Complex Fuzzy Computing to Time Series Prediction — A Multi-Swarm PSO Learning Approach

Chunshien Li and **Tai-Wei Chiang**

Laboratory of Intelligent Systems and Applications
Department of Information Management
National Central University, Taiwan, ROC.
jamesli@mgt.ncu.edu.tw

Abstract. A new complex fuzzy computing paradigm using complex fuzzy sets (CFSs) to the problem of time series forecasting is proposed in this study. Distinctive from traditional type-1 fuzzy set, the membership for elements belong to a CFS is characterized in the unit disc of the complex plane. Based on the property of complex-valued membership, CFSs can be used to design a neural fuzzy system so that the system can have excellent adaptive ability. The proposed system is called the complex neuro-fuzzy system (CNFS). To update the free parameters of the CNFS, we devise a novel hybrid HMSPSO-RLSE learning method. The HMSPSO is a multi-swarm-based optimization method, first devised by us, and it is used to adjust the premise parameters of the CNFS. The RLSE is used to update the consequent parameters. Two examples for time series foresting are used to test the proposed approach. Through the experimental results, excellent performance has been exposed.

Keywords: complex fuzzy set (CFS), complex neuro-fuzzy system (CNFS), hierarchical multi-swarm particle swarm optimization (HMSPSO), recursive least square estimator (RLSE), time series forecasting

1 Introduction

A time series is a sequence of historical statistical observations, for example, oil prices and stock prices in financial market. Time series analysis is to estimate the statistical regularities existed within the observed data and their connection to future tendency. The purpose of time series model is to explore a possible functional relationship with which the historical data are connected to future trend, so that a decision-making can be made in advance. Because accurate forecasting for the future trend is usually difficult in complex and nonlinear real-world problems, many researchers have used intelligent computing methods for time series forecasting, where fuzzy theory and neural networks have been widely investigated [1]-[4]. Although neural networks have excellent mapping ability and link-type distributed structure, they are usually considered as black-box systems, which are not easy to explain with human’s knowledge. In contrast, fuzzy inference systems, providing a complementary alternative to neural networks, can extract human’s experience and knowledge to form If-Then rules, which are easy to be explained by human. Both neural network and fuzzy system are with the property of universal approximator.

Consequently, they can be integrated as a neuro-fuzzy system (NFS) [1], which incorporates the advantages of fuzzy inference and neural-learning. The theory of NFS has become popular and important to modeling problems [1], [3], [5].

In this study, we propose a novel complex neuro-fuzzy system (CNFS) using the theory of complex fuzzy set (CFS) to achieve high prediction accuracy for the problem of time series forecasting. The concept of CFS is an extension from the standard type-1 fuzzy set whose membership is in the real-valued interval of [0, 1]. The membership for elements belong to a complex fuzzy set is expanded to the complex-valued unit disc of the complex plan [6]. This property can be used to augment the adaptability of the proposed CNFS, if compared to its counterpart in NFS form. Although the CFS theory has been proposed [6]-[9], it is hard to construct intuitively understandable complex fuzzy sets for application. The theoretical curiosity on CFS remains. For this reason, we propose the CNFS using the theory of CFS to study its adaptability gain for the problem of time series forecasting. For the training of the proposed CNFS, we devise a new learning method, which combines the novel hierarchical multi-swarm particle swarm optimization (HMSPSO) algorithm with the recursive least square estimator (RLSE) algorithm. The HMSPSO method is devised by us and presented in this paper. It is used to update the premise parameters of the proposed CNFS. In the meanwhile, the RLSE is used to adjust the consequent parameters. The HMSPSO is different from another multiple-swarm version of PSO in [10]. The HMSPSO is devised in hierarchical form to enhance searching multiplicity and to improve the drawback of the standard PSO. With the hybrid HMSPSO-RLSE learning method, it can effectively find the optimum solution for the parameters of the CNFS. Two time series examples are used to test the prediction performance by the proposed approach. The proposed approach shows not only better adaptability in prediction performance than its traditional NFS counterpart, but also the superiority to other compared approaches [11].

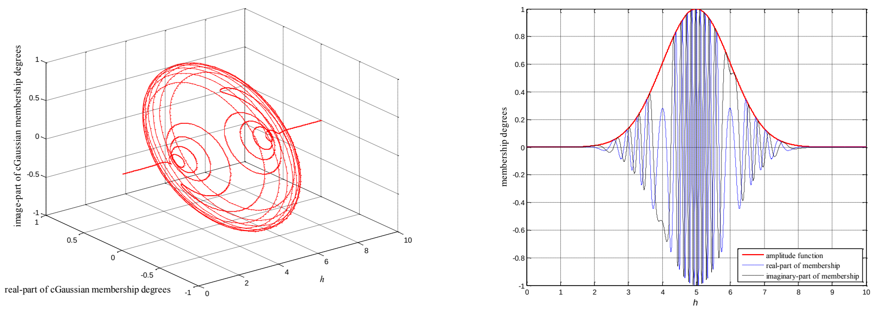
The paper is organized as follows. In Section 2, the proposed complex neuro-fuzzy using complex fuzzy sets is specified. In Section 3, the HMSPSO-RLSE hybrid learning method is given. In Section 4, two examples for time series forecasting are given. Finally, the paper is discussed and concluded.

2 Methodology for Complex Neuro-Fuzzy System

The proposed complex neuro-fuzzy system (CNFS) using complex fuzzy sets is in inheritance of the benefits of both fuzzy inference system and neural network, especially the ability of being universal approximator that can approximate any function to any accuracy theoretically [12]-[13]. For a CFS, the membership state for elements belong to the CFS is within the complex-valued unit disc of the complex plane. This is in contrast with a traditional type-1 fuzzy set, to which the membership for elements belong is within the real-valued unit interval [0, 1]. In the following, we first introduce the notation of CFS, and then present the theory of the proposed CNFS.

2.1 Complex Fuzzy Set

The theory of complex fuzzy set (CFS) can provide a new development for fuzzy system research and application [6]-[9]. The membership of a CFS is complex-valued,



(a) (b)

Fig. 1. Illustration of Gaussian-type complex fuzzy set. (a) 3-D view with the coordinates of base variable, real-part membership and imaginary-part membership. (b) Amplitude membership and imaginary-part membership vs. base variable.

different from fuzzy complex numbers developed in [14] . The membership function to characterize a CFS consists of an amplitude function and a phase function. In other words, the membership of a CFS is in the two-dimensional complex-valued unit disc space, instead of in the one-dimensional real-valued unit interval space. Thus, CFS can be much richer in membership description than traditional fuzzy set. Assume there is a complex fuzzy set S whose membership function $\mu_s(h)$ is given as follows.

$$\begin{aligned}\mu_s(h) &= r_s(h)\exp\big(j\omega_s(h)\big)\\ &= \mathrm{Re}(\mu_s(h)) + j\mathrm{Im}(\mu_s(h))\\ &= r_s(h)\cos(\omega_s(h)) + jr_s(h)\sin(\omega_s(h))\end{aligned} \qquad \textbf{(1)}$$

where $j = \sqrt{-1}$, h is the base variable for the complex fuzzy set, $r_s(h)$ is the amplitude function of the complex membership, $\omega_s(h)$ is the phase function. The property of sinusoidal waves appears obviously in the definition of complex fuzzy set. In the case that $\omega_s(h)$ equals to 0, a traditional fuzzy set is regarded as a special case of a complex fuzzy set. A novel Gaussian-type complex fuzzy is designed in the paper, and an illustration for a Gaussian-type complex fuzzy set is shown in Fig. 1. The Gaussian-type complex fuzzy set, denoted as $cGaussian(h, m, \sigma, \lambda)$, is designed as follows.

$$cGaussian(h,m,\sigma,\lambda) = r_s(h,m,\sigma)\exp(jw_s(h,m,\sigma,\lambda)) \qquad \textbf{(2a)}$$

$$r_s(h,m,\sigma) = Gaussian(h,m,\sigma) = \exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \qquad \textbf{(2b)}$$

$$w_s(h,m,\sigma,\lambda) = -\exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \times \left(\frac{h-m}{\sigma^2}\right) \times \lambda \qquad \textbf{(2c)}$$

In (2a) to (2c), h is the base variable and $\{m, \sigma, \lambda\}$ are the parameters of mean, spread and phase frequency factor for the complex fuzzy set.

2.2 Theory of Complex Neuro-Fuzzy System

Suppose an M-input-one-output complex fuzzy system is designed with K first-order Takagi-Sugeno (T-S) fuzzy rules, given as follows.

$$\begin{aligned}&\text{Rule } i\text{: IF } (x_1 \text{ is } A_1^i(h_1)) \text{ and } (x_2 \text{ is } A_2^i(h_2)) \ldots \text{and } (x_M \text{ is } A_M^i(h_M))\\ &\text{Then } z^i = a_0^i + \sum_{j=1}^{M} a_j^i h_j\end{aligned} \quad \textbf{(3)}$$

for i =1,2,...,K, where x_j is the j-th input linguistic variable, h_j is the j-th input base variables, $A^i_j(h_j)$ is the complex fuzzy set for the j-th premise condition in the i-th rule, z^i is the output of the i-th rule, and $\{a^i_j,\ i=1,2,\ldots,K$ and $j=0,1,\ldots,M\}$ are the consequent parameters. The fuzzy inference of the complex fuzzy system can be cast into neural net structure with six layers to become the complex neuro-fuzzy system (CNFS). The explanation for the six layers is specified as follows.

Layer 1: The layer is called the input layer, which receives the inputs and transmits them to the next layer directly. The input vector at time t is given as follows.

$$\mathbf{H}(t)=[h_1(t), h_2(t), \ldots, h_M(t)]^{\mathrm{T}} \quad \textbf{(4)}$$

Layer 2: The layer is called the fuzzy-set layer. Each node of layer represents a linguistic value characterized by a complex fuzzy set for the premise part of the CNFS and to calculate the complex membership degrees $\{\mu^i_j(h_j(t)),\ i=1,2,\ldots,K$ and $j=0,1,\ldots,M\}$. The complex fuzzy sets $A^i_j(h_j)$ can be designed using the Gaussian-type of complex membership function given in (2a) to (2c).

Layer 3: This layer is for the firing-strengths of fuzzy rules. The nodes perform the *fuzzy-product* operations for the firing strengths of the fuzzy rules. The firing strength of the i-th rule is calculated as follows.

$$\begin{aligned}\beta^i(t) &= \mu_1^i(h_1(t)) * \mu_2^i(h_2(t)) * \ldots * \mu_{\mathrm{M}}^i(h_M(t))\\ &= \prod_{j=1}^{M} r_j^i\left(h_j(t)\right)\exp\left(j\omega_{A_1^i\cap\ldots\cap A_M^i}\right)\end{aligned} \quad \textbf{(5)}$$

where r^i_j is the amplitude of complex membership degree for the j-th fuzzy set of the i-th rule. With (5), the $\omega_{A_1^i\cap\ldots\cap A_M^i}$ is calculated.

Layer 4: This layer is for the normalization of the firing strengths. The normalized firing strength for the i-th rule is written as follows.

$$\lambda^i(t) = \frac{\beta^i(t)}{\sum_{i=1}^{K}\beta^i(t)} = \frac{\prod_{j=1}^{M} r_j^i\left(h_j(t)\right)\exp\left(j\omega_{A_1^i\cap\ldots\cap A_M^i}\right)}{\sum_{i=1}^{K}\prod_{j=1}^{M} r_j^i\left(h_j(t)\right)\exp\left(j\omega_{A_1^i\cap\ldots\cap A_M^i}\right)} \quad \textbf{(6)}$$

Layer 5: The layer is called the consequent layer for calculating normalized consequents. The normalized consequent of the i-th rule is given as follows.

$$\begin{aligned}\xi^i(t) &= \lambda^i(t) \times z^i(t)\\ &= \lambda^i(t) \times \left(a_0^i + \sum_{j=1}^{M} a_j^i\, h_j(t)\right)\\ &= \frac{\prod_{j=1}^{M} r_j^i\left(h_j(t)\right)\exp\left(j\omega_{A_1^i\cap\ldots\cap A_M^i}\right)}{\sum_{i=1}^{K}\prod_{j=1}^{M} r_j^i\left(h_j(t)\right)\exp\left(j\omega_{A_1^i\cap\ldots\cap A_M^i}\right)} \times \left(a_0^i + \sum_{j=1}^{M} a_j^i\, h_j(t)\right)\end{aligned} \quad \textbf{(7)}$$

Layer 6: This layer is called the output layer. The normalized consequents from Layer 5 are congregated into the layer to produce the CNFS output, given as follows.

$$\xi(t) = \sum_{i=1}^{K} \xi^i(t) = \sum_{i=1}^{K} \lambda^i(t) \times z^i\,(t)$$

$$= \sum_{i=1}^{K} \frac{\prod_{j=1}^{M} r_j^i\big(h_j(t)\big)\exp\,(j\omega_{A_1^i\cap\ldots\cap A_M^i})}{\sum_{i=1}^{K}\prod_{j=1}^{M} r_j^i\big(h_j(t)\big)\exp\,(j\omega_{A_1^i\cap\ldots\cap A_M^i})} \times \left(a_0^i + \sum_{j=1}^{M} a_j^i\, h_j(t)\right) \quad \textbf{(8)}$$

Generally the output of the CNFS is complex-valued and can be expressed as follows.

$$\begin{aligned}\xi(t) &= \xi_{\mathrm{Re}}(t) + j\xi_{\mathrm{Im}}(t)\\ &= |\xi(t)| \times \exp(j\omega_\xi)\\ &= |\xi(t)| \times \cos\big(j\omega_\xi\big) + j|\xi(t)| \times \sin(j\omega_\xi)\end{aligned} \quad \textbf{(9)}$$

where $\xi_{\mathrm{Re}}(t)$ is the real part of the output of the CNFS, and $\xi_{\mathrm{Im}}(t)$ is the imaginary part. Based on (9), the complex inference system can be viewed as a complex function, expressed as follows.

$$\xi(t) = \mathrm{F}(\mathbf{H}(t), \mathbf{W}) = \mathrm{F}_{\mathrm{Re}}(\mathbf{H}(t), \mathbf{W}) + j\mathrm{F}_{\mathrm{Im}}(\mathbf{H}(t), \mathbf{W}) \quad \textbf{(10)}$$

where $\mathrm{F}_{\mathrm{Re}}(.)$ is the real part of the CNFS output, $\mathrm{F}_{\mathrm{Im}}(.)$ is the imaginary part of the output, $\mathbf{H}(t)$ is the input vector to the CNFS, $\mathbf{W}$ denotes the parameter set of the CNFS, which is composed of the subset of the premise parameters and the subset of the consequent parameters, denoted as $\mathbf{W}_{\mathrm{If}}$ and $\mathbf{W}_{\mathrm{Then}}$, respectively.

$$\mathbf{W} = \mathbf{W}_{\mathrm{If}} \cup \mathbf{W}_{\mathrm{Then}} \quad \textbf{(11)}$$

3 Multi-Swarm-Based Hybrid Learning for the Proposed CNFS

We devise a hybrid learning method including a multi-swarm-based particle swarm optimization and the recursive least squares estimator (RLSE) method to update the $\mathbf{W}_{\mathrm{If}}$ and $\mathbf{W}_{\mathrm{Then}}$, respectively. Particle swarm optimization (PSO) is a swarm-based optimization method [15]-[18], motivated by the food searching behavior of bird flocking. There are many particles in a PSO swarm. The best location for a particle in the search process is denoted as **pbest**. The particles in the swarm compete each other to become the best particle of the swarm, whose location is denoted as **gbest**. In this study, we propose a new scheme for PSO-based method, which involves multiple PSO swarms, called the Hierarchical Multi-swarm PSO (HMSPSO). This HMSPSO enhances search ability for the optimal solution. It is different from another multi-group PSO-based method [10]. Several researches in literature have been proposed to improve the easily-trapped problem at local minimum by the standard PSO and its variants. The HMSPSO is with a multi-level hierarchical architecture to balance the independent search by each swarm and the cooperative search among the swarms. The proposed HMSPSO is described by the following equations.

$$\begin{aligned}\mathbf{V}_i(k+1) = w \times \mathbf{V}_i(k) &+ c_1 \times \zeta_1 \times \big(\mathbf{pbest}_i(k) - \mathbf{L}_i(k)\big)\\ &+ c_2 \times \zeta_2 \times \left(\mathbf{gbest}_{1,q}(k) - \boldsymbol{L}_i(k)\right)\\ &+ \cdots + c_n \times \zeta_n \times \left(\mathbf{gbest}_{j,q}(k) - \boldsymbol{L}_i(k)\right)\end{aligned} \quad \textbf{(12a)}$$

$$\mathbf{L}_i(k+1) = \mathbf{L}_i(k) + \mathbf{V}_i(k+1) \quad \textbf{(12b)}$$

where $\mathbf{V}_i(k)=[v_{i,1}(k), v_{i,2}(k),\ldots, v_{i,Q}(k)]^T$ is the velocity of the i-th particle in k-th iteration, $\mathbf{L}_i(k)=[l_{i,1}(k), l_{i,2}(k),\ldots, l_{i,Q}(k)]^T$ is the location of the i-th particle, w is the inertia weight, $\{c_p, p=1,2,\ldots,n\}$ are the acceleration factors, $\mathbf{gbest}_{j,q}$ is the j-th level of q-th PSO swarm, and $\{\zeta_p, p=1,2,\ldots,n\}$ are random number between 0 and 1.
The least squares estimation (LSE) problem can be specified with a linear model, given as follows.

$$y=\theta_1 f_1(u)+\theta_2 f_2(u)+\cdots+\theta_m f_m(u)+\varepsilon \quad \textbf{(13)}$$

where y is the target, u is the input to model, $\{f_i(u), i=1,2,..,m\}$ are known functions of u, $\{\theta_i , i=1,2,\ldots,m\}$ are the_unknown parameters to be estimated, and ε is the model error. Note that the parameters $\{\theta_i , i=1,2,\ldots,m\}$ can be viewed as the consequent parameters of the proposed CNFS. Observed samples can be collected to use as training data for the proposed CNFS. The training data (TD) is denoted as follows.

$$\mathrm{TD}=\{(u_i, y_i), i=1,2,\ldots,N\} \quad \textbf{(14)}$$

where (u_i, y_i) is the i-th data pair in the form of (*input, target*).Substituting data pairs into (13), we have a set of N linear equations in matrix notation, given below.

$$\mathbf{y}=\mathbf{A}\boldsymbol{\theta}+\boldsymbol{\varepsilon} \quad \textbf{(25)}$$

where $\boldsymbol{\theta}=[\theta_1, \theta_2, \ldots,\theta_m]^T$, $\mathbf{y}=[y_1, y_2, \ldots,y_N]^T$, $\boldsymbol{\varepsilon}=[\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]^T$, and $\mathbf{A}$ is the matrix formed by $\{f_i(u_j), i=1,2,\cdots,m \text{ and } j=1,2,\ldots,N\}$. The optimal estimator for $\boldsymbol{\theta}$ can be obtained with the recursive least squares estimator (RLSE) method, given below.

$$\mathbf{P}_{k+1}=\mathbf{P}_k-\frac{\mathbf{P}_k\mathbf{b}_{k+1}(\mathbf{b}_{k+1})^T\mathbf{P}_k}{1+(\mathbf{b}_{k+1})^T\mathbf{P}_k\mathbf{b}_{k+1}} \quad \textbf{(16a)}$$

$$\boldsymbol{\theta}_{k+1}=\boldsymbol{\theta}_k+\mathbf{P}_{k+1}\mathbf{b}_{k+1}(y_{k+1}-(\mathbf{b}_{k+1})^T\boldsymbol{\theta}_k) \quad \textbf{(16b)}$$

for $k=0,1,\ldots,(N-1)$, where $[\mathbf{b}^T_{\mathbf{k}}, y_k]$ is the k-th row of $[\mathbf{A}, \mathbf{y}]$. To start the RLSE algorithm, we set $\boldsymbol{\theta}_0$ to zero vector and $\mathbf{P}_0=\alpha\mathbf{I}$, where α must be a large value and $\mathbf{I}$ is the identity matrix.

For parameter learning, the proposed CNFS predictor is trained by the hybrid HMSPSO-RLSE learning method, where the HMSPSO is used to update the premise parameters and the RLSE is used to adjust the consequent parameters. The training procedure for the proposed HMSPSO-RLSE method is given as follows.

Step 1. Collect training data. Some portion of the data is used for training, and the rest is for testing.

Step 2. Update the premise parameters by the HMSPSO in (12a) and (12b).

Step 3. Update the consequent parameters by the RLSE in (16a) and (16b), in which the row vector $\mathbf{b}$ and the vector $\boldsymbol{\theta}$ are arranged as follows.

$$\mathbf{b}_{k+1}=[\mathbf{bb}^1(k+1) \quad \mathbf{bb}^2(k+1) \quad \cdots \quad \mathbf{bb}^K(k+1)] \quad \textbf{(17)}$$

$$\mathbf{bb}^i(k+1)=[\lambda^i \quad h_1(k+1)\lambda^i \quad \cdots \quad h_M(k+1)\lambda^i] \quad \textbf{(18)}$$

$$\boldsymbol{\theta}_k=[\boldsymbol{\tau}_k^1 \quad \boldsymbol{\tau}_k^2 \quad \cdots \quad \boldsymbol{\tau}_k^K] \quad \textbf{(19)}$$

$$\boldsymbol{\tau}_k^i=[a_0^i(k) \quad a_1^i(k) \quad \cdots \quad a_M^i(k)] \quad \textbf{(20)}$$

Table 1. Settings for the HMSPSO-RLSE hybrid learning method. (Example 1)

HMSPSO		RLSE	
Number of premise parameters (Dimensions of particle)	18	Number of consequent parameters	16
Swarm size	300	$\boldsymbol{\theta}_0$	Zeros(1, 9)
Initial particle position	Random in $[0, 1]^{18}$	$\mathbf{P}_0$	$\alpha\mathbf{I}$
Initial particle velocity	Random in $[0, 1]^{18}$	α	10^8
acceleration parameters $\{c_1, c_2, c_3\}$	$\{2, 2, 2\}$	$\mathbf{I}$	16×16 identity matrix
Swarm number of PSO	3		

Step 4. Calculate the CNFS output in (10).

Step 5. Calculate the cost in MSE defined below. Note that because the time series forecasting problem is in real-valued domain, only the real part of the CNFS output is involved in MSE.

$$\mathrm{MSE} = \frac{1}{N}\sum_{t=1}^{N}(e(t))^2 = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \mathrm{Re}(\xi(t)))^2 \tag{21}$$

Step 6. Compare the costs of all particles. Update **pbest** and **gbest** in the multiple swarms. If stopping criteria satisfied, **gbest** is the optimal premise parameters for the CNFS and stop. Otherwise, go back to Step 2 and continue the procedure.

4 Experimentation for the Proposed Approach

Example 1- Star Brightness Time Series

The star brightness time series is used to test the proposed approach for forecasting. The dataset records the daily brightness of a variable star on 600 successive midnights [19], denoted as $\{y(t), t=1,2,\ldots,600\}$. The range of dataset is normalized to the unit interval [0, 1]. We use the first 300 samples for training and the remaining 300 samples for testing. The input vector is arranged as $\mathbf{H}(t)=[y(t\text{-}3), y(t\text{-}2), y(t\text{-}1)]^T$ and the target is given as $y(t)$ for the predicting model. Each input linguistic variable has two complex fuzzy sets, and the grid partition is designed in the input space formed by the three variables. Thus, eight T-S fuzzy rules in (3) are designed for the CNFS predictor and the NFS predictor, respectively. The cost function is designed with MSE in (21). For parameter learning, the HMSPSO algorithm in (12a) and (12b) is used to update the antecedent parameters and the consequent parameters is adjusted by the RLSE. The settings for the HMSPSO-RLSE hybrid learning method are given in Table 1. The proposed CNFS is compared to its NFS counterpart. The Gaussian-type complex fuzzy sets in (2a) to (2c) are designed for the CNFS and the traditional Gaussian fuzzy sets in (2b) are used for the NFS. Moreover, the proposed approach is compared to other approaches in [11]. The experiments with 20 trails for each are conducted. The performance comparison in average and standard deviation is shown

Table 2. Performance Comparison. (Example 1)

Method	MSE (std)	
	Training phase	Testing phase
TSK-NFIS [11]	3.14×10^{-4} (5.90×10^{-2})	3.31×10^{-4} (6.09×10^{-2})
AR [11]	3.14×10^{-4} (5.81×10^{-2})	3.22×10^{-4} (6.01×10^{-2})
NAR [11]	3.20×10^{-4} (5.96×10^{-2})	3.12×10^{-4} (5.92×10^{-2})
Neural Net [11]	3.01×10^{-4} (5.78×10^{-2})	3.11×10^{-4} (5.91×10^{-2})
PSO-RLSE for NFS	1.99×10^{-4} (4.45×10^{-6})	3.24×10^{-4} (3.27×10^{-5})
PSO-RLSE for CNFS	1.98×10^{-4} (1.03×10^{-5})	2.80×10^{-4} (1.95×10^{-5})
HMSPSO-RLSE for CNFS	1.98×10^{-4} (9.91×10^{-6})	2.72×10^{-4} (1.79×10^{-5})
Note that the results above are based on 20 trails for average and standard deviation.		

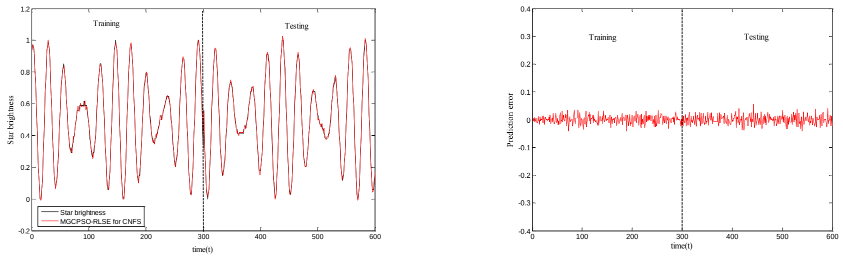


Fig. 2. (a) Prediction by the proposed approach for star brightness. (b) Prediction error.

in Table 2. For one of the 20 trials, the response by the proposed CNFS and its prediction error are shown in Figs. (2a) and (2b).

Example 2- Oil Price Time Series

The oil price time series records the average annual price of oil, which is a small data set with 128 samples [20], denoted as $\{y(t), t=1,2,\ldots,128\}$. The range of data is normalized to the interval [0, 1]. The first 64 samples are used for training the proposed CNFS and the remaining 64 samples for testing. The input vector is arranged as $\mathbf{H}(t)=[\ y(t\text{-}2), y(t\text{-}1)]^T$ and the target is given as $y(t)$ for the predicting model. Each input has two Gaussian type complex fuzzy sets in (2a) to (2c). There are four T-S fuzzy rules in (3) for the CNFS and its NFS counterpart, respectively. The CNFS has 12 premise parameters and 12 consequent parameters. The premise parameters are updated by the HMSPSO algorithm and the consequent parameters are tuned by the RLSE, as stated previously. The cost function is designed with the concept of MSE in (21). Each experiment is performed with 20 trails, and the performance comparison in average and standard deviation is shown in Table 3.

5 Discussion and Conclusion

The proposed complex neuro-fuzzy system (CNFS) with complex fuzzy sets has been presented for the problem of time series forecasting. The CNFS is trained by the newly devised HMSPSO-RLSE hybrid learning method for the purpose of accurate forecasting. Two examples has been demonstrated for time series forecasting, and the

Table 3. Performance Comparison. (Example 2)

Method	MSE (std)	
	Training phase	Testing phase
TSK-NFIS [11]	4.31×10^{-3} (3.42×10^{-1})	3.31×10^{-2} (6.29×10^{-1})
AR [11]	5.45×10^{-3} (3.84×10^{-1})	3.22×10^{-2} (6.38×10^{-1})
NAR [11]	4.99×10^{-3} (3.68×10^{-1})	3.12×10^{-2} (7.39×10^{-1})
Neural Net [11]	4.69×10^{-3} (3.56×10^{-1})	3.11×10^{-2} (6.50×10^{-1})
PSO-RLSE for NFS	1.98×10^{-3} (1.52×10^{-4})	2.59×10^{-2} (3.27×10^{-2})
PSO-RLSE for CNFS	2.03×10^{-3} (4.29×10^{-4})	1.63×10^{-2} (5.44×10^{-3})
HMSPSO-RLSE for CNFS	2.21×10^{-3} (2.20×10^{-4})	1.34×10^{-2} (1.34×10^{-3})
Note that the results above are based on 20 trails for average and standard deviation.		

proposed approach has shown very excellent prediction performance. This confirms our thought that the property of complex fuzzy sets (CFSs) designed into the proposed system can augment the functional mapping ability in forecasting accuracy. The notion of CFS is in contrast with that of standard type-1 fuzzy set in membership depiction. It is clearly contrasted that the membership for elements belong to a CFS is characterized within the complex-valued unit disc of the complex plane while the membership of a type-1 fuzzy set is within the real-valued unit interval between 0 and 1. Based on this contrasted property, the CNFS computing paradigm designed with complex fuzzy sets can expand the mapping flexibility for predication capability in terms of forecasting accuracy. For parameter learning of the proposed CNFS, with the divide-and-conquer concept we separate the system parameters into two smaller subsets for If-part and Then-part parameters, and then the HMSPSO-RLSE is used to train the proposed system for fast learning purpose. The idea is that the smaller the search space the easier and faster the solution can be found. This has been justified with the experiments in the two examples. In this newly devised hybrid learning method, we have implemented the multi-swam-based PSO with the RLSE algorithm, showing very good performance in terms of fast learning convergence and forecasting accuracy. For performance comparison, the proposed approach has been compared to the NFS counterpart and other approaches, the results, shown in Tables 2 and 3, show that the proposed CNFS is superior to its NFS counterpart (trained by the hybrid PSO-RLSE method). Moreover, for the forecasting accuracy, the CNFS predictor outperforms the compared approaches. Through this study, the excellence of the proposed CNFS computing approach to time series forecasting has been exposed.

Acknowledgment

This research work is supported by the National Science Council, Taiwan, ROC, under the Grant contract no. NSC99-2221-E-008-088.

References

1. Jang, S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, pp. 665-685 (1993)

2. Herrera, L. J., Pomares, H., Rojas, I., Guillén, A., González, J., Awad, M., Herrera, A.: Multigrid-based fuzzy systems for time series prediction: CATS competition. Neurocomputing, vol. 70, pp. 2410-2425 (2007)
3. Boyacioglu, M.A., Avci, D.: An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the prediction of stock market return: The case of the Istanbul Stock Exchange. Expert Systems with Applications, vol. 37, pp. 7908-7912 (2010)
4. Deng, X., Wang, X.: Incremental learning of dynamic fuzzy neural networks for accurate system modeling. Fuzzy Sets and Systems, vol. 160, pp. 972-987 (2009)
5. Mousavi, S.J., Ponnambalam, K., and Karray, F.: Inferring operating rules for reservoir operations using fuzzy regression and ANFIS. Fuzzy Sets and Systems, vol. 158, pp. 1064-1082 (2007)
6. Ramot, D., Milo, R., Friedman, M., Kandel, A.: Complex fuzzy sets. IEEE Transactions on Fuzzy Systems, vol. 10, pp. 171-186 (2002)
7. Dick, S.: Toward complex fuzzy logic. IEEE Transactions on Fuzzy Systems, vol. 13, pp. 405-414 (2005)
8. Moses, D., Degani, O., Teodorescu, H. N., Friedman, M., Kandel, A.: Linguistic coordinate transformations for complex fuzzy sets. In: Fuzzy Systems *Conference Proceedings.* vol.3, pp. 1340-1345 (1999)
9. Ramot, D., Milo, R., Friedman, M., Kandel, A.: Complex fuzzy logic. IEEE Transactions on Fuzzy Systems, vol. 11, pp. 450-461 (2003)
10. Niu, Ben., Zhu, Y., He, X., Wu, H.: M*CPSO*: A multi-swarm cooperative particle swarm optimizer. Applied Mathematics and Computation, vol. 185, pp. 1050-1062 (2007)
11. Graves, D., Pedrycz, W.: Fuzzy prediction architecture using recurrent neural networks. Neurocomputing, vol. 72, pp. 1668-1678 (2009)
12. Castro, J.L.: Fuzzy logic controllers are universal approximators. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 25, pp. 629-635 (1995)
13. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks, vol. 2, pp. 359-366 (1989)
14. Buckley, J.J.: Fuzzy complex numbers. Fuzzy Sets and Systems, vol. 33, pp. 333-345 (1989)
15. Eberhart, R., Kennedy J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science. (1995)
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks Proceedings. (1995)
17. Yuhui, S., Eberhart, R.C.: Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 Congress on Evolutionary Computation. (2001)
18. Mansour, M.M., Mekhamer, S.F., El-Kharbawe, N.E.S.: A Modified Particle Swarm Optimizer for the Coordination of Directional Overcurrent Relays. IEEE Transactions on Power Delivery, vol. 22, pp. 1400-1410 (2007)
19. Time Series Data Library, Physics, Daily brightness of a variable star, http://www-personal.buseco.monash.edu.au/hyndman/TSDL/S.
20. Time Series Data Library, Micro-Economics, Oil prices in constant 1997 dollars, http://www-personal.buseco.monash.edu.au/hyndman/TSDL/S.