Title

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*—Stock volatility is a time series of data. Time series prediction is an important research topic, and artificial intelligence is currently being widely used in this topic, such as: neuro-fuzzy systems. This paper proposes a complex neuro-fuzzy system and applies it to multi-target time series prediction. This model has multiple complex-valued outputs. For each complex-valued output, the real part and the imaginary part can be predicted separately for two different real-valued targets. For the feature selection, this study uses multi-target feature selection to screen out the features that are beneficial to all targets, and use them as a model input to reduce the overall model computational burden and improve the efficiency of data application. In terms of model, a multi-layer neural network is constructed by input layer, complex fuzzy sets layer, premise neural layer, Takagi-Sugeno neural layer and output layer. In terms of parameter learning, we use the divide-and-conquer principle when training model. The parameters of the complex fuzzy sets layer are optimized using different algorithms, such as particle swarm optimization (PSO) and artificial bee colony optimization (ABCO); The parameters of the T-S neural layer are optimized using recursive least-squares estimation (RLSE); other neural layers have no parameters to be optimized. In terms of experiments, we designed three experiments to test the performance of the model, combining the PSO-RLSE and ABCO-RLSE experimental results with the investment strategy, the calculated model profits are compared with each other and compared with different literature methods. This study proposes a new investment strategy, compared with the past investment methods, and through the result of the performance comparison and the profit comparison. The multi-target prediction method proposed in this paper shows excellent performance and investment effect.

Keywords—Time series; Complex fuzzy set; Complex neuro-fuzzy system; Artificial bee colony optimization; Particle swarm optimization; RLSE;

# Introduction

Time series data is a sequence of data arranged in order of time. For example, the stock price and exchange rate are all time series data. The time series can be regarded as data observed in discrete time order, and can be analyzed by mathematical methods or statistical models, which are commonly used method today. With the rapid development of the Internet and the advancement of technology, the amount of information has increased rapidly. Using artificial intelligence, such as machine learning or data mining, the hidden relationships or rules hidden behind the data can be found, which cannot be found by human intelligence.

The prediction of time series data is widely used in various fields, and there is many research, such as financial exchange rate, stock ups and downs, etc.; In addition to financial field, energy consumption, disease prediction and so on can be used in the allocation of the resources, which helps countries or institutions to make effective resource allocation. Among the above-mentioned fields, the most relevant to us is the financial forecast. How to effectively use the money investment to make profits is a topic worth exploring. Time series data forecasting in the financial economy is the most difficult because it includes many influence factors, such as company situations, global situation, and overall economic environment. Many factors make financial time series highly variable. Therefore, if there is a model with good prediction performance which has effective input data and correct investment strategy, can help investors get profits from it.

In the prediction of finance, the neural network is the most widely used method. Since 2005, there have been many research outputs on financial issues in the real world. For example, Zhi-Bin et al. [44] used the adaptive neuro-fuzzy inference system (ANFIS) and the artificial neural network (ANN) to predict annual excess returns and compare performance. He found that both predictions are very good; In 2006, Patel et al. [31] used ANFIS, fuzzy inference system and ANFIS to assist investors in making economic decisions; Yao [41] proposed a method in 2007 to deal with foreign currency trading strategies, profiting from forecasting exchange rates; Li et al. [25] used group intelligence and self-organized neuro-fuzzy system (NFS) to predict the exchange rate between RMB and US dollars; In addition to the exchange rate, stock forecasts also have many research literatures. For example, Abbasi used ANFIS in 2008 with four independent variables, trading volume, price-earnings ratio and earning per share forecasting stock closing price [1]; in 2017, Koijen et al. [22] predicted each other's data through stocks and bonds; Pan et al. [30] compared short-term and long-term leverage by predicting stock volatility.

In the stock market, there are several famous theories, one of which is the random walk theory proposed by Kendall et al. [18] in 1953, which means that the stock price changes are independent, and there is no regularity or period; Its extension is efficient market hypothesis (EMH) [11], which was proposed by scholar Fama in 1970, this theory holds that the transaction price is acceptable to both the buyer and the seller, and that all participants in the market can obtain information without compensation, so the information collected by the investor cannot make excessive profits. But other experts believe that stocks are predictable, so they research continuously. In 1990, Kimoto et al. [21] used the back propagation neural network (BPNN) to match the two indicators to predict the rise and fall of the Nikkei and the timing of buying and selling; In 1999, Yao et al. [40] used a number of technical indicators to predict the stock market by using back propagation neural networks; Kim et al. [20] added the genetic algorithm (GA) to the neural network in 2000, which not only improved the learning speed of the neural network, but also reduced the complexity of the feature space. The back propagation neural network appears to be more excellent, and it is found that the nonlinear neural network has better predictive ability; In 2012, Wei [38] proposed an ANFIS based on empirical mode decomposition (EMD), and calculated the profit with investment strategy. In virtual investment, there is a good profit-making effect; In the same year, Li et al. [26] used the autoregressive integrated moving average (ARIMA) model combined with the neuro-fuzzy system (NFS) to predict the closing price of stocks. The following year, Li et al. [27] used a complex neuro-fuzzy system (CNFS) to simultaneously predict dual targets.

As can be seen from the above, the NFS is very useful for predicting financial aspects, and a well-trained model performance is much more accurate. Therefore, this study will use the NFS to combine with other artificial intelligence techniques to predict multiple targets. In the process of model training, problems such as prediction bias or long-term model operation often occur. Therefore, how to select effective input data and control the size of the model is the main topic of most research. In the process of forecasting, the parameters need to be optimized to increase performance. Many machine learning algorithms are currently used to find the optimal solution of parameters. Such as particle swarm optimization (PSO) [19], artificial bee colony optimization (ABCO) [17], gene algorithm (GA) [10] and so on. At present, most of the literatures are based on the Takagi-Sugeno fuzzy system [36]. Therefore, there are many parameters that need to be optimized, which will make the algorithm parameters too large, which causes the effect less than expected. Besides, it makes the algorithm convergence too slow and fall into the local minimum.

In this study, a hybrid algorithm is used to optimize the parameters. The algorithm PSO and ABCO are respectively combined to recursive least squares estimation (RLSE) algorithm [16] to form PSO-RLSE [23] and ABCO-RLSE. Different algorithms are used to optimize the parameters of different parts by the divide-and-conquer method, which reduces the probability that machine learning will fall into the local minimum when seeking the solution. As mentioned above, stock volatility is formed by many factors. Therefore, this study will select features from multiple stocks, and predict multiple targets at the same time, extracting content that is beneficial to each target from a large amount of data. In order to improve the prediction effect, the neural network system is combined with the fuzzy rule of T-S (Takagi-Sugeno) type to form a neuro-fuzzy model. And the parameters of the neuro-fuzzy model are divided into two parts, namely the if-part parameters and the then-part parameters.

Traditionally, the design of fuzzy sets is based on mathematic functions to establish fuzzy sets [43]. After data input, the membership degree which is between 0 and 1 can be obtained. In 2002, Ramot proposed complex fuzzy sets (CFSs) [32], adding an imaginary part to the original membership degree. The membership degree of CFSs is that traditional one-dimensional space projecting to the unit complex disc which the axis x is real part and the axis y is imaginary part. This makes it have a richer information than the original real-type membership function.

This study proposes a hybrid algorithm, PSO-RLSE and ABCO-RLSE, to optimize model parameters. There are two part parameters in the model, namely the if-part parameters and the then-part parameters. PSO and ABCO are responsible for optimizing the if-part parameters, and RLSE is to optimize the then-part parameters, to reduce the search dimension of each algorithm.

In this study, firstly, we use the concept of Shannon information entropy [35] to analyze data and make multi-target feature selection [28]. By calculating the amount of information provided by the original data to the target as a standard, and adding the calculation of redundant information, selecting the data that has a good influence on the target as the basis for input to the model can reduce the data complexity and the computation cost. Then, through the PSO-RLSE and ABCO-RLSE algorithms update the parameters of the if-part and the parameters of the then-part. Finally, the results are combined with the investment strategy to calculate the profit brought by the model.

This study is divided into six sections. The section I is the introduction, introducing the background, motivation and purpose of the research, and summarizing the methods and processes used in this study. The section II introduces and summarizes the theoretical methods, including some methods of past feature selection. In addition, the origin of fuzzy sets and the concept of complex fuzzy sets will be introduced. Then introduce the theory of neural network, and related research, and finally mention the widely used neuro-fuzzy system. The section III is the system design and architecture, an overview of multi-target feature selection, and a method of machine learning to determine the size of the model. The introduction of the various neural layers in a complex neuro-fuzzy network is also described in this section. Then the algorithm of parameter learning will be described in detail, including PSO, ABCO, and RLSE, and the process of hybrid algorithm in training phase. Finally, it will mention the operation and concepts of past investment strategies, as well as introduce the new investment strategy methods and new method of calculating profits. The section IV is the experiment and the results. In this study, three experiments were used to study the time series prediction. Experiment 1 predict single target, the second experiment simultaneously forecasts two targets, the third experiment is a forecast for four targets. The above experiments will make two comparisons, one is the performance comparison of the two different hybrid algorithms in the study, and the profit comparison with the investment strategy proposed in the past and the investment strategy proposed in this study. In addition, the performance comparison with the methods proposed in the past literature will be carried out to confirm the pros and cons of the research method. The section V discusses the results of the experiment. The sections VI is the conclusion of this study and the future research direction.

# Literature Review

## Feature Selection

The data calculation model is constructed through machine learning or data mining, such as classification, regression, and clustering. There are two purposes for preprocessing data. One is to reduce the size of the data set, making the analysis more effective. The other is to select the appropriate data for a specific data set. For today's big data, the former is very important for the entire analysis method, so feature selection plays an indispensable role in many fields such as data mining, network anomaly detection, text classification, genetic analysis and so on. The reduction of feature sets is based on the relevance of the features and their redundancy to the target [42]. Feature selection methods can be classified into several [14], the most common are the filter approach [12], the wrapper approach [2], the embedded method [12], and the Hybrid approach [15]. The following is a brief description of each method.

The filter approach was proposed by Guyon [12] in 2003. This method first selects the feature set of the data set, then evaluates and scores the selected feature subsets, and finally sets the threshold value. The process of feature selection is independent of models, which has the advantage of being faster than the wrapper approach. In 1997, Blum proposed wrapper approach [2], the biggest difference from the filter approach is the selection of the feature sub-sets. The scoring criteria are determined by the results of the model. So in the selection process, it will continuously run model computation, which have long calculation time and complicated calculation. For some specific classifiers, the wrapper approach has high classification accuracy. The embedded approach [12] combines feature selection with the training of the model. Therefore, feature selection is also completed after the model training ends. The feature selection time and model computation are between the above two methods. The hybrid approach [15] combines the concepts of the filter approach and the wrapper approach, selects appropriate feature subsets through the filter approach, and then uses these feature subsets in model to select the final feature. This method improves the accuracy of the filter approach and reduces the computational time of the wrapper method, combining the advantages of both.

As mentioned above, the reduction of feature sets is based on the relevance and redundancy of features. Therefore, this study uses concept of entropy to calculate the amount of information provided by features. The word entropy was first proposed by the German physicist Rudolph Clausius in 1854 [8], it is a measure of the disorder of the physical system. When the entropy value is higher, the degree of disorder is higher. Regarding the disorder of information, Shannon proposed the theory of Shannon information entropy [35]. If the randomness of information is higher, the information entropy value will be higher. For a random variable , information entropy The definition is as follows.

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

where, ， is the entropy of the random variable ; is the probability of the event ; is regarded as the disorder information of the . It can be found that the greater the probability of event occurrence is, the smaller the information entropy value is. Conversely, if the probability of an event tends to average, the entropy will approach the maximum. As mentioned before, the larger the entropy value, the higher the randomness of the information provided. This study is based on the theory of Shannon information entropy [35] to make multi-target feature selection [28]. This method takes into account the information and the redundant information. Finally, the user's settings are used to filter out several features as input to the model.

## Complex Fuzzy Sets

The concept of the crisp set is only “belonging” or “non-belonging”, but this concept cannot be applied to the real world. For example, if the temperature is greater than 30 degrees, it is called “heat”, but 29.9 degrees is not much different from 30 degrees but it is not “hot”. This situation conflicts with human thinking. In 1965, Zadeh proposed the concept of fuzzy sets [43]. By this concept, we can calculate the membership degree of the elements belonging to the set through a membership function, and the membership degree is between 0 and 1. The higher membership degree, the higher the level to which the element belongs to the collection. Afterward, Buckley proposed the complex number [3-5]. In 2002, Ramot et al. [32] further proposed complex fuzzy sets (CFSs), which means that the complex-valued membership degree can be obtained by a function, which allows the membership degree to be presented in a complex unit disc with a radius of 1. This concept enriches the membership degree.

## Neural Network

Neural network is a technique that mimics the human brain. The human brain contains lots of neurons which main function is to process information and memory. The neurons are connected by "synapses", and the functions of "synapses" are like weights, which control the flow of information. In 1943, McCulloch et al. [29] used an algorithm called threshold logic to simulate the concept of neurons, while a neuron receives information, it will determine whether it is necessary to generate an excitatory response mechanism. Then the psychologist Hebb proposed Hebb's law [13], which means that if the neuron behind synaptic and the neuron in front of synaptic are activated simultaneously, the strength of the connection between the two neurons is increase; This idea was implemented with mathematic model by Rochester et al. [33] in 1956 to create a perceptron to simulate the human brain, which is arguably the ancestor of deep learning. However, the number of neurons is less, and the weight of the transmitted signal is 0 or 1, still cannot handle the XOR problem (non-linear problem), and the computer did not have enough ability to calculate in that time. Until 1975, Werbos [39] proposed back propagation algorithm (BP), which effectively solved the XOR problem and the problem of training multi-layer neural networks. From that time, many other artificial intelligence methods are proposed, such as support vector machine (SVM) [9], Bayesian classifier (Bayesian classifier) [34] and so on. Until recently, due to the rapid development of technology, the speed of computation has increased, and in some areas, neural networks have surpassed humans, neural network was once again heavily studied.

## Complex Neuro-fuzzy system

In the past, many research methods have been proposed such as ARIMA, fuzzy theory, neural network computing, and so on. Among them, the most commonly proposed are neuro-fuzzy systems (NFSs). The neuro-fuzzy system is a fuzzy system, which can be regarded as a combination of neural network theory [29] and fuzzy theory [43]. There are literatures that classify NFSs into three types [37], cooperative NFS, concurrent NFS, and hybrid NFS.

***Cooperative NFSs:*** First, the neural network is used to calculate the required parameters such as rules, and the parameters are brought into the fuzzy system for computation.

***Concurrent NFSs:*** After the data is processed by the neural network, it is regarded as the input of the fuzzy system and the final result is obtained by fuzzy system.

***Hybrid NFSs:*** It is the most widely studied system among the three, which integrates rules and other elements in the fuzzy system into the neural network and corrects the parameters in the system through learning algorithms.

Due to the characteristics of the neuro-fuzzy system, it has a good effect on the prediction of time series. Most of the research on time prediction in this area now uses a neuro-fuzzy system as the model architecture. In this study, a hybrid neural network is used. In order to enrich the information covered by the fuzzy system, this study uses a complex fuzzy set to replace the traditional fuzzy set to form a complex neuro-fuzzy system (CNFS), which allows the model to predict multiple targets simultaneously, the details will be discussed in section III.

# Methodology

This section will explain the method design and model architecture used in the research. In this study, machine learning is used to determine the model structure size. The model is implemented using a CNFS. Different algorithms are used to optimize the parameters of the if-part in the model. The recursive least squares algorithm optimizes the parameters of the then-part. Before the data enters the model, through the multi-target feature selection [28], select the feature data sets that are more effective for all the targets, and reduce the burden of the model. Finally, the results are combined with the investment strategy to make a comparison of different algorithms.

## Complex Fuzzy Sets

The traditional fuzzy set concept [43], the membership degree of an element to a set is one-to-one. A complex fuzzy set can have a richer membership degree. Through this concept, the complex-valued membership degree of a piece of data can be calculated, which makes the model can predict multiple targets at a time. The process of membership calculation is as follows, assuming that there is a complex fuzzy set ,which can be expressed as follows.

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

where, is the membership degree of element , denoted as follows.

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

where,  is value variable of universal set ; is the amplitude function, it is real-valued between 0 to 1; is the phase function, it is real-valued; is .

This study uses a Gaussian complex fuzzy set, which was proposed by Li et al. [24], is a combination of complex fuzzy set and Gaussian function. Complex Gaussian membership function (cGMF) is denoted as follows.

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

where, are input data, center and spread of the fuzzy set, is the phase frequency parameter, this parameter enters the parameter learning to increase the flexibility of the model. is the first order differential of Gussian function, the purpose is to re-use Gaussian original parameters and reduce the complexity of the parameters during the computation. Through the complex Gaussian fuzzy set, a complex degree can be obtained. We can decompose to obtain a membership degree vector and the components are expressed as follows.

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

where, is the real part of the value; is the imaginary part of the value; is Gussian function as formula (5); is the first order differential of Gaussian function as formula (6). Through the above disassembly, the membership degree different from the traditional fuzzy set can be obtained without increasing the parameters. This method provides a richer, which is convenient for more applications in the future.

## Structure Learning

結構學習是為了透過訓練資料，建造出更適當的模型架構，此外結構學習中的結果，也會成為之後參數學習的一部分。在本研究採用高斯型態的模糊集合，需要中心以及標準差兩個參數，因此會透過減數分群(Subtractive cluster, SC)演算法 [7]分群不同輸入維度的訓練資料。並將分群後的群中心配合每個維度的標準差形成模糊集，各個維度的模糊集個數總和，即為第1層神經元的數量，基於各個輸入維度的模糊集，共可以組成個區塊。

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

其中，為第個輸入維度的分群個數，亦即模糊集個數。以第個區塊的組成為例，如下。

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

，其中，為第個輸入的語意變數;為第個輸入變數，;為第個區塊中第個輸入語意變數的模糊集合，使用高斯函數建構，如公式(5)。



Fig. 1. 模糊集合輸入空間(2維)

兩個輸入維度，各分出3群則會形成共9塊區域，其中z軸為該區的資料密度。



Fig. 2. 資料密度總和

可看出部分區塊資料密度總和較高，代表用於建造模型較有利。

為了降低模型的運算負擔，以提升效率，我們將會篩選出若干個較重要的區塊成為前鑑部神經元，也就是第2層中的神經元。若以兩個輸入維度為例子可得到如Fig. 1的區塊，而後透過資料密度的概念，將資料灑入區塊中，計算資料密度量，累加每個區域的資料密度量後可得到如Fig. 2，從中可挑選資料密度較高的區塊，作為第2層神經元。詳細步驟如下:

1. 從各個輸入維度與模糊集可以得到每個區塊的資料密度量，以第個區塊為例，資料密度量公式如下。

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

其中，為第個輸入維度的第筆資料;為第個區塊中第個輸入維度的模糊集。

1. 將第個區塊的資料密度量累加標記為，公式如下。

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

其中，為資料總筆數。計算平均值標記為，標準差標記為。

1. 查看每個區塊，若，則將累加。設定上下界，標記為和，透過上下界找出，表示最後選取的區塊數目。若介於上下界間，則將設定成;若小於下界，則將設定成;若大於上界則將設定成。
2. 將排序，並保留前個區塊，當作之後模型第2層的神經元。

位於第4層的後鑑部神經元個數，在本研究中與前鑑部神經元數目相同。後鑑部神經元為T–S神經元，由T–S function構成，T–S function公式如下。

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

其中，{}是第個T–S神經元的參數，是第個輸入。

在結構學習後，我們可以透過分群確認第1層的神經元個數，藉由區塊挑選得到個第2層神經元以及第4層神經元個數，藉此創建模型，模型詳細說明將在下個小節探討。

## Complex Neuro-fuzzy System

本研究利用Takagi-Sugeno (T-S) fuzzy 建立模糊系統。T-S模糊模型最早是由Takagi與Sugeno於1985年提出 [36]，以一個複合式非線性系統並藉由一系列的If-Then模糊規則組合而成。本研究是透過複數高斯模糊集非線性的系統，結合線性的T-S function，形成非線性的If-Then模糊規則網路架構。If-Then模糊規則類似人類的經驗法則，因此更容易被人類所理解。本章節將探討模型各類神經層的輸入、計算方法與輸出。

本研究的模型為一個複合式非線性系統。訓練資料集合標記為，為資料總筆數，是-by-1的輸入向量，為輸入維度數量;為-by-1的目標向量，為複數型態目標的數量。透過模型可以得到輸出。

模型是透過條T-S模糊規則組成，每條規則是由前鑑部(If part)與後鑑部(Then part)所結合，其規則形式如下：

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

規則數；為模糊系統的輸入變數；為第條規則的複數模糊集；則是輸入的語意變數。複數模糊集的參數即是前鑑部(If part)參數，為後鑑部(Then part)參數。此複數類神經糢糊模型可轉為一個六層架構的類神經網路模型，如Fig. 3，以下將對各層進行說明。

Layer0

(輸入層)

Layer1

(複數模糊集層)

Layer2

(前鑑部層)

Layer4

(後鑑部層)

Layer5

(輸出層)

Layer3

(正規化層)

Fig. 3. 複數類神經模糊系統模型

**Layer 0**: 此層為輸入層，是原始資料透過多目標特徵選取後，將最後挑出的特徵當作訓練資料，我們將時間序列第個點的輸入向量標記如下。

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

**Layer 1**: 此層為複數模糊集神經層，透過前面結構學習的分群，可在不同維度上建構數個模糊集，每個不同維度的輸入都可經由模糊集得到歸屬程度。透過複數模糊集合可得到多組複數型態的歸屬程度，不同的歸屬程度可以給不同的模型輸出做應用，以達到多目標預測的效果，透過複數模糊集的公式(2)-(9)，可得到歸屬度向量，如下。

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

**Layer 2**: 此層為前鑑部神經層，經過前面的結構學習，我們可以篩選出個對模型較有用的區塊並建構神經元，由於神經元的輸入為上一層的歸屬程度，且輸出為每個輸入維度的歸屬程度相乘結果，故稱之為神經元，每個神經元的輸出為該神經元的啟動強度，由於本研究採用複數模糊集合，因此每個神經元的輸入會是向量型態，輸出亦然如此。

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

，為的乘積，為第個神經元中第個維度歸屬度向量的第項歸屬程度，。

**Layer 3**: 此層為正規化層，會將個輸入向量中的各個元素正規化，之後將結果以向量型態輸出，輸出如下。

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

其中，為第個神經元中第個元素正規化後的值，。本研究中使用的是複數模糊集，因此輸入是複數型態，故輸出也是複數型態。

**Layer 4:** 此層為後鑑部層，經過此層的運算可以得到個模型輸出，公式如下。

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

，為第個T–S神經元輸出；{}是第個T–S神經元的參數。

**Layer 5:** 此層為輸出層，將上一層得到的個神經元輸出加總，即為我們的模型輸出。

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

## Parameter Learning

根據分治法(Divide-and-conquer)的概念，我們將使用不同的機器學習演算法，對各層的參數優化，以便更容易找到最佳解。對於第1層複數模糊集的參數優化，我們使用兩種不同的演算法，其中包含粒子群演算法(Particle swarm optimization, PSO) [19]以及人工蜂群演算法(Artificial bee colony optimization, ABCO) [17]。第2層與第3層沒有參數需要優化。在第4層的部分，採用遞迴最小平方演算法(Recursive least squares estimation, RLSE) [16]最佳化後鑑部參數，希望透過不同演算法的結合，降低所需搜尋的參數維度，為模型帶來更好的效能表現。以下將會詳細敘述三種演算法的運作原理以及公式等細節。

1. Particle Swarm Optimization

粒子群演算法是由J. Kennedy et al. [19]於1995年開發的一種演化計算技術，來源於對一個簡化社會模型的模擬。其原理類似鳥群在尋找食物，除了自身提供的資訊，慣性以及自身最佳位置，也運用到群體智慧中全群最佳位置，用以調節速度，如Fig. 4，此演算法特性為收斂快速，公式如下。

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

其中，為第回合時第個粒子的位置，為第回合時第個粒子的速度，為第回合時第個粒子的最好位置，為第回合時全部粒子中最好的位置，為PSO的參數，、為介於0到1的隨機數。在本實驗中，粒子的位置代表前鑑部的參數，其中包含了每個維度的分群中心、標準差以及相位頻率參數。

目前位置

新位置

透過自我最佳位置

調整方向

透過全群最佳位置調整方向

全群最佳位置

(Gbest)

自我最佳位置

(Pbest)

慣性方向

第顆粒子

Fig. 4. PSO粒子更新位置示意Fig.

1. Artificial Bee Colony Optimization

人工蜂群演算法為Karaboga學者所提出 [17]，原理類似蜜蜂尋找食物來源的概念，其特性包含群體智慧與隨機性，如蜜蜂傳達食物位置時會彼此透過搖擺舞溝通，而搖擺舞所指示食物位置會有偏差帶有隨機性。此演算法中，總共有三種蜜蜂，包括工蜂(Employed bee)、觀察蜂(Onlooker bee)以及偵查蜂(Scout bee)。其中，工蜂負責尋找食物源的位置，然後以跳舞傳達食物源的收益訊息，每次在傳達時會帶有隨機性，代表著整體蜜蜂大維度的搜尋；觀察蜂負責在其中一個食物源附近搜尋，首先會觀察各個食物收益度，並以輪盤法選擇一個食物源，到食物源的鄰近周遭進行搜尋，代表著食物源附近小維度的搜尋；偵查蜂作用為當食物源經過一定的開發次數後，若食物源的收益度沒有改善時，則會派出偵查蜂取代食物源，意味著隨機的探索一個全新的食物源，可以避免演算法陷入區域最佳解中。其步驟與真實的蜜蜂找尋食物不盡相同，演算法步驟如下:

1. 隨機尋找其中一隻工蜂，並透過公式形成新位置，此位置為該次迭代的食物源位置，公式如下。

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

其中，為第個食物源的第個維度;為第隻工蜂的第個維度;為其他隨機工蜂的第個維度的值。

1. 觀察蜂使用輪盤法(Roulette method)挑選一食物源，其中食物源收益越好越容易被選中，輪盤機率公式如下。

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

其中，為第個食物源被選中的機率;為收益程度，本研究將成本函數的倒數視為收益程度;為食物源的總數目。

1. 派出每隻觀察蜂在剛剛被選中的食物源位置附近搜尋，公式如下。

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

其中，為第隻觀察蜂的第個維度;為被選中的食物源第個維度;為被其他隨機的食物源第個維度的值，若觀察蜂位置比被選中食物源位置好，則取代食物源位置。

1. 判斷每隻工蜂是否已經達到限制回合都未更新，若為真則派出偵查蜂取代，偵查蜂位置產生公式如下。

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

其中，為第個食物源的第個維度;為所有食物源中第個維度的最大值;為所有食物源中第個維度的最小值。

重複steps 2~4，直到反覆運算結束。

1. Recursive Least Squares Estimation

本研究使用遞迴式最小平方演算法(Recursive least squares estimation, RLSE) [16]更新T–S神經元參數，RLSE方法在更新參數時是利用每筆資料，不斷的更新比起一次接收所有資料的LSE方法更加有效，一般來說LSE問題可以被視為一個線性的問題，如下。

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

其中，y是目標;u是模型的輸出;{}是u已知的方程式;{, =1,2,…,m}是我們估計的未知參數，則是整個模型的誤差LSE的問題也可以被寫成矩陣的方式表達，如下。

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

其中，

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

是輸入的矩陣，是我們估計的未知參數矩陣，是目標矩陣，是誤差的向量。要最佳化，可透過RLSE的等式運算。

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

其中，是遞迴次數，{}，為資料總筆數，是的第行，再開始RLSE演算法時，會設定為0，則設定為，為一極大整數，為單位矩陣。

混合型演算法中各部分參數優化會交由不同演算法負責，本研究使用到PSO-RLSE以及ABCO-RLSE兩種混合型演算法，PSO和ABCO負責前鑑部的參數學習，RLSE則負責線性的T-S function 參數學習。混合演算法與模型計算的流程如下:

1. 準備訓練資料及測試資料。
2. 以前鑑部演算法(PSO或ABCO)粒子位置作為模糊集參數，將訓練資料帶入模型，並計算每個神經元啟動強度。
3. 用RLSE更新T-S神經元的參數，RLSE算式中的和向量如下。

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

其中。因為多目標預測中，正規化後的啟動強度為一向量，使為一矩陣，因此在原本的公式(36)中，利用單位矩陣取代原本的常數項1，改良後公式如下。

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

1. 更新完所有參數後，計算出模型的輸出。
2. 計算成本，更新前鑑部演算法粒子的位置及相關數據。

重複Steps 2~5，直到迭代結束。

## Investment Stratgy

為了評估模型是否對投資有實質上的幫助，使用成本函數是不夠的，因為從中無法看出是否有利潤，只能瞭解模型的配適率，而配適率高並不代表投資效益高，因此本實驗將預測出來的收盤價配合投資策略 [38]，進一步決定要買進或是賣出，買進與賣出公式如下。

|  |  |
| --- | --- |
| 買進: if , |  |
| 賣出: if , |  |

其中，為門檻參數，同時也代表股票的漲跌；為模型的輸出，意即預測日的收盤價格;為日實際的收盤價格。若預測明天的收盤價高於今天實際收盤價，代表必須買進;若預測明天的收盤價低於今天實際收盤價，代表模型預測明天會跌，所以要儘快賣出。

為了使得買賣更加謹慎，本研究提出另一買賣策略，策略步驟如下。

1. 使用公式(43)-(44)判定是否買或賣，接著進入第二階段
2. 計算第天的過去30天漲跌平均值，做為第二階段判斷標準。經過多次測試以過去30天平均最佳。
3. 若，且>則買進；若，且，則賣出。
4. 所有交易日需通過兩階段評判，若其中一階段未通過則不操作。

計算利潤的方式，則透過今天實際的收盤價與隔天實際的收盤價做運算，公式如下。

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

其中，為利潤，為策略為買的總天數;為策略為賣的總天數;代表第天的真實收盤價。

透過上述投資策略以及利潤公式，我們可以計算出整個模型所帶來的利潤值，並大致模擬出此模型運用到真實世界的效果，本研究將會在每個實驗中秀出上述兩種策略的利潤值與其他參數與現有文獻做比較。為了使利潤估計更具真實性，本研究提出滑動窗格計算利潤法，當中包含了持有股票概念，步驟如下:

1. 初始化窗格大小，每一窗格代表股票應結清日期，若窗格大小為10，意即每10天會結算一次持有股票。
2. 透過投資策略決定買進或賣出。
3. 若買進，則將持有股票加1，並用本金扣除當天價格；若判斷賣出，則會檢查是否有股票，若持有股票則以當天價格賣出，並將利潤加入本金。

重複步驟Steps 2~3，每當達到設定窗格大小，則將所有持有股票售出，若資料小於窗格大小，則以資料最後一天結清。

# Experimentation

There are three experiments in this study. The first experiment is a single target prediction. The target is the Taiwan stock exchange capitalization weighted stock index (TAIEX) in 2001 year; experiment 2 has two targets. The model is tested for the feasibility of the complex-valued membership degree. The first complex-valued output is used to predict the two targets, the real part is for the first target TAIEX, the imaginary part is for the second target Hang Seng index (HSI); experiment 3 is a multi-target prediction for four targets. The two complex-valued outputs are used, which are more complicated than Experiment 2. It includes the 2001 TAIEX and the Dow Jones industrial average index (DJI), national association of securities dealers automated quotation (NASDAQ) and standard & Poor's 500 (S&P 500). The above targets are very well-known stock indicators, for example, TAIEX is a weighted calculated indicator of listed stocks in Taiwan, which represents fluctuations of listed stocks in Taiwan; HSI is an important indicator reflecting the Hong Kong stock market. The index is calculated from the market capitalization of 50 HSI constituent stocks, which is equivalent to 63% of the 12-month average market capitalization rate of all listed companies on the Hong Kong Stock Exchange; DJI covers nine major industries such as finance, which is a stock price weighted indicator; NASDAQ is a market weighted indicator of more than 3,000 stocks, 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.

In order to compare the performance of the model with other literatures, we will evaluate the model through the error indicators and calculate the profit after the virtual investment. Both the cost function and the evaluation index use the root mean square error (RMSE), and the formula is as follows.

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

, is the number of data; is the error vector of the data in the model; is the target; is the output vector of the data; is Hermitian transpose, means that after the matrix is transposed, the matrix elements are conjugated.

In the part of structure learning, the number of rules for all experiments in this study which also are the number of the neurons in the second layer, the upper limit is set to 15, and the lower limit is set to 4, so the number of neurons in the second layer of the model will be between 4 and 15.

In terms of simulation investment, the threshold of all the investment strategies of the experiment will be between 0 and 0.1, because the threshold parameter represents the fluctuation of the stock, while the rise and fall limit of the Taiwan stock is 10%, so the best will be selected from this interval. This study set 0.001 as the iterated step, and the search the threshold from 0 to 0.1. All experiments will find the best profit threshold through the training data, and calculate the profit of the testing data. If the profit in the training phase is 0, then testing data profit is set to 0, that is, it does not participate in the investment.

## Example 1—Time Series of Daily TAIEX

This experiment uses real-world time series data to verify the performance of the model, the target is TAIEX. The goal of the experiment is to establish an appropriate model that is trained to predict the daily stock price index. The model output is complex-valued, and the real part is used to predict. The data used is the daily closing price of TAIEX in 2001. The original data of this experiment was 278. After making first difference, 30 features were extracted from the data. Each feature has 247 data and the first 181 data are used as training data, the rest is used as testing data. After multi-target feature selection, the features are selected as model input data. The part of structure learning is clustered by the SC algorithm which is mentioned in the selection III, as described in the section III-B, there are {3, 3, 3, 3} complex fuzzy sets for each input dimension of the experiment. After the block selection, the original 144 π neurons were screened to the 15 π neurons. The model is properly constructed by the data drive concept. Each complex fuzzy set has three parameters, including the center, standard deviation, and phase frequency parameters. There are 12 complex fuzzy sets in total, so the number of parameters in the if-part is 36. The then-part type is T-S function, so there will be a total of parameters, is the number of then-parts, and is the number of model input dimensions. The parameters of the overall model after structure learning, as shown by TABLE I. Machine learning parameter settings, as shown in TABLE II. The results of this example are compared with the methods proposed in other papers [38], such as Chen, Yu, SVR, and ANFIS, comparison is shown as TABLE III. In order to test the stability of the model, we ran ten trials of experiments, and the performance statistics are shown in TABLE IV. The virtual investment profit is shown as TABLE V, it can be found that the lower standard deviation of the investment strategy proposed in this study, which means the lower the investment risk, and the average of the profit still is positive. The profit comparison of the simulated investment is shown in TABLE VI, and the profit from the model in this study is highest. The profit of the sliding window calculation method is shown in TABLE VII. The results of the target and model output are shown in Fig. 5; the machine learning curve of the model is shown in Fig. 6; the prediction errors are shown in Fig. 7.

TABLE I

Model setting (Experiment 1)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 3, 3, 3} |
| Type of fuzzy sets | CFS |
| Number of complex-valued targets\* | 1 |
| Number of neurons | 15 |
| Number of parameters in the CFS layer | 36 |
| Number of T–S neurons | 15 |
| Number of parameters in the T–S layer | 75 |

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

TABLE II

Machine Learning setting

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 50 |
| Iterations | 100 |
|  | {0.8, 2.0, 2.0} |
|  | Random in [0,1] |
| Initial particle positions | By SC algorithm in section III-B |
| Initial particle velocities | 0 |
| **ABCO** |  |
| Number of employee bees | 40 |
| Number of onlooker bees | 10 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 25-by-1 zero vector |
|  | **I** |
| **I** | 25-by-25identify matrix |

TABLE IV

Ten Trials Performance (Experiment 1)

|  |  |  |
| --- | --- | --- |
| Trials | RMSE | |
| PSO-RLSE | ABCO-RLSE |
| 1 | 102.33 | 102.94 |
| 2 | 102.17 | 105.22 |
| 3 | 104.88 | 102.69 |
| 4 | 102.69 | 102.75 |
| 5 | 102.96 | **101.93** |
| 6 | **102.01** | 102.97 |
| 7 | 103.17 | 103.15 |
| 8 | 117.92 | 103.01 |
| 9 | 102.54 | 104.63 |
| 10 | 103.96 | 106.87 |

TABLE III

Performance Comparison (TAIEX, Experiment 1)

|  |  |
| --- | --- |
| Methods | RMSE |
| Chen [38] | 167 |
| Yu [38] | 148 |
| AR(1) [38] | 115 |
| SVR [38] | 114 |
| ANFIS [38] | 120 |
| Wei [38] | 110 |
| PSO-RLSE (proposed) | 102.01 |
| ABCO-RLSE (proposed) | **101.93** |

TABLE V

Virtual Investment Profits (Experiment 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| Profit | | Profit | | Profit | | Profit | |
| Best | 0.052 | | 0.027 | | 0.057 | | 0.048 | |
| 1 | **990.76** |  | -430.77 |  | 0 |  | 0 |  |
| 2 | 698.90 |  | -484.46 |  | 102.42 |  | -25.62 |  |
| 3 | -1.7452 |  | -700.05 |  | **205.99** |  | 0 |  |
| 4 | -826.35 |  | **790.05** |  | -86.05 |  | 0 |  |
| 5 | 235.74 |  | 666.28 |  | 0 |  | **187.81** |  |
| 6 | 283.94 |  | 490.75 |  | 102.01 |  | 108.23 |  |
| 7 | -356.82 |  | 207.13 |  | 161.61 |  | 12.12 |  |
| 8 | 465.34 |  | 530.23 |  | -21.53 |  | 82.03 |  |
| 9 | 152.48 |  | -382.58 |  | 12.12 |  | -21.53 |  |
| 10 | -713.04 |  | 145.88 |  | 188.41 |  | 80.18 |  |
| Average | -81.43 | | 83.24 | | 66.49 | | 42.32 | |
| Std | 826.24 | | 541.78 | | 99.29 | | 69.52 | |
| Maximum | 990.76 | | 790.05 | | 205.99 | | 187.81 | |
| Minimum | -826.35 | | -700.05 | | -86.05 | | -21.53 | |

\*The result of the proposed investment strategy.

TABLE VI

Virtual Profit Comparison (Experiment 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Chen [38] | Yu [38] | AR(1) [38] | SVR [38] | ANFIS [38] | Wei [38] | PSO-RLSE | ABCO-RLSE |
| Best | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.052 | 0.027 |
| Profit (TAIEX) | -92 | -73 | 671 | 202 | 686 | 795 | **990.76** | 790.05 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TABLE VII  Sliding Window Profit (Experiment 1) | | | | | | | | | |
| Trials | PSO-RLSE | | | |  | ABCO-RLSE | | | |
| Window size | 5 | 10 | 20 | 30 |  | 5 | 10 | 20 | 30 |
| Best | 0.019 | 0.056 | 0.001 | 0.056 |  | 0.048 | 0.009 | 0.048 | 0.048 |
| 1 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 |  | **177.02** | **269.14** | **704.83** | **269.14** |
| 6 | 102.01 | **161.69** | **161.69** | 9.18 |  | 82.03 | 0 | 609.84 | 174.15 |
| 7 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |  | 82.03 | 0 | 609.84 | 174.15 |
| 9 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | **319.26** |  | 0 | 0 | 0 | 0 |
| Average | 10.2 | 90.96 | 16.17 | 32.84 |  | 34.11 | 26.91 | 192.45 | 61.74 |
| Std | 32.26 | 148.68 | 51.13 | 100.67 |  | 60.7 | 85.11 | 310.95 | 102.72 |
| Maximum | 102.01 | 161.69 | 161.7 | 319.26 |  | 177.02 | 269.14 | 704.83 | 269.14 |
| Minimum | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |



(a) TAIEX Forecasting (PSO-RLSE)



(b) TAIEX Forecasting (ABCO-RLSE)

Fig. 5. Forecasting result. The blue line is the target, the red dash line is the model output. The result has great performance in the training and testing phase.



(a) PSO-RLSE



(b) ABCO-RLSE

Fig. 6. Model forecasting errors. The errors in both two algorithms are in a random number range which is from -50 to 50, indicating that the prediction ability of the model is stable.

(a) PSO-RLSE



(b) ABCO-RLSE

Fig. 7. Learning curve. (Experiment 1)

## Example 2—Double Time Series of Daily TAIEX and HSI

This experiment predicts two targets at the same time, namely TAIEX and HSI. The goal of the experiment is to establish an appropriate model that is trained to predict the daily stock price index. The model output is complex-valued, the real part is used to predict TAIEX and imaginary part is used to predict HSI. The data used is the daily closing price of TAIEX and HSI in 2000. The original data of this experiment was 277. After making first difference, we can get 276 data and the 30 features are extracted from each set of data for a total of 60 features. Each feature has 247 data and the first 204 data are used as training data, the rest is used as testing data. The first to the 30th features are TAIEX data, and the 31th to the 60th features are HSI. After multi-target feature selection, the features are selected as model input data. The part of structure learning is clustered by the SC algorithm which is mentioned in the selection III, as described in the section III-B, there are {3, 3, 3, 3} complex fuzzy sets for each input dimension of the experiment. After the block selection, the original 81 π neurons were screened to the 9 π neurons. The model is properly constructed by the data drive concept. Each complex fuzzy set has three parameters, including the center, standard deviation, and phase frequency parameters. There are 12 complex fuzzy sets in total, so the number of parameters in the if-part is 36. The then-part type is T-S function, so there will be a total of parameters, is the number of then-parts, and is the number of model input dimensions. The parameters of the overall model after structure learning, as shown by TABLE VIII. Machine learning parameter settings, as shown in TABLE IX. The results of this example are compared with the methods proposed in other papers [6], such as Chen, Yu, SR+ANFIS etc., comparison is shown as TABLE X. In order to test the stability of the model, we ran ten trials of experiments, and the performance statistics are shown in TABLE XI. The virtual investment profit is shown as TABLE XII, it can be found that the lower standard deviation of the investment strategy proposed in this study, which means the lower the investment risk, and the average of the profit still is positive. The profit comparison of the simulated investment is shown in TABLE XIII, and the profit from the model in this study is highest. The profit of the sliding window calculation method is shown in TABLE XIV. The results of the target and model output are shown in Fig. 8; the machine learning curve of the model is shown in Fig. 9; the prediction errors are shown in Fig. 10.

TABLE VIII

Model Setting (Experiment 2)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 3, 3, 3} |
| Type of fuzzy sets | CFS |
| Number of complex-valued targets\* | 1 |
| Number of neurons | 9 |
| Number of parameters in the CFS layer | 36 |
| Number of T–S neurons | 9 |
| Number of parameters in the T–S layer | 45 |

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

TABLE IX

Machine Learning

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 50 |
| Iterations | 100 |
|  | {0.8, 2.0, 2.0} |
|  | Random in [0,1] |
| Initial particle positions | By SC algorithm in section III-B |
| Initial particle velocities | 0 |
| **ABCO** | |
| Number of employee bees | 40 |
| Number of onlooker bees | 10 |
| Iterations | 100 |
| Limits | 20 |
| **RLSE** | |
|  |  |
|  | 25-by-1 zero vector |
|  | **I** |
| **I** | 25-by-25 identity matrix |

TABLE X

Performance Comparison (Experiment 2)

|  |  |  |
| --- | --- | --- |
| **Method** | **TAIEX** | **HSI** |
| Chen [6] | 413.27 | 280.15 |
| Yu [6] | 419.64 | 297.05 |
| SR+ANFIS [6] | 454.63 | 356.70 |
| SR+SVR [6] | 255.87 | 356.81 |
| Elman [6] | 154.21 | 302.27 |
| Cheng [6] | **150.55** | 251.70 |
| PSO-RLSE | 151.06 | 254.97 |
| ABCO-RLSE | 153.45 | **250.51** |

TABLE XI

Ten Trials Performance (Experiment 2)

|  |  |  |
| --- | --- | --- |
| Trials | PSO-RLSE | ABCO-RLSE |
| 1 | 298.49 | 296.49 |
| 2 | **296.36** | 304.76 |
| 3 | 302.48 | 296.26 |
| 4 | 298.20 | 296.25 |
| 5 | 305.08 | 296.59 |
| 6 | 332.39 | **293.77** |
| 7 | 308.42 | 314.39 |
| 8 | 303.56 | 295.34 |
| 9 | 297.36 | 297.58 |
| 10 | 329.41 | 307.30 |

TABLE XII

Virtual Investment Profit (Experiment 2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| Profit | | Profit | | Profit | | Profit | |
| Best | 0.029 | | 0.073 | | 0.047 | | 0.011 | |
| 1 | -127.26 |  | -1540.7 |  | 167.42 |  | 128.39 |  |
| 2 | -1712.9 |  | **2062.4** |  | 122.46 |  | 35.49 |  |
| 3 | -2726.6 |  | -1337 |  | 166.24 |  | -61.21 |  |
| 4 | -2022.7 |  | -822.64 |  | 62.53 |  | -143.9 |  |
| 5 | 909.16 |  | 814.5 |  | 515.98 |  | **285.72** |  |
| 6 | -969.99 |  | -1902.1 |  | 500.97 |  | -209.4 |  |
| 7 | 1453.9 |  | 163.68 |  | 25.05 |  | 149.63 |  |
| 8 | -2712.9 |  | 665.28 |  | 66.42 |  | -95.68 |  |
| 9 | **2515.3** |  | -1274 |  | **800.26** |  | 128.39 |  |
| 10 | -826.09 |  | -941.3 |  | -121.6 |  | 202.16 |  |
| Average | -622 | | -411.17 | | 230.57 | | 41.95 | |
| Std | 1790 | | 1276.3 | | 282.92 | | 162.95 | |
| Maximum | 2515.3 | | 2062.4 | | 800.26 | | 285.72 | |
| Minimum | -2726.6 | | -1902.1 | | -121.6 | | -209.4 | |

\*The result of the proposed investment strategy.

TABLE XIII

Virtual Investment Profit Comparison (Experiment 2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Chen [6] | Yu  [6] | SR+ANFIS [6] | SR+SVR [6] | Elman [6] | Cheng et al. [6] | PSO-RLSE | ABCO-RLSE |
| Best | 0.012 | 0.012 | 0.012 | 0.012 | 0.012 | 0.012 | 0.029 | 0.073 |
| Profit (TAIEX) | 0 | 0 | 0 | 0 | 0 | -231.02 | **1367.8** | 922.95 |
| Profit (HSI) | -1471 | -1368 | -602.94 | 190.71 | 2342 | 1793.12 | **2393.1** | 1139.4 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TABLE XIV  Sliding Window Profit (Experiment 2) | | | | | | | | | |
| Trials | PSO-RLSE | | | |  | ABCO-RLSE | | | |
| Window size | 5 | 10 | 20 | 30 |  | 5 | 10 | 20 | 30 |
| Best | 0.023 | 0.024 | 0.098 | 0.015 |  | 0.001 | 0.001 | 0.002 | 0.002 |
| 1 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| Average | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| Std | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| Maximum | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |
| Minimum | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 |



(a) TAIEX Forecasting (PSO-RLSE)

(b) TAIEX Forecasting (ABCO-RLSE)



(c) HSI Forecasting (ABCO-RLSE)



(d) HIS Forecasting (ABCO-RLSE)

Fig. 8. The model output. The blue line is the target, red dash line is the model output, it can be found the two algorithm have great performance.



(a) PSO-RLSE



(b) ABCO-RLSE

Fig. 9. Model prediction errors. The errors in the both algorithm are in the random range, indicating the model prediction ability is stable.



(a) PSO-RLSE



(b) ABCO-RLSE

Fig. 10. Learning Curve (Experiment 2)

PSO-RLSE is stable in the 75th iteration, and the ABCO-RLSE is stable in the 25th iteration.

## Example 3—Quadruple Time Series of Daily TAIEX, DJI, NASDAQ and S&P500

This experiment predicts four targets at the same time, namely TAIEX, DJI, NASDAQ, S&P500. The goal of the experiment is to establish an appropriate model that is trained to predict the daily stock price index. The model has two complex-valued outputs, the real part of the first output is used to predict TAIEX, the imaginary part of the first output is used to predict DJI, the real part of the second output is used to predict NASDAQ and the imaginary part of the second output is used to predict S&P500. The data used is the daily closing price of TAIEX, DJI, NASDAQ and S&P500 in 2001. The original data of this experiment was 278. After making first difference, we can get 277 data and the 30 features are extracted from each set of data for a total of 120 features. Each feature has 247 data and the first 181 data are used as training data, the rest is used as testing data. The first to the 30th features are TAIEX data, the 31th to the 60th features are HIS, the 61th to the 90th features are NADSDAQ data, and the 91th to the 120th features are S&P500. After multi-target feature selection, the features are selected as model input data. The part of structure learning is clustered by the SC algorithm which is mentioned in the selection III, as described in the section III-B, there are {3, 3, 3, 3} complex fuzzy sets for each input dimension of the experiment. After the block selection, the original 81 π neurons were screened to the 9 π neurons. The model is properly constructed by the data drive concept. Each complex fuzzy set has three parameters, including the center, standard deviation, and phase frequency parameters. There are 12 complex fuzzy sets in total, so the number of parameters in the if-part is 36. The then-part type is T-S function, so there will be a total of parameters, is the number of then-parts, and is the number of model input dimensions. The parameters of the overall model after structure learning, as shown by TABLE XV. Machine learning parameter settings, as shown in TABLE XVI. The results of this example are compared with the methods proposed in other papers [27], such as SVR, ANFIS, RBF and CNFS-ARIMA etc., comparison is shown as TABLE XVII. In order to test the stability of the model, we ran ten trials of experiments, and the performance statistics are shown in TABLE XVIII. The virtual investment profit is shown as TABLE XIX, it can be found that the lower standard deviation of the investment strategy proposed in this study, which means the lower the investment risk, and the average of the profit still is positive. The profit comparison of the simulated investment is shown in TABLE XX, and the profit from the model in this study is highest. The profit of the sliding window calculation method is shown in TABLE XXI. The results of the target and model output are shown in Fig. 11; the machine learning curve of the model is shown in Fig. 12; the prediction errors are shown in Fig. 13.

TABLE XV

Model Setting (Experiment 3)

|  |  |
| --- | --- |
| **Parameters** | **Value** |
| Feature variables as model inputs | {} |
| Number of input fuzzy sets | {3, 3, 3, 3} |
| Type of fuzzy sets | CFS |
| Number of complex–valued targets\* | 2 |
| Number of neurons | 9 |
| Number of parameters in the SCFS layer | 36 |
| Number of T–S neurons | 9 |
| Number of parameters in the T–S layer | 45 |

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

TABLE XVI

Machine Learning

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 50 |
| Iterations | 100 |
|  | {0.8, 2.0, 2.0} |
|  | Random in [0,1] |
| Initial particle positions | By SC algorithm in the section III-B |
| Initial particle velocities | 0 |
| **ABCO** |  |
| Number of employee bees | 40 |
| Number of onlooker bees | 10 |
| Iterations | 100 |
| Limits | 20 |
| **RLSE** | |
|  |  |
|  | 25-by-1 zero vector |
|  | **I** |
| **I** | 25-by-25 identity matrix |

TABLE XVII

Performance comparison (Experiment 3)

|  |  |  |
| --- | --- | --- |
| method | TAIEX | DJI |
| SVR (two models, each with single output) [27] | 162.46 | 101.44 |
| ANFIS (two models, each with single output) [27] | 147.36 | 105.56 |
| ANFIS (one model with two outputs) [27] | 151.62 | 128.20 |
| RBF (two models, each with single output) [27] | 134.32 | 106.33 |
| RBF (one model with two outputs) [27] | 137.58 | 181.79 |
| CNFS(5)-ARIMA (one model with two outputs) [27] | 115.82 | 103.06 |
| PSO-RLSE | **101.61** | 101.82 |
| ABCO-RLSE | 102.61 | **100.79** |

TABLE XVIII

Ten Trial Performance Comparison (Experiment 3)

|  |  |  |
| --- | --- | --- |
| Trials | PSO-RLSE | ABCO-RLSE |
| 1 | 185.99 | 199.32 |
| 2 | 183.75 | 183.89 |
| 3 | 183.72 | 204.56 |
| 4 | 183.81 | 256.10 |
| 5 | 184.39 | **180.82** |
| 6 | 191.31 | 199.00 |
| 7 | **181.90** | 217.09 |
| 8 | 281.103 | 182.13 |
| 9 | 183.68 | 188.24 |
| 10 | 183.67 | 183.89 |

TABLE XIX

Virtual Investment Profit (Experiment 3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| Profit | | Profit | | Profit | | Profit | |
| Best | 0.049 | | 0.049 | | 0.059 | | 0.025 | |
| 1 | -444.90 |  | 942.69 |  | 341.79 |  | **951.12** |  |
| 2 | 615.12 |  | 1206 |  | 0 |  | 266.13 |  |
| 3 | 880.74 |  | -850.1 |  | 52.399 |  | -139.9 |  |
| 4 | -1629.7 |  | 67.36 |  | 0 |  | 352.73 |  |
| 5 | -365.89 |  | **3201.7** |  | 149.49 |  | 682.72 |  |
| 6 | 762.83 |  | 1374.9 |  | **740.06** |  | 141.16 |  |
| 7 | **2146.7** |  | 2156.9 |  | 80.34 |  | 149.49 |  |
| 8 | -805.41 |  | 784.34 |  | 310.11 |  | 410.97 |  |
| 9 | -1741.7 |  | 3460.7 |  | 0 |  | 324.16 |  |
| 10 | -1189.6 |  | 1206 |  | 0 |  | 266.13 |  |
| Average | -177.19 | | 1355.1 | | 167.42 | | 340.47 | |
| Std | 1252.4 | | 1314 | | 238.49 | | 300.64 | |
| Maximum | 2146.7 | | 3201.7 | | 740.06 | | 951.12 | |
| Minimum | -1741.7 | | -850.1 | | 0 | | -139.9 | |

\*The result of the proposed investment strategy.

TABLE XX

Virtual Investment Profit Comparison (Experiment 3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Chen [38] | Yu [38] | AR(1) [38] | SVR [38] | ANFIS [38] | Wei [38] | PSO-RLSE | ABCO-RLSE |
| Best | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.049 | 0.049 |
| Profit (TAIEX) | -92 | -73 | 671 | 202 | 686 | 795 | 1104.6 | **1409.9** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TABLE XXI  Sliding Window Profit (Experiment 3) | | | | | | | | | |
| Trials | PSO-RLSE | | | |  | ABCO-RLSE | | | |
| Window size | 5 | 10 | 20 | 30 |  | 5 | 10 | 20 | 30 |
| Best | 0.022 | 0.022 | 0.017 | 0.064 |  | 0.006 | 0.006 | 0.05 | 0.022 |
| 1 | **457.97** | **1225.6** | 325.31 | 140.61 |  | 0 | 0 | 1837.7 | 602.43 |
| 2 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1724.5 | 658.46 |
| 3 | 354.04 | 538.28 | **1409.7** | **1159.8** |  | 177.02 | 269.14 | 1760.4 | 453.38 |
| 4 | 0 | 0 | 0 | 0 |  | 48.85 | 48.85 | 1704.2 | 397.15 |
| 5 | 0 | 0 | 0 | 107.53 |  | 0 | 0 | **2292.7** | 618.23 |
| 6 | 280.44 | 955.11 | 325.31 | 364.55 |  | 412.09 | 471.78 | 471.78 | 619.26 |
| 7 | 150.82 | 242.94 | 678.63 | 496.28 |  | 0 | 0 | 119.82 | 0 |
| 8 | 0 | 0 | 0 | 0 |  | 410.98 | **1760.3** | 650.62 | 0 |
| 9 | 0 | 0 | 0 | 0 |  | **766.13** | 1010.1 | 1881.4 | **857.54** |
| 10 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1724.5 | 658.46 |
| Average | 124.33 | **296.19** | 273.89 | 226.88 |  | 181.51 | 356.01 | **1416.8** | 456.49 |
| Std | 176.98 | 457.74 | 460.58 | 371.34 |  | 264.57 | 591.50 | 723.62 | 284.61 |
| Maximum | 457.97 | 1225.6 | 1409.7 | 1159.8 |  | 766.13 | 1760.3 | 2292.7 | 857.54 |
| Minimum | 0 | 0 | 0 | 0 |  | 0 | 0 | 119.82 | 0 |



(a) TAIEX Forecasting (PSO-RLSE)



(b) TAIEX Forecasting (ABCO-RLSE)



(c) DJI Forecasting (PSO-RLSE)



(d) DJI Forecasting (ABCO-RLSE)



(e) NASDAQ Forecasting (PSO-RLSE)



(f) NASDAQ Forecasting (ABCO-RLSE)



(g) S&P500 Forecasting (PSO-RLSE)



(h) S&P500 Forecasting (ABCO-RLSE)

Fig. 11. Model forecasting. The blue line is the target, red dash line is the model output, it can be found the two algorithm have great performance.



(a) PSO-RLSE



(b) ABCO-RLSE

Fig. 12. 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 3)



(a) PSO-RLSE



(b) ABCO-RLSE

Fig. 13. Learning curve. The RMSE variation can be seen from this curve. PSO-RLSE is stable in the 60th iteration, and the ABCO-RLSE is stable in the 75th iteration.

# Discussion

This study proposes to optimize the model parameters with the hybrid algorithm PSO-RLSE and ABCO-RLSE, and use the divide and conquer principle to reduce the search loading of the algorithm. When the model constructing, the model size is controlled by machine learning through the concept of data driven. Before the training data enters the model, the feature selection is performed in advance, and the feature data that is more favorable for the prediction target is selected for time series prediction. According to the 3 experiments, it is verified that the series of methods proposed in this study have better prediction performance for the prediction of time series.

Data preprocessing of this study uses multi-target feature selection [28], taken from Shannon information entropy [35]. The information entropy and the information entropy difference between the data are used to calculate the amount of information that the data candidate feature can provide for the target data, and to consider the amount of redundant information caused by the selected feature. Finally, from a large amount of data, the feature data that is useful for single target or multiple targets is selected to reduce the burden and computational complexity caused by too much data. Structure learning will determine the size of the model. Firstly, the initial position of the complex fuzzy set is preliminarily determined by the SC algorithm. Finally, the concept of data density is introduced, a smaller number of blocks are selected from a large number of blocks as the rules for reducing the burden of the model. From the 3 experiments, it can be found that the original number of blocks is reduced from 144, 81, 81 to 15, 9, and 9, which obviously reduces the burden on the model calculation. The establishment of the number of rules will also affect the number of the consequence layer. Therefore, the effective reduction of the number of rules will also affect the number of parameters that need to be learned during parameter learning, making it easier to find the best solution in the parameter search process.

The CNFS is established by CFSs and T-S fuzzy systems. Through the nonlinear Gaussian CFSs and the linear T-S fuzzy system combined into a nonlinear fuzzy system, the one-to-one relationship of the neurons is generated by the If-Then rule, so that the system can be understood by humans. Compared with the traditional fuzzy set, the CFS extends from the one-dimensional real space to the planar space of a two-dimensional unit disk. This method enables the membership degree to accommodate more information, which helps to improve the reasoning ability and application performance of the fuzzy system. Using the complex-valued membership degree makes the model system can generate a set of complex-valued outputs, so that it can simultaneously predict two real targets. Through the deconstruction of the membership degree, 3 complex-valued membership degree can be obtained, so that the original 3 outputs can predict 6 targets, means the prediction ability of the model is enhanced.

In the parameter learning of the CNFS, the if-part uses PSO and ABCO to learn, both of which have the concept of using swarm intelligence. Through the information exchange in the group find a better location. PSO has 3 characteristics. First, the automatic pace adjustment will adjust the position through weight and swarm intelligence. Second, randomness, the component with random variables when updating velocity, contributes to particle activity; third, stability, each time when updating the position and velocity, its moving direction is according to the best solution of the group and its best solution; ABCO also has 3 characteristics. First, randomness, onlooker bee uses roulette method when selecting the food source, so the selected food source has a randomness; Second, local search, Through the onlooker bees searching near the food source, it can avoid missing the best solution. Third, the jump mechanism can judge the food source effect by whether there is any update. If the food source is not updated within certain iterations, then a scout bee will replace the food source location, reducing the probability of being trapped in a local minimum. The then-part uses the RLSE algorithm to find a linear function by using the input data points and the previous calculation results, so that the square error of the data points and the function reaches a minimum value, thereby finding a function that conforms to the data. To optimize the parameters of the consequence part by continuous recursive calculations.

Through a series of methods, it can be seen that the 3 experimental results of this study are better than other methods, indicating that the model has good prediction ability in single target, double target and multi-target. In the 3 experiments, it can be found that the ABCO-RLSE hybrid algorithm performs better than the PSO-RLSE, means local search and tripping mechanisms has a certain effect, making the model output more accurate. In the virtual investment, we have an interesting finding that the performance of the model is not positively related to the virtual investment profit. When a model performs well, it does not mean that the virtual investment will be good. That is, in the 10 model trials, the trial of best performance is not the same as the simulation investment profit best trial, because the curve fitting of the model represents the matching rate, but it does not mean that it can effectively earn profits. Buying and selling are the critical factors. The operation of buying and selling is determined by the threshold parameter in the investment strategy. This parameter will affect the profit. Therefore, it can be found the best is different in PSO-RLSE and ABCO-RLSE in TABLE VI, TABLE XIII and TABLE XX, it is different from other literatures too. This study has a better performance compared with the past literature, but the predicting of the stock market's ups and downs are still difficult. In the new investment strategy proposed in this study, the past rise and fall values can be used to set another threshold in an objective manner, making investment more cautious. As expected, the standard deviation from TABLE V, TABLE XII and TABLE XIX can be found to be lower than the previous investment strategy, which means that the stability of investment profit is higher, and the average value is higher, indicating that the probability of making money is high. The sliding window profit calculation method is closer to the real world transaction. From 3 experiments, it can be found that different models and different data have different window size, but most of them have a best window size of not less than 10.

# Concluding Remarks

In this study, a new hybrid algorithm ABCO-RLSE is proposed, which combines the artificial bee colony algorithm and the recursive least squares method to optimize the parameter set of the CNFS. The model combines a complex fuzzy set, a T-S model system, and a neural network. Before the data enters the model, the feature selection is to avoid redundant input data entering the model, which decreases computational efficiency and increases the burden of the model. The complex fuzzy set makes the model have multiple complex-valued outputs. Through the deconstruction of the complex-valued membership degree, the real part and the imaginary part of the membership degree can take out other membership degrees, so that the model can generate three complex-valued outputs and predict up to six real targets, making the model different from the traditional fuzzy system and capable of simultaneous multi-target prediction. Experiment 2 and Experiment 3 demonstrate the contribution of this approach. In terms of structure learning, the model does reduce the size effectively, and through the concept of data density, select the blocks that are more useful for the data, objectively use the data to build the model, avoid complex calculations and improve the efficiency of the model. When parameters learning, ABCO can find the best solution more easily than PSO because of the jump mechanism and local search characteristics. The performance in three experiments is better than the methods proposed in other literatures. The investment strategy proposed in this study has a large chance of making money through the cooperation of threshold value. The standard deviation of the 10 simulated investment profits is lower than that of the investment strategy proposed in other literatures, which helps to reduce the risk of investors and provides the objective information through the data to investors.

Through the experiments in this study, we can find that the performance of the model is better than other literature methods, which proves the optimization ability of PSO and ABCO. However, these two algorithms still have their own limitations. For example, PSO may be premature, which means that the model converges in the first few iterations, which makes the model easy to fall into the local minimum; ABCO is limited to the local minimum when searching. The onlooker bee will search near the food source, when the food source update effect is not good, the overall effect will be bad, and the calculation time required is also more longer than the PSO algorithm. Therefore, other algorithms can be used in the future, combined with CNFS to try to find better performance. In terms of investment strategy, we find that the best threshold parameters are different for different models. Therefore, in the future, we can optimize the threshold parameters through machine learning algorithms, or adjust the weight of the investment according to the predicted rise and fall for more precise investment operations and making profits from the stock market.

# 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] E. Abbasi and A. Abouec, "Stock price forecast by using neuro-fuzzy inference system," *Proceedings of World Academy of Science, Engineering and Technology,* vol. 36, pp. 320-323, 2008.

[2] P. L. Avrim L. Blum, "Selection of relevant features and examples in machine learning," *Artificial Intelligence,* vol. 97, no. 1-2, 1997.

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

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

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

[6] C.-H. Cheng and J.-H. Yang, "Fuzzy time-series model based on rough set rule induction for forecasting stock price," *Neurocomputing,* vol. 302, pp. 33-45, 2018.

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

[8] R. Clausius, "Über eine veränderte Form des zweiten Hauptsatzes der mechanischen Wärmetheorie," *Annalen der Physik,* vol. 169, no. 12, pp. 481-506, 1854.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, no. 3, pp. 273-297, 1995.

[10] L. Davis, "Handbook of genetic algorithms," *Artificial Intelligence,* vol. 100, no. 1-2, pp. 325-330, 1991.

[11] E. F. Fama, "Random walks in stock market prices," *Financial analysts journal,* vol. 51, no. 1, pp. 75-80, 1995.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, pp. 1157-1182, 2003.

[13] D. O. Hebb, "The organization of behavior: A neuropsychological theory," ed: New York: Wiley, 1949.

[14] N. Hoque, D. Bhattacharyya, and J. K. Kalita, "MIFS-ND: a mutual information-based feature selection method," *Expert Systems with Applications,* vol. 41, no. 14, pp. 6371-6385, 2014.

[15] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications,* vol. 38, no. 7, pp. 8144-8150, 2011.

[16] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence," *Automatic Control,* vol. 42, no. 10, Oct 1997.

[17] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," 2005.

[18] M. G. Kendall and A. B. Hill, "The analysis of economic time-series-part i: Prices," *Journal of the Royal Statistical Society. Series A (General),* vol. 116, no. 1, pp. 11-34, 1953.

[19] J. Kennedy and R. Eberhart, "PSO optimization," *Proc. IEEE Int. Conf. Neural Networks,* vol. 4, pp. 1941-1948, 1995.

[20] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert systems with Applications,* vol. 19, no. 2, pp. 125-132, 2000.

[21] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," *Neural Networks,* pp. 1-6, 1990.

[22] R. S. Koijen, H. Lustig, and S. Van Nieuwerburgh, "The cross-section and time series of stock and bond returns," *Journal of Monetary Economics,* vol. 88, pp. 50-69, 2017.

[23] C. Li and T.-W. Chiang, "Complex neuro-fuzzy self-learning approach to function approximation," presented at the Asian Conference on Intelligent Information and Database Systems, 2010.

[24] C. Li and T.-W. Chiang, "Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach," presented at the Asian Conference on Intelligent Information and Database Systems, 2011.

[25] C. Li, C. W. Lin, and H. Huang, "Neural Fuzzy Forecasting of the China Yuan to US Dollar Exchange Rate—A Swarm Intelligence Approach," presented at the International Conference in Swarm Intelligence, 2011.

[26] 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.

[27] 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.

[28] C. Li, "國立中央大學資訊管理所李俊賢教授, 研究生訓練課程內容," 2016-2018.

[29] 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.

[30] 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, 2018.

[31] P. B. Patel and T. Marwala, "Neural networks, fuzzy inference systems and adaptive-neuro fuzzy inference systems for financial decision making," in *International Conference on Neural Information Processing*, 2006, pp. 430-439: Springer.

[32] 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.

[33] 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.

[34] N. Russel, *Artificial Intelligenz-A Modern Approach, 2nd*. 2002.

[35] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 5, no. 1, pp. 3-55, 2001.

[36] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *Readings in Fuzzy Sets for Intelligent Systems,* vol. SMC-15, no. 1, pp. 387-403, 1985.

[37] 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.

[38] 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.

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

[40] J. Yao, C. L. Tan, and H.-L. Poh, "Neural networks for technical analysis: a study on KLCI," *International journal of theoretical and applied finance,* vol. 2, no. 02, pp. 221-241, 1999.

[41] S. Yao, M. Pasquier, and C. Quek, "A foreign exchange portfolio management mechanism based on fuzzy neural networks," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 2576-2583.

[42] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of machine learning research,* vol. 5, no. Oct, pp. 1205-1224, 2004.

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

[44] X. Zhi-Bin and L. Rong-Jun, "Credit risk evaluation with fuzzy neural networks on listed corporations of China," presented at the VLSI Design and Video Technology, 2005.