | SCFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 13 |
| Number of parameters in the SCFS layer | 48 |
| Number of aim object neurons | 3 |
| Number of T–S neurons | 3 |
| Number of parameters in the T–S layer | 15 |

\* Each complex-valued target whose real and imaginary parts contain two real-valued targets, respectively.

Machine Learning Setting

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | {0.8, 2.0, 2.0} |
|  | Random in [0,1] |
| Initial particle positions | By SC algorithm in section II-C |
| Initial particle velocities | 0 |
| **RLSE** | |
|  |  |
|  | 15-by-1 zero vector |
|  | **I** |
| **I** | 15-by-15 identity matrix |

The results of this experiment will compare with the methods proposed in other papers, such as HiMMI [13], ANN-GA-HMM-Interpolation [13], ANN-GA-HMM-WA [13], ARIMA[41], Bayesian ANN[41]. We use the first complex-valued output to compare with other methods, the results are shown in Table XVI. The performance of 10 trials are shown as Table X. Learning curve of the model is shown as Fig. 10, we can find learning is gradually stable in the 45th iterations. The model output and targets are shown as Fig. 11, APPLE stock price is about 5 US dollars, so the forecast curve looks fluctuating. The prediction errors are shown as Fig. 12. The errors are in a random number range which is from -2 to 2, indicating that the prediction ability of the model is stable.



1. Learning curve. The RMSE variation can be seen from this curve. At the 45th iteration, the learning is gradually stable. (Experiment 3)

Ten Trails Performance (Experiment 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Performance (MAPE)** | | | |
| **Trials** | **APPLE** | **IBM** | **DELL** | **Microsoft** |
| 1 | 1.8909 | 0.8156 | 0.6173 | 0.7007 |
| 2 | 2.3692 | 0.8286 | 0.5957 | 0.7111 |
| 3 | **1.8453** | **0.8051** | **0.6187** | **0.8591** |
| 4 | 2.0273 | 0.8172 | 0.6184 | 0.7048 |
| 5 | 2.0476 | 0.8183 | 0.6100 | 0.7099 |
| 6 | 2.2186 | 0.8005 | 0.6270 | 0.7399 |
| 7 | 2.0239 | 0.8313 | 0.6219 | 0.7128 |
| 8 | 2.2186 | 0.8005 | 0.6270 | 0.7399 |
| 9 | 2.0239 | 0.8313 | 0.6219 | 0.7128 |
| 10 | 1.8636 | 0.8159 | 0.6159 | 0.7045 |



(a)



(b)



(c)

1. The actual closing prices and the model outputs, (a) IBM (2003) (b) APPLE (2003) (c) DELL. The x-axis is the trading day, and the y-axis is the stock price. It can be seen that both the training and test phases have good results. (Experiment 3)



1. Prediction errors. The errors are in a random number range which is from -2 to 2, indicating that the prediction ability of the model is stable. (Experiment 3)

Performance Comparison (Experiment 3)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **MAPE** | | | | |
| **Method** |  | **APPLE** |  | **IBM** |  | **DELL** |
| HiMMI[13] |  | 2.8373 |  | 1.2186 |  | 1.0117 |
| ANN-GA-HMM-Interpolation[13] |  | 2.1649 |  | 1.0555 |  | 0.8446 |
| ANN-GA-HMM-WA[13] |  | 1.9247 |  | 0.8487 |  | 0.6992 |
| Bayesian ANN[41] |  | 1.9688 |  | 0.7441 |  | - |
| ARIMA[41] |  | 1.8009 |  | 0.9723 |  | 0.6604 |
| SCNFS(proposed) training phase |  | 2.4175 |  | 1.1758 |  | 0.8955 |
| SCNFS(proposed) testing phase |  | 1.8453 |  | 0.8051 |  | 0.6187 |

# Discussion

This study proposes PSO-RLSE hybrid algorithm to optimize the model parameters. Before the training data enters the model, we make the feature selection to select more favorable features for multi-target prediction. The part of multi-target feature selection, we use the theory of the Shannon information entropy [39] to calculate the information that the feature data can provide to each target data, and we consider the redundancy information between the feature and the selected features to avoid reducing overall contribution. Selecting the maximum selection gain means that the feature can be provided to the target more information. In addition, the user can select the number of features to enter the model through the second selection to increase the overall efficiency and reduce the burden of model for multiple targets. In the process of structure learning, through the concept of data density, we select and construct important neurons. The user can select the number of neurons from the selected neurons by setting the upper and lower bounds. This method reduces the number of neurons, reduces the time required for the operation, and can form different models for different input data. Size, improve model adaptability. The user can select the number of neurons from the selected neurons by setting the upper and lower limits. This method reduces the number of neurons, shorten the time required for the computation, and can form different model size for different input data to improve model adaptability. The SCNFS is combined with a SCFS and a T-S fuzzy system. The T-S fuzzy system can process fuzzy information and describe the strength of the input data with a non-linear type, and represent the model output with a linear rule, so that the system can be understood by humans. The SCFS provides at least four complex-valued membership degree. Compared with the traditional fuzzy sets for one single target, the model can be used to predict eight targets at the same time. In addition, it is also possible to deconstruct the complex-valued values to predict more targets at the same time. The SCFS has a higher scalability and information richness. In the part of parameter learning, the PSO and RLSE are used to optimize parameters of two parts in the model, respectively. The PSO has three characteristics. First, it can automatically adjust the pace with the swarm intelligence; secondly, when the velocity updating, random parameters can participate in the process, which can enhance the activity of the particles; thirdly, it will follow a certain correct direction. However, it has the disadvantage of lower search dimension, so it uses the divide and conquer method and cooperates with RLSE to optimize parameters. The RLSE uses the previous calculation results to find a linear function that minimizes the squared error between the data and the function. This method is continuously recursive so that the model output approximates the target and optimizes the parameters.

After three experiments, the research method of this study is shown to have predictive ability for time series data. Through the table of the model parameters, we can find that the neurons decrease from nearly one hundred to about ten, which apparently controls the model size. The SCFNS has the ability to predict multiple targets, and we can find that the performance of each target is not bad than other papers, or even better. The PSO-RLSE hybrid method is proved to have a certain level. The part of parameter learning may be limited by the characteristics of the PSO, which converges quickly and easily falls into the local minimum. From Fig. 2, it is not difficult to find that the PSO algorithm has approached the best RMSE in the first few iterations. Therefore, in the case of a large data volume, the required the number of the search dimensions increase, and the PSO performance will be poor which may limit the overall performance of the model.

A series of methods proposed in this study can effectively predict multiple targets. It is proved that the compositions of sphere complex fuzzy sets increase the richness of information. The size of the model can be controlled by the data and the user's settings, and the speed of the model computation can be improved. In addition to structure learning, machine learning is also applied to parameter learning. The PSO-RLSE method adopts the concept of divide and conquer, which makes the model still have good results when it optimizes multiple parameters.

# Concluding Remarks

This study proposes a PSO-RLSE method, combined with PSO and RLSE to optimize the parameter set of a SCNFS. The model uses SCFSs, T-S system, and neural network concept. Data preprocessing uses multi-target feature selection to reduce the model burden. Before the input data entering the model, filter out the candidate features which generated by the raw data to avoid the redundancy input data. A SCFS allows the model to have multiple complex-valued membership degrees which make the model have multiple complex-valued output. It is different from general fuzzy sets, the richness of information is increased, so that the model can have the ability to predict multiple targets at the same time. Three experiments can demonstrate the contribution of this approach. The parameters of the model determine the outcome of the forecast. The PSO-RLSE hybrid algorithm optimizes different regional parameters using different machine learning algorithms through the concept of divide-and-conquer, minimizes problems to reduces the search dimension of the algorithm and enhances the overall performance of the model.

The PSO algorithm still has the disadvantage of small search dimensions and premature, that is, it falls into the local solution in a short time. In the future, SCFNS can combine with different machine learning algorithms such as random optimization (RO) [45], cooperative particle swarm optimization (CPSO) [4], and ant colony optimization (ACO) [8] et al. May be able to overcome the problems encountered by the current PSO and make more accurate predictions. In recent years, the development of deep learning has been rapid. The six-layer neural network is a relatively small structure compared to other papers. In the future, more hidden layers can be added into the neural network, and more sophisticated calculations can be used to find better prediction.

##### Acknowledgment

This work was supported by the research projects MOST 105–2221–E–008–091 and MOST 104–2221–E–008–116, Ministry of Science & Technology, Taiwan.

##### References

1. M. H. Alobaidi, F. Chebana and M. A. Megiud, “Robust ensemble learning framework for day-ahead forecasting of household based energy consumption,” Applied Energy, vol. 212, pp. 997-1012, Feb. 2018.
2. P. R. L. Alves, L. G. S. Duarte and L.A.C.P. da Mota, “A new characterization of chaos from aa time series,” Chaos, Solitons and Fractals, vol. 104, pp. 323-326, Aug. 2017.
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 forecasting emergency department demand in Western Australia,” Journal of Biomedical Informatics, vol. 57, pp. 62–73, Oct. 2015.
4. F. van den Bergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 225–239, Jun. 2004.
5. J. J. Buckley and Y. Qu, “Fuzzy complex analysis–II: Integration,” Fuzzy Sets System, vol. 49, no. 2, pp. 171–179, 1992.
6. J. J. Buckley and Y. Qu, “Fuzzy complex analysis–I: Differentiation,” Fuzzy Sets System, vol. 41, no. 3, pp. 269–284, 1991.
7. J. J. Buckley, “Fuzzy complex numbers,” Fuzzy Sets System, vol. 33, no. 3, pp. 333–345, 1989.
8. A. Colorni, M. Dorigo and V. Maniezzo, “Distributed optimization by ant colonies,” Proceedings of the 1st European Conference on Artificial Life, pp. 134-142, 1991.
9. H. d. N. Camelo, P. S. Lucio, J. B. V. L. Junior, P. C. M. d. Carvalho and D. v. G. d. Santos, “Innovative hybrid models for forecasting time series applied in wind generation based on the combination of time series models with artificial neural networks,” Energy, in press.
10. C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, vol. 20, pp. 273-297, 1995.
11. S. Chiu, “Fuzzy model identification based on cluster estimation,” Journal of Intelligent and Fuzzy Systems, pp. 267–278 Jan. 1994.
12. R. Clausius, “Ueber eine veränderte Form des zweiten Hauptsatzes der mechanischen Wärmetheorie,” Annalen der Physik und Chemie, vol. 93, no. 12, pp. 481–506, 1854.
13. 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, no. 1, pp. 171-180, Jul. 2007.
14. 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, no. 13–15, pp. 2410–2425, Aug. 2007.
15. D. Hebb: The Organization of Behavior, 1949.
16. 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, no. 1, pp. 123–139, Feb. 2009.
17. J. S. R. Jang, C. T. Sun and E. Mizutani, “Neuro–fuzzy and soft computing: A computational approach to learning and machine intelligence,” IEEE Transactions on Automatic Control, vol. 42, no. 10, Oct. 1997.
18. J. S. R. Jang, “ANFIS: Adaptive–network–based fuzzy inference system,” IEEE Transactions on System, Man, and Cybernetics, vol. 23, no. 3, pp. 665–685, May/Jun, 1993.
19. J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, Nov.–Dec. 1995.
20. R. S. J. Koijen, H. Lustig and S. V. Nieuwerburgh, “The cross-section and time series of stock and bond returns,” Journal of Monetary Economics, vol. 88, pp. 50-69, Jun. 2017.
21. M. Z.–Kermani and M. Teshnehlab, “Using adaptive neuro–fuzzy inference system for hydrological time series prediction.” Applied Soft Computing, vol. 8, no. 2, pp. 928–936, Mar. 2008.
22. C. Li, “Multi–target feature selection,” research training material for graduate students in the dept. of Information Management, National Central University, Taiwan, unpublished, 2017.
23. 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, Jun. 2013.
24. C. Li and T.–W. Chiang, “Complex fuzzy computing to time series prediction—A multi–swarm PSO learning approach,” Lecture Notes in Computer Science, vol. 6592, pp. 242–251, 2011.
25. C. Li and T.–W. Chiang, “Complex neuro–fuzzy self–learning approach to function approximation,” Lecture Notes in Computer Science, vol. 5991, pp. 289–299, 2010.
26. C. Li, T.–W. Chiang. J.–W. Hu, and T. Wu, “Complex neuro–fuzzy intelligent approach to function approximation,” Advanced Computational Intelligence. 2010 3rd International Workshop on Advanced Computational Intelligence, pp. 151–156, 2010.
27. 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, no. 2, pp. 295–308, Mar. 2012.
28. W. Liang-Ying, “A hybrid time series model based on AR-EMD and volatility for medical data forecasting: A case study in the emergency department,” International Journal of Management, Economics and Social Sciences, vol. 6, no. Special issue, pp. 166-184, Oct. 2017.
29. K. O. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz and R. Jenssen, “Time series cluster kernel foe learning similarities between multivariate time series with missing data,” Pattern Recognition, vol. 76, pp. 569-581, Apr. 2018.
30. W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” Bulletin of Mathematical Biophysics, vol. 5 1943.
31. D. Osthus, K. S. Hickmann, P. C. Caragea, D. Higdon and S. Y. Valle, “Forecasting seasonal influenza with a state-space SIR model,” The Annals of Applied Statistics, vol. 11, no. 1, pp. 202-224, Sep. 2017.
32. Z. Pan and L. Liu, “Forecasting stock return volatility: A comparison between the roles of short-term and long-term leverage effects,” Physica A: Statistical Mechanics and its Applications, vol. 492, pp. 168-180, Feb. 2018.
33. N. Rochester, J. H. Holland, L. H. Haibt and W. L. Duda, “Test on a cell assembly theory of the action of the brain, using a large digital computer,” IRE Transactions on Information Theory, 1956.
34. D. Ramot, R. Milo, M. Friedman, and A, Kandel, “Complex fuzzy sets,” IEEE Transactions on Fuzzy Systems, vol. 10, no. 2, pp. 171–186, Apr. 2002.
35. 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, no. 9, pp. 1260–1275, May 2006.
36. S. Z. Selim and M. S. Kamel, “On the mathematical and numerical properties of the fuzzy c-means algorithm1,” Fuzzy Sets and Systems, vol. 49, no. 2, pp. 181-191, Jul. 1992.
37. I. Sugiarto and S. Natarajan, “Parameter estimation using least square method for MIMO Takagi–Sugeno neuro–fuzzy in time series forecasting,” Jurnal Teknik Elektro, vol. 7, no. 2, pp. 82–87, Sep. 2007.
38. R. Stuart and N. Peter: Artificial Intelligence: A Modern Approach 2nd. Prentice Hall, 1995.
39. E. C. Shannon, “A mathematical theory of communication,” The Bell System Technical Journal, vol. 27, pp. 379–423, Jul. 1948.
40. T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC–15, no. 1, pp. 116–132, Jan.–Feb. 1985.
41. L. Ticknor, “A Bayesian regularized artificial neural network for stock market forecasting,” Expert Systems with Applications, vol. 40, no. 14, Oct. 2013.
42. P. Werbos, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,” PhD thesis, Harvard University, 1974.
43. Y. Zhang and P. Shang, “Refined composite multiscale weighted-permutation entropy of financial time series,” Physica A: Statistical Mechanics and its Application, vol. 496, pp. 189-199, Apr. 2018.
44. L. A. Zadeh, “Fuzzy sets”, Information and Control, vol. 8, no. 3, pp. 338–353, Jun. 1965.
45. Z. B. Zabinsky, “Random search algorithms,” Technical report, University of Washington, Seattle, Apr. 2009.