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

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

Chi Feng Lina and Chunshien Lib

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

aEmail: j8888888871@yahoo.com.tw

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

*Abstract*— This paper mainly discusses the prediction performance of different algorithms in time series, and conducts performance through simulated investment strategies. Time series prediction is a wide and important issue. This paper proposes a sphere complex neuro-fuzzy system (SCNFS) to predict the time series. A complex neuro-fuzzy system (CNFS) has a complex-valued output, the real part and the imaginary part can predict different target. In this paper, we improve the complex fuzzy sets (CFSs) in CNFS and called it sphere complex fuzzy sets (SCFSs), the membership degrees are also complex-valued, but SCNFS can have multiple targets, which makes the model can predict more than two targets simultaneously. In the part of model design, Gaussian-type SCFSs are used in the premise part, and Takagi-Sugeno linear functions are used in the consequence part, both use the IF-THEN rule type link. In addition, in order to optimize the model output, this paper uses the concept of divide-and-conquer, the premise part respectively uses particle swarm optimization (PSO), artificial bee colony optimization (ABCO) to optimize the parameters, the consequence part uses recursive least squares estimator (RLSE) to optimize the parameters. Finally, three experiments are used to testify performance of different algorithms and combine with the investment strategy proposed by this paper to simulate the investment effect.

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

# Introduction

Artificial intelligence has many application fields in reality. From the 18th and 19th centuries, there are many related researches, but due to hardware equipment problems, the results have not been applied to ordinary life. Nowadays, with the development of hardware devices, artificial intelligence has already brought a lot of help to human beings, and research is even more numerous, like financial and economic forecasts. In 2017, Jardin et al. [7] proposed a quantitative approach to defining a company's health status to predict whether a company is bankrupt; Ravi et al. [26] used chaos theory, multi-layer perceptron and multi-target optimization algorithms to predict exchange rates; Heaton et al. [12] proposed a deep learning model for the financial portfolio. In the picture classification, Feng et al. proposed Discriminative Locality Alignment Network (DLANet) for image recognition [9], which improves the ability to recognize scenes; Howard et al. [14] proposed MobileNets to give neural network weights through a streamlined structure, which is specifically provided for mobile devices for image recognition; Zbontar et al. [38] proposed a method for applying the convolutional neural network to the image patch to make pairing of the stereograms easier. In terms of energy forecasting, Fumo et al. [10] used regression analysis to predict the residential energy consumption; Jovanović et al. [16] used back-propagation neural networks to predict heating energy, with good precision.

From the above, it can be known that the application of artificial intelligence is very extensive. Among them, financial finance is the most widely studied because it is a multi-factor problem involving global trends, local culture, political influence, etc. Therefore, exchange rate and stock investment are the most difficult to predict. The data of these problems are sorted according to time, and can be regarded as the prediction problem of time series. In the past, many scholars have proposed different methods to predict the prediction problem of time series, like ARIMA[21, 22], fuzzy theory, neural network, neuro-fuzzy systems (NFSs), etc. Among them, the most commonly proposed is the NFSs [13, 28, 30, 40], which has the best performance for time series prediction.

NFSs are based on the extension of the Neural Network, a network in which multiple neuron layers are connected, transmitted through layers between layers to get the outputs, originated in 1943, McCulloch et al. [24] used threshold logic to simulate neurons in the human brain, meaning that when a neuron receives information, it determines whether a mechanism for excitatory responses is to be produced. In 1956, Rochester [27] further created the perceptron, but still could not solve the nonlinear problem. It was not until 1975 that Werbos [36] proposed a back-propagation algorithm that allowed multiple layers of neural networks to be trained, which not only solved the nonlinear problem, but also created the concept of training in the neural network in the future. Today's deep learning is a kind of neural network, and there have been many studies in this field [9, 11, 38].

NFSs are composed of fuzzy systems and neural networks. The fuzzy systems have so-called IF-THEN rules. These rules are like human experience, which are easy to be understood for human. Nowadays, the widely used NFSs are hybrid NFSs [34], which means that the elements in the fuzzy system are used in the neural network to make the overall architecture more flexible. As mentioned above, the characteristics of the neuro-fuzzy system make him have a good effect on the prediction of time series. Therefore, most of the research on time prediction now uses a neural network as the model architecture.

This paper uses a NFS as the framework, which contains the IF-THEN rules to build neurons of multiple layers. In term of model implementation, IF-part uses the Gaussian sphere complex fuzzy sets, which make the model can have multiple outputs to predict multiple targets, and the THEN-part use Takagi-Sugeno linear function [31]. The two neurons are combined by the IF-THEN rule concept, which is a one-to-one relationship. Through this model and machine learning, we expect that the prediction of time series can be excellent.

Regarding fuzzy sets, in 1965, Zadeh first proposed the concept of fuzzy sets [39], so that data can obtain a membership degree of a set between 0 and 1 through a function. Then in 2002, another study proposed the concept of complex fuzzy sets (CFSs) [25]. The complex-valued membership degree can be obtained by a function, which allows the membership degree to be presented in a complex unit disk of radius 1. This concept makes the original membership degree more abundant. We can get the complex-valued outputs by a complex neuro-fuzzy system (CNFS) [19, 20], the real and imaginary part can predict for two different targets, respectively. There are already many research outputs in the prediction of the two targets [19, 20, 22]. In order to predict more targets at the same time, this paper improves the original CNFS and changes the CFSs into sphere complex fuzzy sets (SCFSs), which membership degree is also complex-valued but can present in a 3-D space. SCFSs can have more complex-valued outputs, means can predict more targets at the same time.

In this study, in order to apply the data effectively, in the data preprocessing, we calculate the information contribution of the individual features to the target through Shannon Entropy [29]. In addition, we through the concept of multi-target feature selection [23], the effective information amount of each feature to the target is calculated as the basis for selecting training data. Extracting the most effective data from the data can not only effectively improve the performance of the prediction but also reduce the computational burden of the model. Finally, the parameter learning part, we use the well-known particle swarm optimization (PSO)[8], artificial bee colony optimization (ABCO)[17] two algorithm to combine with recursive least squares estimator (RLSE)[15] for parameter optimization, which are called PSO-RLSE [18] and ABCO-RLSE. We use different algorithm to train the premise and consequence parameters, expect reduce the search dimension through the divide-and-conquer concept and make the model find the best solution easier.

# Methodology

This section will illustrate a series of research methods in this study to model the design. In this study, the data-driven method is used to determine the size of the model structure. The model is implemented using a sphere complex fuzzy set, and the parameters of if-part are optimized by different algorithms, and the recursive least squares algorithm is used to optimize the parameters in the then-part. Before the data entering the model, we select the useful feature by multi-target feature selection [23]. Finally, combine with the investment strategy to compare with different algorithm.

## 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. Afterward, Buckley proposed the complex number [1-3]. In 2002, Ramot et al. [25] further proposed complex fuzzy sets (CFSs), which can have more richer information. 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 [29] is used, and the multi-target feature selection method [23] is used to select the suitable features. Finally, training data are obtained from the selected features.

In 1948, Shannon proposed the concept of information entropy [29]. 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 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 [23].

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

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 [23] 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

Through the data-driven concept, the model structure form automatically. This study use machine learning to make the training data more logically apply in the model construction. In addition, the result of the structure learning will become the part of the parameters. In this study, the training data of different input dimensions are clustered by the subtractive clustering algorithm [5]. 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.

This study model is constructed by traditional if-then rules concept, so the number of neurons in the premise layer is the same as the consequence layer. Through the block selection, neurons in the second layer can be obtained. Finally, construct neurons in the fourth layer too. The neurons in consequence layer is T-S neuron, constructed by T-S function as follows.

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

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

Through the structure learning, this paper construct the model objectively, the computation process and detail of the model 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 .

The model is constructed by T-S fuzzy rule, each rule is combination of the If part and the Then part as follows.

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

; are the input variables of the fuzzy system; are the CFSs of the rule; are the input linguistic variables; are the parameters of CFS, which are the parameters of if-part; are the parameters of then-part. This SCNFS can turn into a 6-layers neural network.

**Layer 0:** This is the input layer. Inputs data are the results of multi-target feature selection. The time of the input vector is marked as follows.

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

**Layer 1:** This is SCFS layer, through the structure learning, several fuzzy sets can be constructed in different dimensions, and the input in each dimension can obtain membership degree via SCFSs. Through the SCFSs, the multiple complex-valued membership degrees can be obtained. Different membership degree can be applied to different model outputs for multi-target prediction. Through the formulas of sphere complex fuzzy sets (1)-(7), a membership degree vector can be obtained as follows.

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

**Layer 2:** This is if-part (premise) layer. Through the structure learning, we can select neurons, since the output of neuron is the product of each input, it is called neuron. Output of neuron is firing strength. And this study use the CFSs, input of each neuron is vector, the output is the same.

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

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

**Layer 3:** This is normalization layer, the function is to normalize each element in input vector, and the output is a vector as follows.

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

where, is the normalized value of the element in the neuron, . This study uses CFSs, the inputs are complex-valued and the outputs are the same.

**Layer 4:** This is T-S layer, through the calculation in this layer, model outputs can be obtained as follows.

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

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

**Layer 5:** This is output layer, the sum of neuron outputs is model output.

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

## Parameter Learning

According to the concept of divide and conquer, we will use different machine learning algorithms to optimize the parameters of each layer to make it easier to find the best solution. For the parameter optimization of the first layer (CFS layer), we use two different algorithms, including particle swarm optimization (PSO) and artificial bee colony optimization (ABCO). There are no parameters need to be optimized in the second layer and the third layer. In the fourth layer, the recursive least squares estimation (RLSE) is used to optimize the parameters of consequence part. We expect that the combination of different algorithms will reduce the parameter dimensions of the search and bring better performance to the model. The details of three algorithms and the formulas are described below.

1. Particle Swarm Optimization

PSO algorithm is an evolutionary computational technique developed by J. Kennedy et al. [8] in 1995, derived from the simulation of society model. The principle is similar to the birds looking for food. In addition to the information provided by particle itself, inertia and its own best position, it also applies the group best position to adjust the speed. The characteristics of this algorithm are fast convergence, and the formula is as follows.

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

where, is the particle position in the iteration; is the particle velocity in the iteration; is the particle personal best position in the iteration; is the group best position in the iteration, are the parameters of PSO, are random number between 0 to 1. In this study, particle position means the parameters of the if-part, including the center, standard deviation and phase frequency parameter in each dimension Artificial Bee Colony Optimization.

1. Artificial Bee Colony Optimization

The artificial bee colony algorithm was proposed by the Karaboga [17], and the principle is similar to the concept of bees searching for food sources. Its characteristics include swarm intelligence and randomness. For example, bees communicate about food source by a swing dance, and the swing dance includes the bias and randomness. In ABCO, there are three types of bees, including employed bee, onlooker bee, and scout bee. Where, employed bees seek the food source and dance to deliver the information, each time the communication has randomness, it represents the search for the overall dimension; The onlooker bee is responsible for searching near food sources. Firstly, use the roulette method to select one of the food sources and search around the food source, means parameters optimization for small dimension; The role of the scout bee is that when the food source has passed a certain number of developments, if the food source is not improved, the scout bee will replace the food source, which means randomly explore a new food source, avoiding the algorithm falling into local minimum. The process of ABCO is not the same as that bees seeking food in real world, the algorithm steps are as follows.

1. Randomly select an employed bee and form a new position which is food source, as follows.

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

where, it the dimension of the food source; is the dimension of the employed bee; is the dimension of another random employed bee.

1. Onlooker bee uses roulette method to select one food source, where the better food source income, the easier it is to be selected. Roulette probability is as follows.

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

where, is the probability of being selected for the food source; is the income, this study uses the reciprocal of the cost as the income; is the number of food sources.

1. Send all onlooker bees to search around the selected food source, formula is as follows.

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

where, is the dimension of the onlooker bee; is the dimension of the selected food source; is the dimension of the other random food source, if the onlooker bee position is better than food source position, replace the food source position.

1. If an employed bee has reached the limit iteration and has no update, send the scout bee to replace it, formula of the scout bee position form is as follows.

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

where, is the dimension of the food source; is the maximum of the dimension of the all food sources; is the minimum of the dimension of the all food sources.

1. Repeat steps 2~4 until the iterations end.
2. Recursive Lest squares estimation

This study uses the Recursive least squares estimation (RLSE) [15] to update the T–S neuron parameters. The RLSE method uses each piece of data when updating parameters, and the recursive update is more effective than the LSE method that receives all the data at a time. In general LSE problems can be considered as a linear problem, as follows.

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

where, is the target; is the model output; {} is the u known equations; {, =1,2,…,m} is the unknown parameter we estimate; is the model error. The LSE problem can also be expressed in the form of a matrix, as follows.

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

where,

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

is the input matrix, is the estimated unknown parameter matrix, is the target matrix, is the error matrix. It can optimize through the RLSE equations.

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

where, is the number of recursive, {}, is the number of data, is the row of , when the RLSE algorithm start, is set to 0, is set to **,**  is a very big positive value, is an identity matrix.

In the hybrid algorithm, different part of the parameter optimization is responsible for different algorithms. This study use PSO-RLSE and ABCO-RLSE hybrid algorithm, PSO and ABCO is for if-part parameter optimization, RLSE is for parameter learning of the linear T-S function. Hybrid algorithm and model computation process are as follows.

1. Prepare the training data and testing data.
2. Use the particle position as the fuzzy set parameters. Input the training data into the model and calculate the firing strength of neuron.
3. Use RLSE to update the parameters of T-S neurons, and in RLSE equations are as follows.

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

where, . In the multi-target prediction, the normalized firing strength is a vector, which makes as a matrix. Therefore, we use identity matrix to replace the constant 1 in the equation (36), the improved equation is as follows.

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

1. After update all parameters, calculate the model output.
2. Calculate the cost, update the particle position in if-part and the relative data.
3. Repeat the steps 2~5 until iterations end.

## Methodology for financial application

In order to evaluate whether the model is helpful to the investment, the use of the cost function is not enough, because cost function means the matching rate, and the high matching rate does not equal high profit. Therefore, this study will predict the closing price in conjunction with the investment strategy [35], and further decide whether to buy or sell, the buying and selling formula is as follows.

|  |  |
| --- | --- |
| buying: if  , |  |
| selling: if  , |  |

where, is the threshold parameter, present the fluctuation of stocks; is the model output, means the predicted closing price of the day; is the actual closing price of the dat. If the predicted closing price is higher than the closing price, means investor has to buy, vice versa.

In order to make investment strategy more cautious, this paper proposes a new strategy as follows.

1. Use formulas (43)-(44) to decide buying or selling, and go to the second phase.
2. Calculate the mean of the past 30 days of the day as the standard of the second phase. Note that after many tests the average of the past 30 days is best.
3. If , and , then buying; if , and , then selling.
4. All trading days need to pass two phase check, if failed in any phase, then not operation.

The profit is calculated by the today closing price and the next day closing price, as follows.

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

where, is the model profit, is the total days when strategy is buying; is the total days when strategy is selling; is the actual closing price of the day.

Through the investment strategy and the profit formula, we can calculate the profits brought by the model to simulate the effect of real-world investment. This study will show the result of two strategies and the other parameters.

# Expeimentation

There are a total of three experiments in this study. The first experiment is a two-target prediction. The target is the Taiwan stock exchange capitalization weighted stock index (TAIEX) and the Hang Seng index (HSI); experiment 2 is the prediction for four targets, test whether the model is feasible by complex-valued membership degree. The complex-valued outputs are used to predict the four targets. For the first complex-valued output, the real part is responsible for the first target, and the imaginary part is responsible for the second target; For the second output, the real part is responsible for the third target, and the imaginary part is responsible for the fourth target. In this experiment, the model simultaneously predicts TAIEX, Dow Jones industrial average index (DJI), national association of securities dealers automated quotation (NASDAQ) and standard and Poor’s 500 (S&P 500). Experiment 3 is also a prediction of four goals, using the two complex-valued outputs, including the closing prices of Apple, IBM, DELL and Microsoft. The third experiment is four well-known technology companies. In addition, the other experiments are very well-known stock indicators. Among them, TAIEX is a weighted calculation index of Taiwan-listed stocks, which represents the fluctuation of Taiwan-listed stocks; HSI is an important indicator reflecting the Hong Kong stock market. The index is calculated from the market value of 50 HSI constituent stocks; DJI covers 9 major industries such as finance, which is a stock price weighted indicator; NASDAQ is a market-weighted indicator of more than 3,000 stocks which are mostly in the technology industry; The S&P500 is the market capitalization of the top 500 US companies, including 11 industries such as IT. These indicators represent the trend of a country or even global stocks, so accurate prediction can bring a lot of help to investors.

For comparison of the other literatures, we will evaluate the model through the error indicator and calculate the profit after the investment. The cost function and the evaluation use root mean square error (RMSE) as follows.

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

, is the number of data; is the error vector of the data; is the target vector; is the output vector; is the Hermitian transpose, means the matrix elements conjugate after the matrix transposing.

In the part of structure learning, the number of rules in this paper which are the number of neurons in the second layer is between 4 and 15. Because the upper limit in block selection is set to 15, and the lower limit is set to 4.

In the investment, the threshold parameter in all experiments is between 0 and 0.1, because means the fluctuation of the stocks, and the stocks ups and downs limit are 10%. For the best selection, this study set 0.001 as the iterated step, and search the threshold from 0 to 0.1. All experiments will find the best threshold through the training data, and calculate the profit in testing data, if the profit in training phase is 0, the profit in testing phase is set to 0, means not invest.

## Example 1—Double Time Series of Daily Taiwan Stock Exchange Capitalization Weighted Stock Index

In this example, we validate the model's performance in real-world time series data. The four targets used are TAIEX and HSI, the closing price in one year. This paper predicts the closing price of 2002 and compares performance with other literature methods. The total number of data is 248. The data of the first ten months of each year is used as training data, and the rest is used as testing data. The training data will extract 30 features, a total of 60 features, each feature data is 205, the features are sorted by TAIEX and HSI. The model proposed in this paper can have multiple complex-valued outputs at a time, so we can predict multiple targets. The real part of the first complex-valued target used in the example is the closing price of TAIEX, and the imaginary part is HSI. The parameter after structure learning are shown as Table I, the number of premises is 15, before the block selection the number of blocks is 625, means the structure learning has a great effect of model size control. Machine learning setting is shown as Table II. In order to validate the stability of the model, this paper runs the model 10 times. The performance is shown as Table III, the profit is shown as Table IV. This experiment will compare with the methods proposed in other literatures, the performance comparison is shown as Table V, the profit comparison is shown as Table VI. The learning curve is shown as Fig. 4. The target and the model output are shown as Fig. 5.

Model Setting (Experiment 1)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | {3, 3, 3, 3} |
| Number of outputs (complex-valued)\* | 2 |
| Type of premises | SCFS |
| Number of premises (after selection) | 15 |
| Number of premise parameters | 48 |
| Number of consequences | 15 |
| Number of consequence parameters | 75 |

\* 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 position | Given by SC algorithm in section II |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 75x1 zero vector |
|  | **I** |
| **I** | 75x75 identity matrix |

Ten Trials Comparison (Experiment 1, RMSE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |
| TAIEX | HSI | TAIEX | HSI |
| 1 | 96.96 | 83.05 | **85.25** | **80.62** |
| 2 | 90.68 | 86.59 | 87.82 | 86.38 |
| 3 | 104.11 | 186.15 | 36748 | 32343 |
| 4 | 109.61 | 112.37 | 4352.6 | 2340.3 |
| 5 | 87.71 | 93.62 | 88.69 | 84.88 |
| 6 | 211.39 | 313.15 | 84.30 | 85.56 |
| 7 | 94.29 | 89.77 | 2093.3 | 1365.1 |
| 8 | **89.77** | **85.72** | 90.01 | 85.14 |
| 9 | 183.06 | 221.61 | 90.07 | 85.68 |
| 10 | 83.27 | 103.40 | 91.43 | 89.47 |

Investment Profit (Experiment 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| Profit | | Profit | | Profit | | Profit | |
| Best | 0.07 | | 0.05 | | 0.047 | | 0.052 | |
| 1 | -1786.5 |  | 643.46 |  | 129.68 |  | 25.08 |  |
| 2 | 88.62 |  | 806.72 |  | 682.75 |  | 157.62 |  |
| 3 | -463.39 |  | -36.57 |  | 475.26 |  | 202.11 |  |
| 4 | -914.08 |  | 1664.9 |  | 64.19 |  | 82.35 |  |
| 5 | -936.58 |  | -355.81 |  | 173.98 |  | **961.18** |  |
| 6 | -173.21 |  | -689.96 |  | 14.83 |  | -3.451 |  |
| 7 | -1070.4 |  | -911.59 |  | **767.09** |  | 59.93 |  |
| 8 | **2524** |  | -1640.9 |  | 64.19 |  | -30.80 |  |
| 9 | -376.84 |  | 1620.8 |  | 119.86 |  | 514.65 |  |
| 10 | 124.96 |  | **2469.8** |  | 364.09 |  | 478.03 |  |
| Average | -298.34 | | 357.09 | | 350.48 | | 244.67 | |
| Std. | 1151.83 | | 1308.77 | | 345.16 | | 315.57 | |
| Maximum | 2524 | | 2469.8 | | 767.09 | | 961.18 | |
| Minimum | -1786.5 | | -1640.9 | | 14.83 | | -30.8 | |

\*The result of the investment strategy proposed in this paper

Performance Comparison (RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | HSI |  | TAIEX | HSI |
| Chen [35] | - | - |  | 75 | 187 |
| Yu [35] | - | - |  | 101 | 170 |
| AR(1) [35] | - | - |  | 66 | 105 |
| SVR [35] | - | - |  | 66 | 107 |
| ANFIS [35] | - | - |  | 65 | 106 |
| ANFIS (EMD) [35] | - | - |  | 52 | 97 |
| PSO-RLSE (proposed) | **99.16** | **98.19** |  | **89.77** | **85.72** |
| ABCO-RLSE (proposed) | **98.77** | **97.12** |  | **85.25** | **80.62** |

Investment Profit Comparison (Experiment 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Chen [4] | Yu [4] | SR+ANFIS [4] | SR+SVR [4] | Elman [4] | Cheng et al. [4] | PSO-RLSE | ABCO-RLSE |
| Best | 0.012 | 0.012 | 0.012 | 0.012 | 0.012 | 0.012 | 0.07 | 0.05 |
| Profit (TAIEX) | 0 | 0 | 0 | 0 | 0 | -231.02 | **1284.2** | 847.8 |
| Profit (HSI) | -1471 | -1368 | -602.94 | 190.71 | **2342** | 1793.12 | 1239.7 | 1622.0 |



(a)



(b)

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



(a)



(b)



(c)



(d)

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

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

In this example, the used targets are TAIX, DJI, NASDAQ and S&P500. This paper predicts the closing price of 2001 and compares performance with other literature methods. The data of the first ten months of each year is used as training data, and the rest is used as testing data. The training data is 181, and the testing data is 66. The training data will extract 30 features, a total of 120 features for four targets, the features are sorted by TAIEX, DJI, NASDAQ and S&P500. The model proposed in this paper can have multiple complex-valued outputs at a time. In this experiment has two target, the real part of the first complex-valued target is the closing price of TAIEX, and the imaginary part is DJI, the real part of the second target is NASDAQ, and the imaginary part is S&P500. In the structure learning part, the number of the premise is reduced from 81 to 9 by the block selection, which significantly reduces the size of the model and the model parameters. The model parameters are shown as Table VII. Machine learning setting is shown as Table VIII. In this experiment, except the comparison between PSO-RLSE and ABCO-RLSE, also compares with other literature methods, such as ANFIS[22], CNFS-ARIMA[22], RBF network[22] and SVR[22]. Except the SVR, the other models can predict two targets at the same time. So we use the first complex-valued output to compare with other methods, the result is shown as Table IX, the profit is shown as Table X. In order to validate the model stability, this paper runs model 10 times, result is shown as Table XI. Learning curve of machine learning is shown as Fig. 6; targets and the model output are shown as Fig. 7.

Model Setting (Experiment 2)

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

\* 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 position | Given by SC algorithm in section II |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 45x1 zero vector |
|  | **I** |
| **I** | 45x45 identity matrix |

Performance comparison (RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | DJI |  | TAIEX | DJI |
| SVR (two models, each with single output) [22] | - | - |  | 162.46 | 101.44 |
| ANFIS (two models, each with single output) [22] | - | - |  | 147.36 | 105.56 |
| ANFIS (one model with two outputs) [22] | - | - |  | 151.62 | 128.20 |
| RBF (two models, each with single output) [22] | - | - |  | 134.32 | 106.33 |
| RBF (one model with two outputs) [22] | - | - |  | 137.58 | 181.79 |
| CNFS(4)-ARIMA (one model with two outputs) [22] | - | - |  | 115.82 | 103.06 |
| PSO-RLSE (proposed) | **92.92** | **91.50** |  | **96.51** | **92.18** |
| ABCO-RLSE (proposed) | **93.69** | **91.06** |  | **87.92** | **91.95** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Profit Comparison | | | | | | | | |
|  | Chen [35] | Yu [35] | AR(1) [35] | SVR [35] | ANFIS [35] | Wei [35] | PSO-RLSE | ABCO-RLSE |
| Best | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.047 | 0.012 |
| Profit (TAIEX) | -92 | -73 | 671 | 202 | 686 | **795** | 745.1 | 458.48 |



(a)



(b)

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



(a)



(b)



(c)



(d)

1. Actual values and model output. (a) PSO-RLSE (TAIEX) (b) PSO-RLSE (DJI) (c) ABCO-RLSE (TAIEX) (d) ABCO-RLSE (DJI)

Ten trials performance (Experiment 2, RMSE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |
| TAIEX | DJI | TAIEX | DJI |
| 1 | 98.95 | 108.39 | **87.92** | **91.95** |
| 2 | 94.15 | 94.49 | 89.21 | 94.14 |
| 3 | 93.24 | 97.72 | 88.69 | 93.54 |
| 4 | 335.68 | 464.85 | 102.07 | 92.31 |
| 5 | 114.71 | 122.97 | 86.22 | 94.15 |
| 6 | **96.51** | **92.18** | 88.19 | 94.27 |
| 7 | 87.59 | 102.31 | 88.96 | 93.62 |
| 8 | 90.24 | 95.65 | 87.44 | 93.98 |
| 9 | 151.31 | 148.51 | 88.70 | 93.91 |
| 10 | 96.29 | 121.39 | 88.47 | 93.48 |

Investment Profit (Experiment 2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| Profit | | Profit | | Profit | | Profit | |
| Best | 0.047 | | 0.012 | | 0.061 | | 0.012 | |
| 1 | -264.81 |  | -274.69 |  | 0 |  | 0 |  |
| 2 | -168.32 |  | 359.55 |  | 77.76 |  | 0 |  |
| 3 | 105.9 |  | 916.80 |  | -21.34 |  | 298.98 |  |
| 4 | -1636.8 |  | -862.40 |  | **363.20** |  | 82.62 |  |
| 5 | -378.64 |  | -274.69 |  | 277.37 |  | 0 |  |
| 6 | 532.59 |  | -274.69 |  | -34.84 |  | 0 |  |
| 7 | **987.72** |  | 359.55 |  | 10.77 |  | 0 |  |
| 8 | -235.02 |  | **916.80** |  | -10.67 |  | **298.98** |  |
| 9 | 412.35 |  | -274.69 |  | 0 |  | 0 |  |
| 10 | 199.98 |  | 359.55 |  | 114.98 |  | 0 |  |
| Average | -44.50 | | 95.11 | | 77.72 | | 68.06 | |
| Std | 702.12 | | 579.65 | | 137.28 | | 124.40 | |
| Maximum | 987.72 | | 916.80 | | 363.20 | | 298.98 | |
| Minimum | -1636.8 | | -862.40 | | -34.84 | | 0 | |

\*The result of the investment strategy proposed in this paper.

## Example 3—Quadruple Time Series of Apple, IBM, Dell and Microsoft

In this experiment, the predicted targets are the closing price of APPLE, IBM, Dell and Microsoft, the period is from 10th February 2003 to 21th January 2005, the number of data is 492. In order to compare with other methods, we use the data from 10th February 2003 to 10th September 2004 as the training data which is 400. And the rest is testing data. In this paper, 30 features were extracted from the training data, and there were 120 features for 4 targets. The features were sorted by APPLE, IBM, Dell and Microsoft. The model proposed in this paper can have multiple complex-valued outputs, so can predict multiple targets. The real part of the first target used in this experiment is closing price of IBM, the imaginary part is APPLE, the real part of the second target is DELL, and the imaginary part is Microsoft. In the part of structure learning, through block selection, 8 blocks can be selected from the original 81 blocks to reduce the number of parameters in the model. The rest of the model settings are shown in Table XIII. Machine learning setting is shown as Table XIV. The result of this experiment compares with other literature methods, such as HiMMI [32], ANN-GA-HMM-Interpolation [32], ANN-GA-HMM-WA [32] and ARIMA [32]. So we use the real and imaginary part of the first complex-valued output to compare. The comparison result is shown as Table XV. The investment profit is shown as Table XVI. The learning curve of the model is shown as Fig. 8; The targets and the mode output are shown as Fig. 9.

Model Setting (Experiment 3)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | {2, 3, 3, 3} |
| Number of outputs (complex-valued)\* | 2 |
| Type of premises | SCFS |
| Number of premises | 8 |
| Number of premise parameters | 44 |
| Number of consequences | 8 |
| Number of consequence parameters | 40 |

\* Each complex-valued target whose real and imaginary part 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 position | Given by SC algorithm in section II |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 40x1 zero vector |
|  | **I** |
| **I** | 40x40 identity matrix |



(a)



(b)

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

Ten Trails Performance (Experiment 3, MAPE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |
| APPLE | IBM | APPLE | IBM |
| 1 | 1.7948 | 0.8043 | 1.7625 | 0.8032 |
| 2 | 1.7972 | 0.8039 | 1.7835 | 0.8042 |
| 3 | 1.7963 | 0.8033 | 1.8273 | 0.8002 |
| 4 | 1.7954 | 0.8043 | 1.7818 | 0.8015 |
| 5 | **1.7776** | **0.8033** | 1.7922 | 0.8033 |
| 6 | 1.7921 | 0.8044 | 2.0309 | 0.8002 |
| 7 | 1.7942 | 0.8064 | **1.7602** | **0.8036** |
| 8 | 1.7967 | 0.8043 | 1.7946 | 0.8031 |
| 9 | 1.7920 | 0.8042 | 1.8567 | 0.8046 |
| 10 | 1.7927 | 0.8041 | 1.7758 | 0.8027 |

Investment Profit (Experiment 3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| Profit | | Profit | | Profit | | Profit | |
| Best | 0.011 | | 0.012 | | 0.021 | | 0.021 | |
| 1 | -4.5715 |  | -10.08 |  | 0 |  | -0.359 |  |
| 2 | 0.94 |  | 1.361 |  | 0 |  | 0 |  |
| 3 | -14.06 |  | -8.678 |  | **1.022** |  | -2.764 |  |
| 4 | -3.37 |  | **2.061** |  | 0 |  | -1.441 |  |
| 5 | **12.22** |  | -8.678 |  | -1.163 |  | **-2.764** |  |
| 6 | 8.72 |  | 2.061 |  | 0.176 |  | -1.441 |  |
| 7 | -10.18 |  | -0.3700 |  | -0.334 |  | 0 |  |
| 8 | 0.94 |  | -8.678 |  | 0 |  | -2.764 |  |
| 9 | -14.06 |  | 2.061 |  | 1.022 |  | -1.441 |  |
| 10 | -3.37 |  | -0.37 |  | 0 |  | 0 |  |
| Average | -2.6794 | | -2.9308 | | 0.072 | | -1.2974 | |
| Std | 8.8075 | | 5.3385 | | 0.626 | | 1.1758 | |
| Maximum | 12.22 | | 2.061 | | 1.022 | | 0 | |
| Minimum | -14.06 | | -8.678 | | -1.163 | | -2.764 | |

\*The result of the investment strategy proposed in this paper.



(a)



(b)



(c)



(d)

1. Actual values and the model output. (a) PSO-RLSE (APPLE) (b) PSO-RLSE (IBM) (c) ACO-RLSE (APPLE) (d) ACO-RLSE (IBM) (e) ABCO-RLSE (APPLE) (f) ABCO-RLSE (IBM)

Performance Comparison (MAPE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **MAPE** | | | | |
| Training phase | |  | Testing phase | |
| APPLE | IBM |  | APPLE | IBM |
| HiMMI [32] | - | - |  | 2.8373 | 1.2186 |
| ANN-GA-HMM-Interpolation [32] | - | - |  | 2.1649 | 1.0555 |
| ANN-GA-HMM-WA [32] | - | - |  | 1.9247 | 0.8487 |
| Bayesian ANN [32] | - | - |  | 1.9688 | 0.7441 |
| ARIMA [32] | - | - |  | 1.8009 | 0.9723 |
| PSO-RLSE (proposed) | **1.7840** | **1.1960** |  | **1.7776** | **0.8033** |
| ABCO-RLSE (proposed) | **1.7654** | **1.1958** |  | **1.7602** | **0.8036** |

# Discussion

Through three experiments, we can find that the series of methods proposed in this paper have multi-target prediction ability. In the performance of the prediction, it can be found from Experiment 1 that the prediction performance of the dual target is better than that proposed by other literatures. Representing the model of this paper, even if it predicts two goals at a time, it can have excellent performance. It predicts the four goals in experiment 2 and experiment 3, and proves the prediction ability of the model again. Through multi-target feature selection, features that are beneficial to all targets can be extracted from more than 100 features, and training data can be applied efficiently. In the structure learning part, the first experiment selects 15 useful blocks from the 625 blocks as the model premise neurons, in the second and third experiments, the blocks are reduced from 81 blocks. The number of the parameters in the model, is reduced objectively, the size of the model is effectively controlled, and the burden of the model computation is reduced. This is one of the advantages of this model. In the parameter learning part, it can be seen from the performance that the hybrid algorithm has a good performance for parameters searching, and the matching rate is better than other literature methods. Among them, ABCO-RLSE is better than PSO-RLSE, showing the advantages of the ABCO algorithm. Through local search by onlooker bees and the jump out mechanism by scout bees, the ABCO performance is better than PSO. Finally, in the simulation investment part, the profit of the model proposed in this paper in some experiments is better than the past literature, and the experiment lost to other literatures is close. It can be found from the 10 trials performance tables and the investment income statement that there is no positive correlation between performance and investment profit. The investment strategy proposed in this paper improves the standard of decision-making in investment, and judges daily transactions more cautiously. From the results, it can be found that the profit earned by the investment strategy has increased, and the standard deviation has decreased in each experiment. It means better stability when investing, which can be provided to investors for decision-making.

# Conclusion

This paper proposes a hybrid algorithm ABCO-RLSE, which has better performance than PSO-RLSE, both of which have multi-target prediction capabilities. It can be seen from the experiment that both the two targets and the four targets can be predicted at the same time. Through the concept of the sphere complex fuzzy set, even more targets can be predicted at one time. Multi-target feature selection can extract useful data from the original data according to different data, and control the size of the data entering the model, which is an important process for a large amount of data. In the part of structure learning, uses the data-driven concept to automatically adjust the input data according to the data, and the model can adapt to generate different structures when faced with different data. From the 3 experiments, it can be found that different data can be effectively predicted in this model, which means two algorithms combined with RLSE has a certain level. Finally, the investment strategy proposed in this paper helps to reduce the risk of investment and provides an objective reference data for investors.

The part of machine learning may be limited by the characteristics of the algorithm. For example, PSO converges quickly and easily fall into the local minimum. Therefore, in the case of a large amount of data, the required search dimensions are increased, and the PSO performance will be poor, which may limit the overall performance of the model. In the case of ABCO, if it falls into the local minimum, it will re-search after the limit iterations arrives. Relatively speaking, the probability of jumping out the local minimum is higher for ABCO, but the formula for re-searching is through the maximum and minimum values, the initial values of all bees are very important, which will deeply affect the training of the model and make model unstable. In the future, different machine learning algorithms can be combined with SCFNS, such as random search [37], ant colony algorithm [6], etc., to overcome the problems encountered by current algorithms, and also through swarm intelligence concept, such as CPSO [33], etc. looks for the best solution. In terms of simulation investment, we can learn from experiments that each algorithm can make a profit, but the smaller the error of the model, not means the more profit it earns, the two are not positively related. In the investment strategy, the threshold parameter is the most important, so in the future, the threshold parameter can be added to the machine learning to increase the profit brought by the model.

##### ACKNOWLEDGMENT

##### REFERENCES

[1] J. Buckley, "Fuzzy complex numbers," *Fuzzy Sets and Systems,* vol. 33, no. 3, pp. 333-345, 1989.

[2] J. Buckley and Y. Qu, "Fuzzy complex analysis I: differentiation," *Fuzzy Sets and Systems,* vol. 41, no. 3, pp. 269-284, 1991.

[3] J. J. Buckley, "Fuzzy complex analysis II: integration," *Fuzzy Sets and Systems,* vol. 49, no. 2, pp. 171-179, 1992.

[4] C.-H. Cheng and J.-H. Yang, "Fuzzy time-series model based on rough set rule induction for forecasting stock price," *Neurocomputing,* vol. 302, no. 9, pp. 33-45, 2018.

[5] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & fuzzy systems,* vol. 2, no. 3, pp. 267-278, 1994.

[6] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," presented at the Proceedings of the first European conference on artificial life, 1991.

[7] P. du Jardin, "Dynamics of firm financial evolution and bankruptcy prediction," *Expert Systems with Applications,* vol. 75, no. 1, pp. 25-43, 2017.

[8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95., Proceedings of the Sixth International Symposium on Micro Machine and Human Science* 1995, pp. 39-43: IEEE.

[9] Z. Feng, L. Jin, D. Tao, and S. Huang, "DLANet: a manifold-learning-based discriminative feature learning network for scene classification," *Neurocomputing,* vol. 157, no. 1, pp. 11-21, 2015.

[10] N. Fumo and M. R. Biswas, "Regression analysis for prediction of residential energy consumption," *Renewable and Sustainable Energy Reviews,* vol. 47, pp. 332-343, 2015.

[11] H. Greenspan, B. Van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging,* vol. 35, no. 5, pp. 1153-1159, 2016.

[12] J. Heaton, N. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry,* vol. 33, no. 1, pp. 3-12, 2017.

[13] L. J. Herrera *et al.*, "Multigrid-based fuzzy systems for time series prediction: CATS competition," *Neurocomputing,* vol. 70, no. 13-15, pp. 2410-2425, 2007.

[14] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.

[15] C. T. S. a. E. M. J. S. R. Jang, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence prentice hall*. Upper Saddle River, 1997.

[16] R. Ž. Jovanović, A. A. Sretenović, and B. D. Živković, "Ensemble of various neural networks for prediction of heating energy consumption," *Energy and Buildings,* vol. 94, no. 1, pp. 189-199, 2015.

[17] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization,* vol. 39, no. 3, pp. 459-471, 2007.

[18] C. Li and T.-W. Chiang, "Complex neuro-fuzzy self-learning approach to function approximation," in *Asian Conference on Intelligent Information and Database Systems*, Berlin, Heidelberg, 2010, pp. 289-299: Springer.

[19] C. Li, T.-W. Chiang, J.-W. Hu, and T. Wu, "Complex neuro-fuzzy intelligent approach to function approximation," in *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on*, 2010, pp. 151-156: IEEE.

[20] C. Li and T.-W. Chiang, "Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach," in *Asian Conference on Intelligent Information and Database Systems*, 2011, pp. 242-251: Springer.

[21] 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, 2012.

[22] C. Li and T.-W. Chiang, "Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems,* vol. 21, no. 3, pp. 567-584, 2013.

[23] C. Li, "Multi-target feature selection," *unpublished,* 2017.

[24] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics,* vol. 5, no. 4, pp. 115-133, 1943.

[25] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE Transactions on Fuzzy Systems,* vol. 10, no. 2, pp. 171-186, 2002.

[26] V. Ravi, D. Pradeepkumar, and K. Deb, "Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms," *Swarm and Evolutionary Computation,* vol. 36, pp. 136-149, 2017.

[27] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *IRE Transactions on information Theory,* vol. 2, no. 3, pp. 80-93, 1956.

[28] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction," *Fuzzy sets and systems,* vol. 157, no. 9, pp. 1260-1275, 2006.

[29] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review,* vol. 5, no. 1, pp. 3-55, 2001.

[30] 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, 2008.

[31] 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, 1985.

[32] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications,* vol. 40, no. 14, pp. 5501-5506, 2013.

[33] 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, 2004.

[34] J. Vieira, F. M. Dias, and A. Mota, "Neuro-fuzzy systems: a survey," in *5th WSEAS NNA international conference on neural networks and applications, Udine, Italia*, 2004.

[35] L.-Y. Wei, "A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting," *Applied Soft Computing,* vol. 42, pp. 368-376, 2016.

[36] P. Werbos, "Beyond regression: new fools for prediction and analysis in the behavioral sciences," *PhD thesis, Harvard University,* 1974.

[37] Z. B. Zabinsky, "Random search algorithms," *Wiley Encyclopedia of Operations Research and Management Science,* 2010.

[38] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research,* vol. 17, no. 1-32, p. 2, 2016.

[39] L. A. Zedeh, "Fuzzy sets," *Information and control,* vol. 8, no. 3, pp. 338-353, 1965.

[40] M. Zounemat-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, 2008.