Neurocomputing 99 (2013) 467–476

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A novel self-organizing complex neuro-fuzzy approach to the problem of time series forecasting

Chunshien Li [a,*], Tai-Wei Chiang [a], Long-Ching Yeh [b]

[a] Laboratory of Intelligent Systems and Applications, Department of Information Management, National Central University, Taiwan, ROC
[b] Department of Electrical Engineering, National Central University, Taiwan, ROC

ARTICLE INFO

Article history:
Received 12 December 2011
Received in revised form
2 April 2012
Accepted 4 July 2012
Communicated by P. Zhang
Available online 14 August 2012

Keywords:
Complex fuzzy set (CFS)
Complex neuro-fuzzy system (CNFS)
Hybrid learning
Self-organization
Time series forecasting

ABSTRACT

A self-organizing complex neuro-fuzzy intelligent approach using complex fuzzy sets (CFSs) is presented in this paper for the problem of time series forecasting. CFS is an advanced fuzzy set whose membership function is characterized within a unit disc of the complex plane. With CFSs, the proposed complex neuro-fuzzy system (CNFS) that acts as a predictor has excellent adaptive ability. The design for the proposed predictor comprises the structure and parameter learning stages. For structure learning, the FCM-Based Splitting Algorithm for clustering was used to determine an appropriate number of fuzzy rules for the predictor. For parameter learning, we devised a learning method that integrates the method of particle swarm optimization and the recursive least squares estimator in a hybrid and cooperative way to optimize the predictor for accurate forecasting. Four examples were used to test the proposed approach whose performance was then compared to other approaches. The experimental results indicate that the proposed approach has shown very good performance and accurate forecasting.

1. Introduction

A time series is a temporal sequence of data measured at successive times, such as daily stock market closing price, unemployment rate, economic growth rate, and many others. Accurate and reliable forecasting for the future trend is a crucial issue in investment decision making for modern organizations. Since modern business and economic activities are frequently vibrant, it is extremely challenging to capture the dependency between the future and past. Such time series are inherently nonlinear and complex in nature. Since traditional approaches require mathematical models that are usually simplified through linearization for analytical purpose, they are hardly capable of attacking such problems with satisfactory performance. Recently, several intelligent approaches [1–4] with fuzzy system and/or artificial neural network (ANN) were used to the problem of time series forecasting, due to their flexible modeling capability and excellent nonlinear prediction ability. For such intelligent approaches, optimization algorithms are used to design and fine-tune such adaptive prediction models. For example, Luna et al. (2007) [1] suggested a constructive fuzzy model based on the Takagi–Sugeno (T–S) fuzzy system [5] and the expectation maximization technique for time series prediction. Rojas et al. (2008) [2] designed a hybrid model using ANN and autoregressive moving average (ARMA) models to solve time series problems. Wong et al. (2010) [3] proposed a novel adaptive neural network (ADNN) with the adaptive metrics of inputs and a new mechanism for admixture of outputs for time series prediction. Although the excellent mapping ability and link-type distributed structure of ANN have been proved in the literatures [1–4,6–8], ANN is considered as a black box modeling method. On the other hand, a fuzzy system is composed of a set of fuzzy If–Then rules which can easily be mapped into practical domains of applications by the human's thought, expertise and experience [9]. With the merit of neural network and fuzzy system, Jang (1993) [10] presented an adaptive network based fuzzy inference system (ANFIS), where the backpropagation method and the recursive least-square estimator are integrated as a hybrid learning method to adapt the free parameters. Recently, Ramot et al. [11,12] proposed a new aspect of complex fuzzy sets (CFSs) to the fuzzy theory. In traditional crisp sets, the set-element relationship is either appurtenant or not. However, such a set–element definition of crisp sets was not so practical in the complex real world that the concept of fuzzy sets was derived [9,13] to relax and enrich the set-element relationship, where membership degrees can be defined in the unit real-valued interval [0,1]. Furthermore, the concept of CFSs is a brand new viewpoint to the fuzzy theory. A CFS is characterized by a complex-valued membership function that comprises an amplitude function and a phase function [11,12]. The membership degrees of CFS are in a two-dimensional unit disc of the complex plane. With complex-valued property of membership degrees, CFSs can enrich the

* Corresponding author.
E-mail address: jamesli@mgt.ncu.edu.tw (C. Li).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.07.014

set-element relationship, and they can be used to design adaptive system to promote the nonlinear mapping ability and achieve high flexibility and adaptability. Though the concept of CFSs was proposed [11,12,14–17], they are still not intuitively understandable due to the fact that the CFSs are beyond human experience, especially the meaning of phase in membership degree. Zhang et al. (2009) [16] focused on arguing about operation properties for CFSs without conferring on the design method on complex fuzzy set and its reasoning procedures. An adaptive neural complex fuzzy inferential system for the problem of time series forecasting was presented by Chen et al. (2011) [17], where the design of CFSs is based on the equivalence that the phase of CFSs is their support and such a design lacks convincing explanation for CFSs.

In this paper, for time series forecasting, we proposed a self-organizing complex neural fuzzy system (CNFS) using CFSs and a clustering method to design an intelligent predictor for which we devised a hybrid learning algorithm to fine-tune the predictor for accurate prediction. Such a proposed intelligent approach to time series forecasting is a fully data-driven method for modeling and prediction. Based on our previous work [18–21], we applied a class of novel Gaussian complex fuzzy sets to the Takagi–Sugeno (T–S) fuzzy rules of the proposed predictor in this paper to study its adaptability gain for time series forecasting. The design for the proposed intelligent predictor includes two learning stages: the structure learning stage for the structure arrangement of the predictor and the parameter learning stage for the optimal performance by the predictor. Initially, the size and structure of rule base for the CNFS are automatically determined by a clustering algorithm, called the fuzzy c-mean (FCM) Based Splitting Algorithm (FBSA) [22]. To adapt the free parameters of the CNFS predictor, a PSO–RLSE hybrid learning algorithm is used, where the particle swarm optimization (PSO) [23,24] and the recursive least squares estimator (RLSE) [25,26] are integrated in a hybrid and cooperative way. Such a hybrid learning method is based on the divide-and-conquer principle where the whole set of free parameters of the CNFS is spiritually separated into two smaller subsets of If-part and Then-part parameters. The PSO–RLSE method is applied to the CNFS in the way that the PSO is used to update the subset of If-part parameters and the RLSE is used to adjust the consequent parameters recurrsively. After learning, the CNFS predictor can have the prediction performance as accurate as possible. Four examples of time series are used to test the proposed approach, whose experimental results are then compared to other approaches [27–29,33,35–42] in the literature for performance comparison. Through the experimental results, it is shown that the proposed approach is superior to the compared approaches.

We organize the rest of the paper as follows. In Section 2, the proposed CNFS model for prediction is specified. In Section 3, the strategy is explained for two learning phases: the structure and parameter learning phases. In Section 4, we use four examples to test the proposed approach, whose performance is compared to other approaches. Finally, the paper is discussed and concluded.

2. System architecture for complex neuro-fuzzy system

With CFSs, the rationale of the proposed complex neuro-fuzzy system (CNFS) is specified in this section, which is composed of several first-order Takagi–Sugeno (T–S) fuzzy rules with multiple inputs and a single output, given below:

$$\begin{aligned}\text{Rule } i: \quad & \text{IF } x_1 = {}^is_1(h_1(t)) \text{ and } x_2 = {}^is_2(h_2(t)) \cdots \text{and } x_M = {}^is_M(h_M(t)) \\ & \text{THEN } {}^iy(t) = {}^ia_0 + {}^ia_1 h_1(t) + \cdots + {}^ia_M h_M(t) \end{aligned} \tag{1}$$

for $i=1,2,\ldots,K$, where the index i stands for the ith fuzzy rule; x_l is the lth linguistic variable whose base variable is denoted as h_l; $h_1(t)$, $h_2(t)$, $\cdots$, $h_M(t)$ are the numerical inputs to the CNFS at time t;${}^iy(t)$ is the nominal output of the ith rule; $\{{}^is_0, {}^is_1, \cdots, {}^is_M)$ are the premise CFSs of the ith fuzzy rule whose consequent parameters are $\{{}^ia_0, {}^ia_1, \cdots, {}^ia_M)$. According to the theory of neuro-fuzzy system, the complex-fuzzy inference of the model in (1) can be cast into a neural-network structure to become a complex neuro-fuzzy system [19,21]. The output of complex neuro-fuzzy system can be expressed as follows:

$$\xi(t) = \sum_{i=1}^{K} {}^i\gamma(t)\, {}^iy(t), \tag{2}$$

where

$${}^i\gamma(t) = \frac{{}^i\beta(t)}{\sum_{i=1}^{K} {}^i\beta(t)}, \tag{3}$$

$${}^i\beta(t) = \prod_{l=1}^{M} {}^i\mu_l(h_l(t)), \tag{4}$$

where ${}^i\gamma(t)$ is the normalized firing strength for the ith rule; ${}^i\beta(t)$ is the firing strength of the ith fuzzy rule, in which the *fuzzy-product* operator is used for the t-norm calculation; ${}^i\mu_l(\cdot)$ is the complex-valued membership function of ${}^is_l(\cdot)$ that is the lth complex fuzzy set of the ith rule.

In this paper, we present a class of Gaussian complex fuzzy sets, whose membership function, denoted as $\mu_{\mathrm{GCFS}}(h, m, \sigma, \lambda)$, is designed as follows:

$$\mu_{\mathrm{GCFS}}(h, m, \sigma, \lambda) = r_{\mathrm{GCFS}}(h, m, \sigma)\exp(j\omega_{\mathrm{GCFS}}(h, m, \sigma, \lambda)), \tag{5}$$

$$r_{\mathrm{GCFS}}(h, m, \sigma) = \exp\left[-\frac{(h-m)^2}{2\sigma^2}\right], \tag{6}$$

$$\omega_{\mathrm{GCFS}}(h, m, \sigma, \lambda) = -\exp\left[-\frac{(h-m)^2}{2\sigma^2}\right]\left(\frac{h-m}{\sigma^2}\right)\lambda \tag{7}$$

where $j=\sqrt{-1}$; h is the base variable for the complex fuzzy set; $r_{\mathrm{GCFS}}(\cdot)$ is the amplitude function of the complex-valued membership function; $\omega_{\mathrm{GCFS}}(\cdot)R$ is the phase function; m, σ, λ are the parameters of mean, spread and phase factor of the Gaussian CFS, respectively. An illustration for a Gaussian complex fuzzy set is shown in Fig. 1.

The parameters of the Gaussian CFSs in the premises of fuzzy rules are called the premise parameters. In (2), we have shown the mathematical relation between the input vector $\mathbf{H}(t)$ and the complex-valued output of CNFS $\xi(t)$. The complex-valued output can be further expressed as follows:

$$\begin{aligned}\xi(t) &= \xi_{\mathrm{Re}}(t) + j\xi_{\mathrm{Im}}(t) \\ &= |\xi(t)|\exp(j\omega_\xi) \\ &= |\xi(t)|\cos(\omega_\xi) + j|\xi(t)|\sin(\omega_\xi),\end{aligned} \tag{8}$$

where$\xi_{\mathrm{Re}}(t)$ and $\xi_{\mathrm{Im}}(t)$ indicate the real and imaginary parts of the complex-valued output, respectively, whose magnitude and phase are given below:

$$|\xi(t)| = \sqrt{(\xi_{\mathrm{Re}}(t))^2 + (\xi_{\mathrm{Im}}(t))^2}, \tag{9}$$

$$\omega_\xi(t) = \tan^{-1}\left(\frac{\xi_{\mathrm{Re}}(t)}{\xi_{\mathrm{Im}}(t)}\right). \tag{10}$$

According to Eq. (8), the proposed CNFS can be regarded as a complex-valued function, defined as follows:

$$\xi(t) = f(\mathbf{H}(t), \mathbf{W}) = f_{\mathrm{Re}}(\mathbf{H}(t), \mathbf{W}) + jf_{\mathrm{Im}}(\mathbf{H}(t), \mathbf{W}), \tag{11}$$

where $f_{\mathrm{Re}}(\cdot)$ and $f_{\mathrm{Im}}(\cdot)$ represent the real and imaginary functions, respectively, that are characterized by the proposed CNFS; $\mathbf{W}$ denotes the complete set of parameters of the CNFS, which is composed of the subset of the premise parameters (denoted as $\mathbf{W}_{\mathrm{If}}$) and the subset of the consequent parameters (denoted as $\mathbf{W}_{\mathrm{Then}}$), respectively. Note that the parameters and model structure of CNFS can affect the performance by the proposed CNFS for the application of time series forecasting. In the following, we continue to specify the design of the CNFS by machine-learning methods.

3. Machine learning for the proposed CNFS

To set up the model structure of the proposed approach, we can devise a self-organizing learning method, which has two learning phases: the structure and parameter learning phases. For structure learning, we use the FCM-Based Splitting Algorithm (FBSA) to determine the initial structure of knowledge base for the proposed CNFS, after which the parameter-learning stage follows. For parameter learning, we devise the PSO–RLSE method to fine-tune the parameter set $\mathbf{W}$ that is viewed spiritually as the composition of $\mathbf{W}_{\mathrm{If}}$ and $\mathbf{W}_{\mathrm{Then}}$.

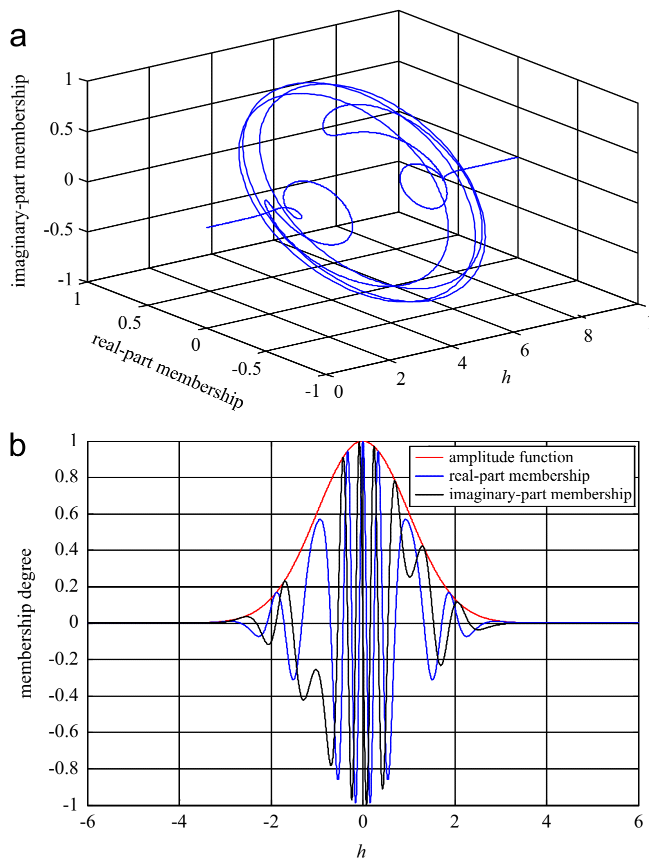

Fig. 1. Illustration of a Gaussian complex fuzzy set: (a) 3-D view with the coordinates of base variable, real-part membership and imaginary-part membership and (b) amplitude, real-part and imaginary-part membership functions vs. base variable.

3.1. Structure learning for CNFS

With a training dataset, the proposed CNFS using the FBSA can self-construct its initial knowledge base. The FBSA method is used to automatically settle on an appropriate number of clusters, based on each of which a fuzzy rule can be generated for the proposed CNFS. The locations of the generated clusters are used as the initial setting for the premise parameters of the complex fuzzy sets in the fuzzy rules given in Eq. (1). The FBSA algorithm was first proposed by Sun et al. (2004) [22], which is a clustering algorithm based on the fuzzy c-mean (FCM) method and a validity index for the clustering results. With the FBSA, we can identify the cluster that is with the worst score at each step of the algorithm and split it into two new clusters while keeping the others. The procedure of FBSA is described in Fig. 2.

To determine the worst cluster, a score function$S(i)$ is defined as follows:

$$S(i) = \frac{\sum_{k=1}^{n} u_{ki}}{\text{number of data in cluster } i}, \tag{12}$$

where $S(i)$ QUOTE is the score of the ith cluster. In general, when $S(i)$ is small, the corresponding cluster tends to contain a large number of data with low-density degree. The cluster with the smallest score needs to split into two new clusters. Suppose a dataset is separated into c clusters whose center set is denoted by $\mathbf{V}=\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_c\}$, where $\mathbf{v}_i=[v_{i,1}\ v_{i,2}\ldots v_{i,s}]^{\mathrm{T}}$ QUOTE is the position vector of the ith cluster center. The set of membership degrees by the FCM is denoted as $\mathbf{U}=u_{ij}$, $i=1,2,\ldots,n$ and $j=1,2,\ldots,c$. Furthermore, the FBSA uses the following validity index [22] to decide the optimal number of clusters:

$$V_d(\mathbf{U},\mathbf{V},c) = \mathrm{Scat}(c) + \frac{\mathrm{Sep}(c)}{\mathrm{Sep}(C_{\max})} \tag{13}$$

where

$$\mathrm{Scat}(c) = \frac{1/c\sum_{i=1}^{c}\|\sigma(v_i)\|}{\|\sigma(x)\|};\ \mathrm{Sep}(c) = \frac{D_{\min}^2}{D_{\max}^2}\sum_{i=1}^{c}\left(\sum_{j=1}^{c}\|v_i - v_j\|^2\right)^{-1};$$

$$D_{\min} = \min_{i\neq j}\|v_i - v_j\|;\ D_{\max} = \max_{i,j}\|v_i - v_j\|.$$

3.2. Parameter learning for CNFS

The parameter-learning stage follows the structure-learning stage specified above. For parameter learning, we devise the hybrid PSO–RLSE method. First proposed by Eberhart and Kennedy [23] and Kennedy and Eberhart [24] the method of particle swarm optimization (PSO) has been used to solve optimization problems widely. The concept of PSO stems from the behavior for food searching by a swarm of birds. Each particle in a PSO swarm can be regarded as a bird seeking food. The location of food can be viewed as the solution to the underlying optimization problem. All the particles form a population called a swarm. In the searching space, each particle owns its position and velocity, which can be

Step 1: Set C_{min} and C_{max}.
Step 2: Initialize C_{min} cluster centers (**V**).
Step 3.1: For c = C_{min} to C_{max}
Step 3.2: Apply the basic FCM algorithm to update the membership matrix (**U**) and the cluster centers (**V**).
Step 3.3: Test for convergence; if not converged, go to **Step 3.2**
Step 3.4: Compute validity value for $V_d(c)$.
Step 3.5: Compute score $S(i)$ for each cluster; split the worst cluster.
Step 4: Obtain the cluster number c_f, where the validity index $V_d(c_f)$ is optimal.

Fig. 2. Implementation procedure of FBSA, where c_f is the optimal number of clusters, based on which we can create the same number of T–S fuzzy If–Then rules for the CNFS.

adjusted by the search experience of its own and the swarm. During the search process, the best location of the ith particle is denoted as $\mathbf{Pbest}_i$ QUOTE and that of the complete swarm is denoted as **Gbest**. Each particle of the swarm changes its search direction according to its best location and the swarm's best location, as follows:

$$\boldsymbol{\varphi}_i(t+1)=\omega\boldsymbol{\varphi}_i(t)+c_1 rand_1(\mathbf{Pbest}_i(t)-\mathbf{P}_i(t))+c_2 rand_2(\mathbf{Gbest}(t)-\mathbf{P}_i(t)), \tag{14}$$

$$\mathbf{P}_i(t+1)=\mathbf{P}_i(t)+\boldsymbol{\varphi}_i(t+1) \tag{15}$$

where $\boldsymbol{\varphi}_i(t)$ is the velocity of the ith particle at the tth learning iteration; c_1,c_2 are the parameters for PSO; ω is the inertia weight; $rand_1$, $rand_2$ are random numbers between 0 and 1; $\mathbf{P}_i(t)$ is the location of the ith particle.

Basically, the method of recursive least squares estimation (RLSE) that is from the least squares estimation (LSE) [25,26] is very efficient for the problem of linear regression optimization. Given a set of training data ($\mathbf{u}_j$,y_j), $j=1, 2,\ldots,n$, a linear regression model can be given as

$$y_j=\sum_{i=1}^{m}\theta_i f_i(\mathbf{u}_j)+\varepsilon_j, \tag{16}$$

where $\mathbf{u}_j$ QUOTE is the input vector to the model whose target is y_j; $f_i(\cdot)$,$i=1,2,\ldots,m$ are known functions of $\mathbf{u}$; θ_i,$i=1, 2,\ldots,m$ are the free parameters to be estimated; ε_j is the model error. With the RLSE method, the solution can be obtained recursively. At iteration k, the parameters can be calculated using the following equations:

$$\mathbf{P}_{k+1}=\mathbf{P}_k-\frac{\mathbf{P}_k\mathbf{b}_{k+1}(\mathbf{b}_{k+1})^{\mathrm{T}}\mathbf{P}_k}{1+(\mathbf{b}_{k+1})^{\mathrm{T}}\mathbf{P}_k\mathbf{b}_{k+1}}, \tag{17}$$

$$\boldsymbol{\theta}_{k+1}=\boldsymbol{\theta}_k+\mathbf{P}_{k+1}\mathbf{b}_{k+1}(y_{k+1}-(\mathbf{b}_{k+1})^{\mathrm{T}}\boldsymbol{\theta}_k), \tag{18}$$

where k is the iteration index, $k=0,1,2,\ldots,n-1$. Before starting the RLSE, $\boldsymbol{\theta}_0$ and $\mathbf{P}_0$ need to be initialized. In the paper, we set $\boldsymbol{\theta}_0$ to zero vector and $\mathbf{P}_0=\alpha\mathbf{I}$, where α is a large positive value and $\mathbf{I}$ is the identity matrix.

With the algorithms of PSO and RLSE, we can devise the hybrid PSO–RLSE method to fine-tune the parameters of CNFS in the way where the PSO is used to evolve the premise parameters ($\mathbf{W}_{\mathrm{If}}$), based on which the RLSE is used to update the consequent parameters ($\mathbf{W}_{\mathrm{Then}}$). The PSO and RLSE work together in a hybrid and cooperative way. The procedure of the PSO–RLSE hybrid method for parameter learning is given below.

Step 1. Collect the sample data. Some of the datasets are used for training and the rest are for testing.

Step 2. Use FBSA to determine the number of fuzzy rules for CNFS. The clusters' centers and spreads are set as one initial position of PSO particles. Initialize other PSO particles.

Step 3. Update $\mathbf{W}_{\mathrm{If}}$ by PSO.

Step 4. Update $\mathbf{W}_{\mathrm{Then}}$ by RLSE.

Step 5. Calculate forecast by the CNFS predictor.

Step 6. Calculate cost in MSE for each particle. Compare the costs of all particles for **Pbest** and **Gbest**. Since the problem of time series forecasting is in real-valued domain, only the real part of the CNFS output is involved in MSE that is defined below:

$$\mathrm{MSE}=\frac{1}{n}\sum_{t=1}^{n}(e(t))^2=\frac{1}{n}\sum_{t=1}^{n}(y(t)-\mathrm{Re}(\xi(t)))^2 \tag{19}$$

where n is the size of training data pairs.

Step 7. Update **Pbest** of each particle and **Gbest** of the PSO swarm.

Step 8. If any of stopping criteria is satisfied, stop the training process. Otherwise, go to **Step 3**.

In general, the training process goes on until any of stopping conditions is satisfied. Such stopping conditions can include the accuracy of performance better than some preset threshold and the number of learning iterations used up. We consider only the latter as the termination condition, because the former is impractical to set in this study.

4. Experimentation for the proposed approach

We use four examples to test the proposed approach, whose results are then compared to other approaches in the literature. These real-world examples include the sequences of Taiwan Semiconductor Manufacturing (TSMC) stock price, weekly exchange rates between US dollar (USD) and Taiwan dollar (TWD), IBM stock price, and sunspots.

Example 1. —*time series of TSMC STOCK Price*

The time series of Taiwan Semiconductor Manufacturing (TSMC) stock price is used in this example to test the forecasting performance of the proposed approach. The TSMC is the largest dedicated independent semiconductor foundry, which is the important weight stock of the Taiwan Stock Exchange Capitalization Weighted Stock Index. From the website of Yahoo Finance (2011) [30], the closing-price data of TSMC were collected from January 4th, 2000 to December 31st, 2002. There are 1304 samples in total, denoted as $\{y(t), t=1,2,\ldots,1304\}$, where t is the time index. Then, these data are arranged in the form of (input, target) for 1302 data pairs, which are denoted as $\{(\mathbf{H}(i), d(i)), i=1,2,\ldots,1302\}$, where $\mathbf{H}(i)=[y(t-1), y(t)]^{\mathrm{T}}$; $d(i)=y(t+1)$; $t=i+1$. The first 3-year data, which contain 808 samples, are used for training and the remaining 496 samples are used for testing. For structure learning, the FBSA clustering method is used to determine the number of fuzzy rules for the CNFS predictor. The settings of the FBSA are given in Table 1. The validity–index curve and values are shown in Fig. 3 and Table 2, indicating that five clusters are determined to be the best.

Therefore, the number of fuzzy rules is set to be five. Based on the locations of the five generated clusters in the input space, we can make five fuzzy If–Then rules, each of which has two inputs and one output. Each input has five Gaussian complex fuzzy sets. The cost function in MSE is used. For parameter learning, the CNFS predictor is trained by the PSO–RLSE hybrid learning method, whose settings are given in Table 3.

After learning, the CNFS predictor was tested with testing data, whose experimental result is summarized in Table 4, where the performance by the proposed approach is compared to other approaches [27–29]. The measure of root mean square error (RMSE) is used as the performance index. Moreover, the CNFS approach is

Table 1
Settings of the FBSA clustering method.

Parameter	Value	Remark
$C_{\min}$	2	Minimum number of clusters
$C_{\max}$	10	Maximum number of clusters
m	2	FCM fuzzifier
$\varepsilon_{\mathrm{FCM}}$	10^{-9}	Stopping threshold for FCM

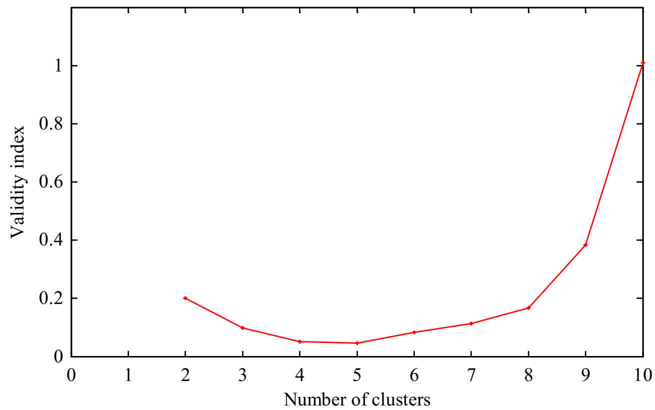

Fig. 3. Validity index curve by FBSA clustering (time series of TSMC stock price).

Table 2
Validity indices for different numbers of clusters by FBSA (time series of TSMC stock price).

Number of clusters	Validity index
2	0.2006
3	0.0983
4	0.0513
5	**0.0456**
6	0.0832
7	0.1120
8	0.1676
9	0.3831
10	1.009

Table 3
Settings of the PSO–RLSE method (time series of TSMC stock price).

PSO	
Dimensions of PSO particles	30
Swarm size	300
Particle velocity initialization	Random in $[0,1]^{30}$
Particle position initialization	Random in $[0,1]^{30}$
Acceleration factors, (c_1, c_2)	(2, 2)
Inertia weight, w	0.8
Maximum number of iterations	300
RLSE	
Number of consequent parameters	15
$\boldsymbol{\theta}_0$	15×1 zero vector
$\mathbf{P}_0$	$\alpha\mathbf{I}$
α	10^8
$\mathbf{I}$	15×15 identity matrix

Table 4
Performance comparison (time series of TSMC stock price).

Method	Rules	RMSE
Chen [27]	N/A	12.77
Weighted fuzzy time series models [28]	N/A	6.72
Fuzzy time-series model [29]	N/A	1.32
SVR	N/A	2.24
ANFIS	5	1.26
NFS (by PSO only)	5	7.43
CNFS (by PSO only)	5	5.68
NFS (by PSO–RLSE)	5	1.22
CNFS (by PSO–RLSE) (Proposed)	**5**	**1.14**

compared to its NFS counterpart that was trained by the PSO method only and the PSO–RLSE hybrid method respectively for performance comparison. The adaptive network based fuzzy inference system (ANFIS) (Jang, 1993) and the support vector regression (SVR) are also involved in the performance comparison. We implemented the ANFIS and the SVR with the fuzzy logic toolbox of Matlab and the LIBSVM toolbox (Chang and Lin, 2011), respectively.

The prediction response and error by the proposed CNFS for testing phase are shown in Figs. 4 and 5, respectively. Through the experimental results, the proposed approach shows superior performance to the compared approaches for the forecasting of TSMC stock price.

Example 2. *—time series of weekly exchange rate between USD and TWD*

The time series of weekly exchange rates between US dollar (USD) and Taiwan dollar (TWD) is used in the example. The prediction of exchange rate is a hot issue for the field of financial time series. We collected 254 data of weekly USD–TWD exchange rate from January 7th, 2007 to November 6th, 2011 from the website of OANDA (2011) [31]. The time series is denoted as $\{y(i), i=1,2,\ldots,254\}$. The range of the dataset is normalized into the unit interval [0,1] for the purpose of implementation. Afterwards, these data are arranged in the form of (input, target) for 252 data pairs, which are denoted as $\{(\mathbf{H}(i), d(i)), i=1,2,\ldots,252\}$, where $\mathbf{H}(i)=[y(t-1),\ y(t)]^{\mathrm{T}}$; $d(i)=y(t+1)$; and $t=i+1$. The first

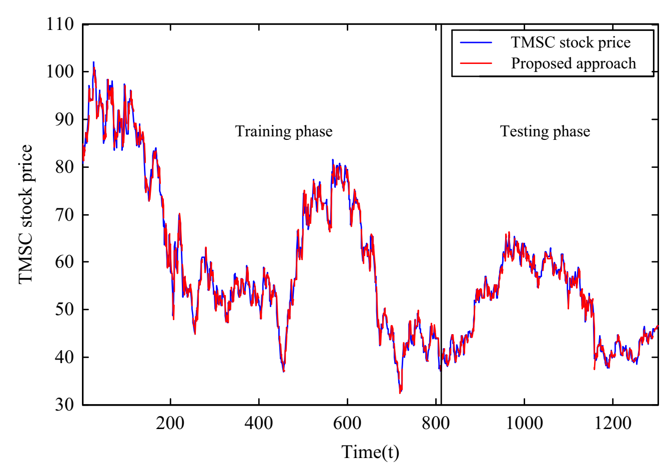

Fig. 4. Prediction response by the proposed approach (time series of TSMC stock price).

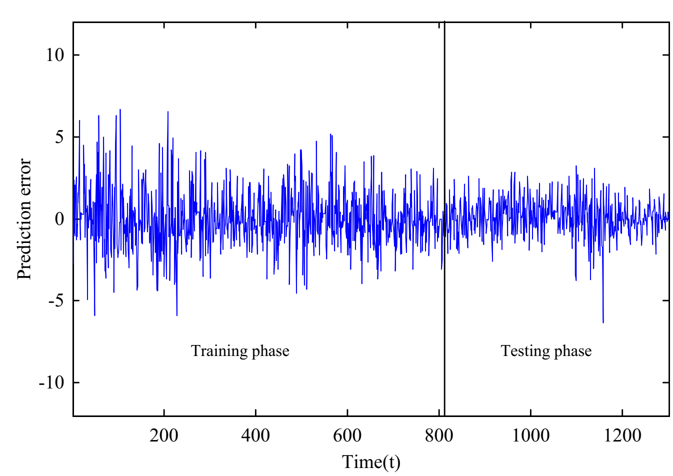

Fig. 5. Prediction error by the proposed approach (time series of TSMC stock price).

200 data pairs are used for training and the rest data pairs are for testing. To settle on an appropriate number of fuzzy rules, the FBSA clustering method is performed to generate appropriate clusters, based on which the fuzzy rules for the CNFS predictor are created. The settings for the FBSA method are given to be the same as in Table 1. The validity indices and validity index curve are listed in Table 5 and Fig. 6, respectively. Five clusters are determined to be the best.

For the proposed CNFS prediction model, five first-order T–S fuzzy If–Then rules with two inputs and one output were generated, where each input linguistic variable has five Gaussian complex fuzzy sets. The cost function is designed with MSE. The PSO–RLSE hybrid learning method whose settings are given to be the same as in Table 2 was used to fine-tune the CNFS, whose experimental result is listed in Table 6, where the performance by the proposed approach is compared to other approaches, such as the NFS counterpart of the CNFS, ANFIS and SVR. The measure of RMSE is used as the performance index.

Table 5
Validity indices for different clusters (USD–TWD exchange rate).

Number of clusters	Validity index
2	0.2920
3	0.1159
4	0.0667
5	**0.0514**
6	0.0663
7	0.0839
8	0.1428
9	0.5300
10	1.0138

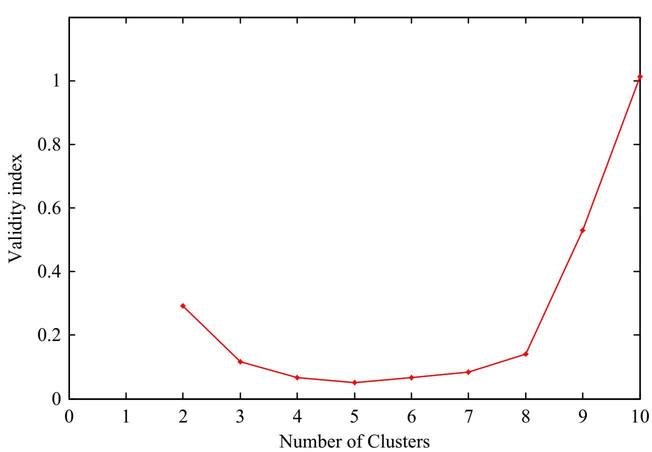

Fig. 6. Validity index curve of clusters (USD–TWD exchange rate).

Table 6
Performance comparison (USD–TWD exchange rate).

Method	Rules	RMSE
SVR	N/A	0.1744
ANFIS	5	0.3538
NFS (by PSO only)	5	0.3351
CNFS (by PSO only)	5	0.2645
NFS (by PSO–RLSE)	5	0.0398
CNFS (by PSO–RLSE) (Proposed)	**5**	**0.0283**

The prediction response and error by the proposed approach are shown in Figs. 7 and 8, respectively. Through the experimental results, the proposed approach has shown excellent performance for the forecasting of weekly exchange rate between USD and TWD.

Example 3. *—time series of daily IBM stock price*

In this example, we use the real-world time series of IBM stock price to test the proposed approach. The time series series is scaled by a factor of 1/100. From January 1st, 1980 to October 8th, 1992, 3333 samples with the daily closing stock price of IBM were collected from [32]. The range of the dataset was normalized to the interval [0,1], as done by [33]. After normalization, the dataset is denoted as $\{y(t), t=1, 2, \ldots, 3333\}$, where t is the time index. The data of the time series are arranged into the form of (input, target) for 3331 data pairs, denoted as $\{(\mathbf{H}(i), d(i)), i=1,2,\ldots,3331\}$, where $\mathbf{H}(i)=[y(t-1),\ y(t)]$; $d(i)=y(t+1)$; and $t=i+1$. $\mathbf{H}(i)$ is the input vector to the proposed model from the ith pair; $d(i)$ is the corresponding target. The first 1500 data pairs are used for training and the remaining are for testing. For structure learning, the FBSA clustering method is applied to determine an appropriate number of fuzzy rules for the CNFS. The settings of the FBSA clustering method are given to be the same as in Table 1. The results by the FBSA are shown in Table 7 and Fig. 9, indicating that four clusters are determined to be the best.

Based on the locations of the generated four clusters in the input space, we can create four fuzzy If–Then rules for the CNFS with two inputs and one output. Each input has four Gaussian complex fuzzy sets. For parameter learning, we applied the PSO–RLSE hybrid learning method, by which the premise parameters are tuned by the PSO and the consequent parameters are updated

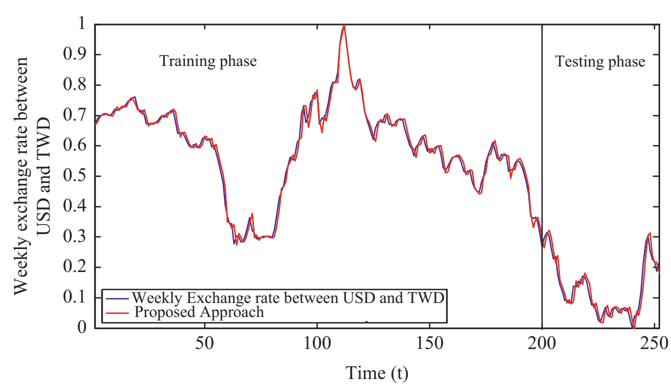

Fig. 7. Prediction response by the proposed approach (USD–TWD exchange rate).

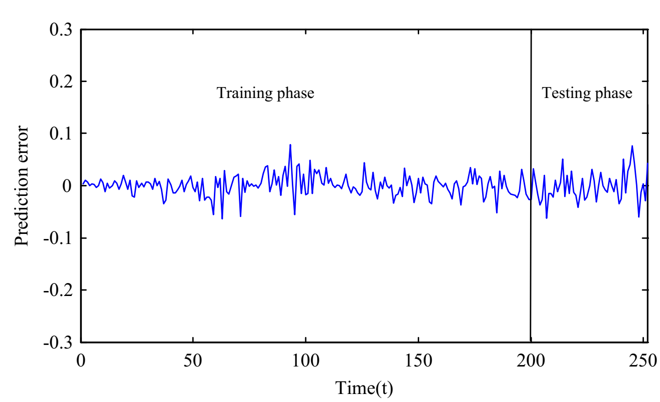

Fig. 8. Prediction error by the proposed approach (USD–TWD exchange rate).

Table 7
Validity indices for different clusters (time series of daily IBM stock price).

Number of clusters	Validity index
2	0.1405
3	0.0515
4	**0.0388**
5	0.0572
6	0.1006
7	0.1458
8	0.2592
9	0.6425
10	1.0050

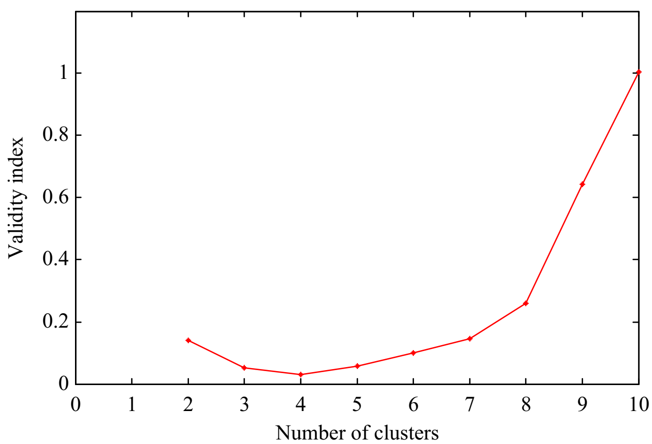

Fig. 9. Validity index curve of clusters (time series of daily IBM stock price).

Table 8
Settings of the PSO–RLSE method (time series of daily IBM stock price).

PSO	
Dimensions of PSO particles	24
Swarm size	300
Particle velocity initialization	Random in $[0,1]^{24}$
Particle position initialization	Random in $[0,1]^{24}$
Acceleration factors (c_1, c_2)	(2, 2)
Inertia weight w	0.8
Maximum number of iterations	300
RLSE	
Number of consequent parameters	12
$\boldsymbol{\theta}_0$	12×1 zero vector
$\mathbf{P}_0$	$\alpha\mathbf{I}$
α	10^8
$\mathbf{I}$	12×12 identity matrix

by the RLSE. The settings for the PSO–RLSE method are given in Table 8.

The cost function was designed with MSE. For performance comparison, the CNFS approach is compared to other approaches [33] and its NFS counterpart that was trained by the PSO method only and the hybrid PSO–RLSE method. The performance comparison is shown in Table 9, where the proposed approach shows superior performance to the compared approaches. The prediction response in the real range by the proposed approach after learning is shown in Fig. 10. The prediction error that is the difference between the actual stock price and its forecast by the proposed CNFS is shown in Fig. 11.

Table 9
Performance comparison (time series of daily IBM stock price).

Method	MSE	
	Training phase	Testing phase
TSK–NFIS [33]	2.10×10^{-4}	2.22×10^{-4}
Autoregressive [33]	2.00×10^{-4}	1.82×10^{-4}
NAR [33]	2.02×10^{-4}	1.61×10^{-3}
Neural network [33]	2.05×10^{-4}	2.36×10^{-4}
NFS (by PSO only)	5.42×10^{-3}	7.16×10^{-3}
CNFS (by PSO only)	7.11×10^{-4}	8.55×10^{-4}
NFS (by PSO–RLSE)	2.58×10^{-4}	2.83×10^{-4}
CNFS (by PSO–RLSE) (Proposed)	$\mathbf{9.35 \times 10^{-5}}$	$\mathbf{1.78 \times 10^{-4}}$

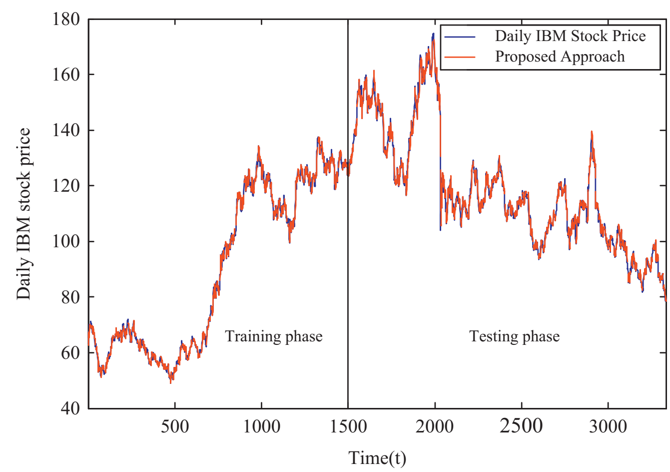

Fig. 10. Prediction response by the proposed approach (time series of daily IBM stock price).

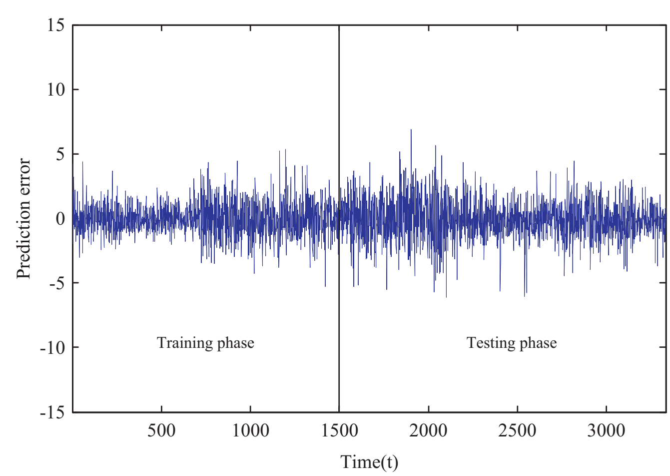

Fig. 11. Prediction error by the proposed approach (time series of daily IBM stock price).

Example 4. —*sunspot time series*

In this example, we use the monthly smoothed sunspot time series that contains the sequence of a month of smoothed sunspots from November 1984 to June 2000 to test the proposed approach. There are 2000 samples in total that were obtained from the SIDC (World Data Center for the Sunspot Index) [34]. The range of the sunspot time series was normalized into the interval [0,1] for fair performance comparison to other approaches in the literatures [35–42]. The dataset after normalization is denoted as $\{y(t), t=1,2,\ldots,2000\}$, where t is the time index. The first 1000 samples are used to train the proposed model and the remaining samples are used for testing. For the preparation of training data, the dataset is arranged into the form of (input, target), which are denoted as $\{(\mathbf{H}(i), d(i)), i=1,2,\ldots,1998\}$, where $\mathbf{H}(i)=[y(t-1), y(t)]$; $d(i)=y(t+1)$; and $t=i+1$. Note that $\mathbf{H}(i)$ is the input vector to the CNFS model and $d(i)$ is the corresponding target. For structure learning, the FBSA clustering method is used to determine the initial knowledge of the proposed CNFS. The settings in Table 1 are used for the FBSA whose results are shown in Table 10 and Fig. 12, indicating that five clusters are best for the CNFS.

For parameter learning, the cost function is design with MSE and the settings of the PSO–RLSE method are given to be the same as in Table 3. For performance comparison, three performance indices that are MSE, RMSE and normalized MSE (NMSE) are used. The definition of NMSE is given as follows:

$$\mathrm{NMSE} = \frac{\sum_{t=1}^{n} (d(t) - \mathrm{Re}((\xi t)))^2}{\sum_{t=1}^{n} (d(t) - \bar{d}))^2} \tag{20}$$

where $\bar{d} = \sum_{t=1}^{n} d(t)/n$.

The performance comparison is shown in Table 11, where the proposed approach shows much better performance than the compared approaches in the testing phase. The prediction response and error by the proposed CNFS are shown in Figs. 13 and 14, respectively.

Table 10
Validity indices for different clusters (sunspot time series).

Number of clusters	Validity index
2	0.2571
3	0.0939
4	0.0572
5	**0.0409**
6	0.0621
7	0.0955
8	0.4453
9	0.7765
10	1.0082

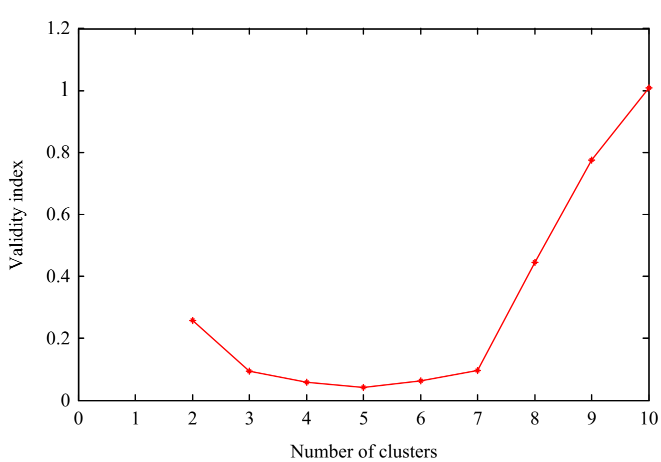

Fig. 12. Validity index curve of clusters (sunspot time series).

Table 11
Performance comparison (sunspot time series).

Method	MSE	RMSE	NMSE
WP–MLP [35]	–	–	0.125
McNish–Lincoln [36]	–	–	0.008
Sello-nonlinear method [37]	–	–	0.340
Waldmeier [37]	–	–	0.560
Denkmayr [38]	–	–	1.85
RBF-OLS [39]	–	–	0.046
LLNF [39]	–	–	0.032
ERNN [40]	–	–	0.0028
MLP [41]	–	–	0.0979
Elman–NARX [42]	1.4078×10^{-4}	0.0119	5.9041×10^{-4}
NFS (by PSO–RLSE)	8.5112×10^{-5}	0.0092	3.5701×10^{-4}
CNFS (by PSO–RLSE) (Proposed)	$\mathbf{4.1490\times10^{-5}}$	**0.0064**	1.7404×10^{-4}

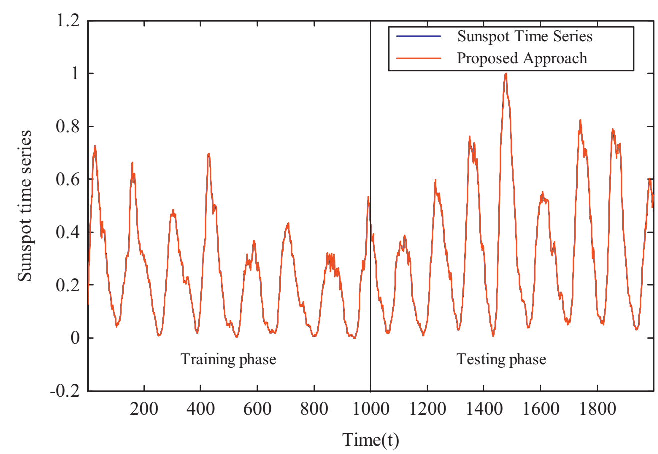

Fig. 13. Prediction response by the proposed approach (sunspot time series).

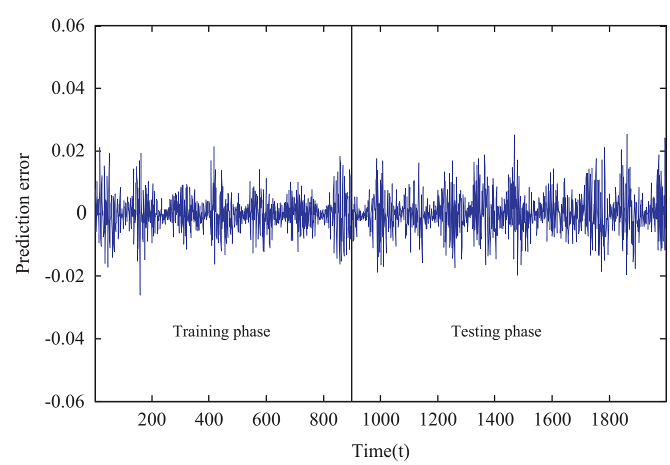

Fig. 14. Prediction error by the proposed approach (sunspot time series).

5. Discussion and conclusion

We have presented a self-organizing complex neuro-fuzzy prediction approach that combines the theories of complex fuzzy set (CFS) and neuro-fuzzy system to form the CNFS model for applications of time series forecasting. The design of the CNFS predictor that is composed of several T–S fuzzy rules includes a two-stage learning procedure, where the FBSA clustering method is used for the structure-learning stage and then the PSO–RLSE hybrid learning method is used to fine-tune the intelligent predictor. For fast learning, we separated spiritually the free parameters of CNFS into two smaller subsets: the subsets of premise parameters and consequent parameters and applied the devised PSO–RLSE method to adapt these parameters. The experimental results show that the proposed approach outperforms the compared approaches in the literature [27–29,33,35–42]. The excellent performance by the proposed approach derives from the non-linear mapping ability of the proposed CNFS model, where CFSs are important elements in achieving this ability because they can provide more degrees of freedom for the learning flexibility of the proposed models than regular fuzzy sets. Four examples of real-world time series have been used to test the performance of the proposed approach.

For Example 1 (time series of TSMC stock price), as shown in Table 4, the proposed approach has the prediction performance 1.14 in RMSE for the testing phase. Such a performance is 91% better than the approach in [27] whose performance is 12.77 in RMSE, 84% better than the weighted fuzzy time series model [28] whose performance is 6.72 in RMSE, 14% better than the fuzzy time-series model [29] whose performance is 1.32 in RMSE, 49% better than the SVR whose performance is 2.24 in RMSE, and 10% better than the ANFIS whose performance is 1.26 in RMSE. Moreover, in Table 4, it is also compared for the CNFS to its NFS counterpart, based on the same learning method. For example, the proposed CNFS (1.14 in RMSE) has superior forecasting accuracy to NFS (1.22 in RMSE). For Example 4 (sunspot time series), the proposed approach has the prediction performance 1.7404×10^{-4} in NMSE for the testing phase. This performance is 99% better than the WP–MLP [35] whose performance is 0.125 in NMSE, 98% better than the McNish–Lincoln [36] whose performance is 0.008 in NMSE, 93% better than the ERNN [40] whose performance is 0.0028 in RMSE, 71% better than the Elman–NARX [42] whose performance is 5.9041×10^{-4} in NMSE, 51% better than the NFS whose performance is 3.5701×10^{-4} in NMSE, and almost 100% better than the other compared methods. Similar results can be observed in the other experiments in Examples 2 and 3, as shown in Tables 6 and 9 respectively. This confirms our thought that the application of CFSs can truly enhance the forecasting ability of the proposed model. And, the experimental results demonstrate that the proposed approach has shown excellent performance in accuracy and has potential in time series forecasting.

In addition, on the basis of same CNFS model, we can compare the performance with the proposed PSO–RLSE hybrid learning method to that with the PSO method only. For example, in Example 1 that is for time series of TSMC stock price, under the same CNFS model, the PSO–RLSE method has the prediction performance 1.14 RMSE for the testing phase. Such a performance is 80% better than that with the PSO method alone whose performance is 5.68 in RMSE, as shown in Table 4. In Example 2 that is for time series of weekly exchange rate between USD and TWD, based on the same CNFS model, the PSO–RLSE method (0.2845 RMSE, testing phase) performs much better than PSO method (0.0283), that is 90% better than the PSO method alone, as shown in Table 6. Similar experimental effect is also observed in Example 3, as shown in Table 9. These experimental results indicate that the proposed PSO–RLSE hybrid learning has much stronger power to promote the forecasting performances than the PSO method alone.

Finally, we conclude the paper with three results in this study. First, we have successfully applied the theory of CFSs to the neuro-fuzzy computing system to form the proposed CNFS models that have been applied to the examples of real-world time series forecasting. The proposed approach can open a new window to fuzzy-system-based research. Second, we have applied the self-organizing data-driven technique to the proposed CNFS approach to determine an adequate model structure automatically. In this way, we can set up a proposed model as objectively as possible to alleviate human intervention and increase the efficiency of modeling. Third, for fast learning we have presented the PSO–RLSE hybrid learning method to fine-tune the proposed models. Through the experimental results in the examples of real-world time series, the excellence of the self-organizing CNFS computing approach to time series forecasting has been exposed.

References

[1] I. Luna, S. Soares, R. Ballini, A constructive-fuzzy system modeling for time series forecasting, in: International Joint Conference on Neural Networks, 2007, pp. 2908–2913.

[2] I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L. Herrera, H. Pomares, L. Marquez, M. Pasadas, Soft-computing techniques and ARMA model for time series prediction, Neurocomputing 71 (2008) 519–537.

[3] W.K. Wong, M. Xia, W.C. Chu, Adaptive neural network model for time-series forecasting, Eur. J. Oper. Res. 207 (2010) 807–816.

[4] L. Zhao, Y. Yang, PSO-based single multiplicative neuron model for time series prediction, Expert Syst. Appl. 36 (2009) 2805–2812.

[5] M. Sugeno, T. Takagi, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1985) 116–132.

[6] A. Cherif, H. Cardot, R. Boné, SOM time series clustering and prediction with recurrent neural networks, Neurocomputing 74 (2011) 1936–1944.

[7] G.P. Zhang, A neural network ensemble method with jittered training data for time series forecasting, Inf. Sci. 177 (2007) 5329–5346.

[8] L. Manevitz, A. Bitar, D.L. Givoli, Neural network time series forecasting of finite-element mesh adaptation, Neurocomputing 63 (2005) 447–463.

[9] L.A. Zadeh, Fuzzy logic and approximate reasoning, Synthese 30 (1975) 407–428.

[10] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (1993) 665–685.

[11] D. Ramot, R. Milo, M. Friedman, A. Kandel, Complex fuzzy sets, IEEE Trans. Fuzzy Syst. 10 (2002) 171–186.

[12] D. Ramot, M. Friedman, G. Langholz, A. Kandel, Complex fuzzy logic, IEEE Trans. Fuzzy Syst. 11 (2003) 450–461 2003.

[13] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.

[14] D. Moses, O. Degani, H.-N. Teodorescu, M. Friedman, A. Kandel, Linguistic coordinate transformations for complex fuzzy sets, in: IEEE International Conference Proceedings on Fuzzy Systems, vol. 3, 1999, pp. 1340–1345.

[15] S. Dick, Toward complex fuzzy logic, IEEE Trans. Fuzzy Syst. 13 (2005) 405–414.

[16] G. Zhang, T.S. Dillon, K.-Y. Cai, J. Ma, J. Lu, Operation properties and δ-equalities of complex fuzzy sets, Int. J. Approximate Reasoning 50 (2009) 1227–1249.

[17] Z. Chen, S. Aghakjani, J. Man, S. Dick, ANCFIS: a neurofuzzy architecture employing complex fuzzy sets, IEEE Trans. Fuzzy Syst. 19 (2011) 305–322.

[18] C. Li, T.-W. Chiang, Complex fuzzy computing to time series prediction—a multi-swarm PSO learning approach, Lect. Notes Artif. Intell. 6592 (2011) 242–251.

[19] C. Li, T.-W. Chiang, Function approximation with complex neuro-fuzzy system using complex fuzzy sets—a new approach, New Gener. Comput. 29 (2011) 261–276.

[20] C. Li, T.-W. Chiang, Complex fuzzy model with PSO–RLSE hybrid learning approach to function approximation, Int. J. Intell. Inf. Database Syst. 5 (2011) 409–430.

[21] C. Li, T.-W. Chiang, Complex neuro-fuzzy self-learning approach to function approximation, Lect. Notes Artif. Intell. 5991 (2010) 289–299.

[22] H. Sun, S. Wang, Q. Jiang, FCM-based model selection algorithms for determining the number of clusters, Pattern Recognition 37 (2004) 2027–2037.

[23] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.

[24] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks Proceedings, vol. 4, 1995, pp. 1942–1948.

[25] L. Ljung, E. Ljung, System Identification: Theory for the User, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1987.

[26] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Upper Saddle River, New Jersey, USA, 1997.

[27] S.-M. Chen, Forecasting enrollments based on fuzzy time series, Fuzzy Sets Syst. 81 (1996) 311–319.

[28] H.-K. Yu, Weighted fuzzy time series models for TAIEX forecasting, Physica A 349 (2005) 609–624.

[29] T.-L. Chen, C.-H. Cheng, J.H. Hia, Fuzzy time-series based on Fibonacci sequence for stock price forecasting, Physica A 380 (2007) 377–390.

[30] Website of Yahoo Finance, Taiwan Semiconductor Manufacturing (TSMC) Stock Price, 2011, Available at 〈http://finance.yahoo.com/q?s=ˆ2330.TW〉 (accessed 30.10.11).

[31] OANDA, Time Series of Weekly Exchange Between USD and TWD, 2011, Available at ⟨http://www.oanda.com/currency/historical-rates⟩ (accessed 15.11.11).
[32] R.J. Hyndman, Time Series Data Library, Physics, Daily Closing Price of IBM Stock, 1 January 1980–8 October 1992, Available at ⟨http://www-personal.buseco.monash.edu.au/hynman/TSDL/S⟩(accessed 20.03.12).
[33] C.-J. Lu, T.-S. Lee, C.-C. Chiu, Financial time series forecasting using independent component analysis and support vector regression, Decision Support Syst. 47 (2009) 115–125.
[34] SIDC (World Data Center for the Sunspot Index), Sunspot Time Series, 2012, Available at ⟨http://sidc.oma.be/index.php3⟩ (accessed 20.03.12).
[35] K. Teo, L. Wang, Z. Lin, Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization, Comput. Sci. 2074 (2001) 310–317.
[36] A.G. McNish, J.V. Lincoln, Prediction of sunspot numbers, Trans. Am. Geophys. Union 30 (1949) 673.
[37] S. Sello, Solar cycle forecasting: a nonlinear dynamics approach, Astron. Astrophys. 377 (2001) 312–320.
[38] K. Denkmayr, P. Cugnon, About sunspot number medium-term predictions, in: G. Heckman, et al., (Eds.), Solar–Terrestrial Prediction Workshop V, Hiraiso Solar Terrestrial Research Center, Japan, 1997, p. 103.
[39] A. Gholipour, B.N. Araabi, C. Lucas, Predicting chaotic time series using neural and neurofuzzy models: a comparative study, Neural Process. Lett. 24 (2006) 217–239.
[40] Q. Ma, Q. Zheng, H. Peng, T. Zhong, L. Xu, Chaotic time series prediction based on evolving recurrent neural networks, in: Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2007.
[41] T. Koskela, M. Lehtokangas, J. Saarinen, K. Kaski, Time series prediction with multilayer perceptron, FIR and Elman neural networks, in: Proceedings of the World Congress on Neural Networks, 1996, pp. 491–496.
[42] M. Ardalani-Farsa, S. Zolfaghari, Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks, Neurocomputing 73 (2010) 2540–2553.

Chunshien Li received the Ph.D. degree in Electrical Engineering and Computer Science in 1996 from the University of Illinois at Chicago, USA. He is currently an associate professor at the Department of Information Management, National Central University, Taiwan. He was with the Department of Computer Science and Information Engineering, National University of Tainan, and the Department of Electrical Engineering, Chang Gung University. Prof. Li was also the Chief of Information System Group of Computer Center, and the Director of Teacher Development Center, National University of Tainan, Taiwan. Prof. Li received the Research Award from the National Science Council, Taiwan, in 1999 and the Research Award of Excellent Teacher from Chang Gung University, Taiwan, in 2000. He has been first author of more than 100 academic and technical publications. He has been biographized in Marquis Who's Who in Science and Engineering, in Medicine and Healthcare, in Asia, and in the World. He has served as a referee of paper review for several IEEE journals and international journals and a PC member for several international academic conferences. His current research interests include computational intelligence, machine learning and optimization, neural fuzzy systems, prediction, pattern recognition, intelligent modeling and forecasting, intelligent control, and intelligent signal processing.

Tai-Wei Chiang is a Ph.D. student at the Department of Information Management, National Central University, Taiwan. He received his B.S. degree in Information Management from Nanhua University, Chiayi, Taiwan, and his M.S. degree in Information Management from National Formosa University, Yunlin, Taiwan, in 2006 and 2008, respectively. His research interests include fuzzy logic, neural networks, intelligent system, machine learning, learning algorithms, and neuro-fuzzy inference systems.

Long-Ching Yeh was born in Taipei, Taiwan, in 1949. He received his B.S. degree from National Taiwan Normal University in 1971 and his M.S. degree from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1975, both majored in physics. From 1980 to 1984, he studied at the University of Florida. In 1984, he received his Ph.D. degree in Electrical Engineering from the University of Florida, in Gainesville, Florida, USA. Dr. Yeh has worked in the industry for more than 26 years. He is currently with the E.E. Department of National Central University as a visiting expert. His research interests are power electronics, renewable energy, analog circuit design and semiconductor device physics.