Complex Neurofuzzy ARIMA Forecasting—A New Approach Using Complex Fuzzy Sets

Chunshien Li and Tai-Wei Chiang

***Abstract*—A novel complex neurofuzzy autoregressive integrated moving average (ARIMA) computing approach is presented for the problem of time-series forecasting. The proposed approach integrates a complex neurofuzzy system (CNFS) using complex fuzzy sets (CFSs) and ARIMA models to form the proposed computing model, which is called the CNFS-ARIMA. The output of CNFS-ARIMA is complex-valued, of which the real and imaginary parts can be used for two different functional mappings. This is the so-called dual-output property. There is no fuzzy IF–THEN rule in the genesis of CNFS-ARIMA. For the formation of CNFS-ARIMA, structure learning and parameter learning are involved to self-organize and self-tune the CNFS-ARIMA. A class of CFSs is used to describe the premise parts of fuzzy IF–THEN rules, whose consequent parts are specified by ARIMA models. CFS is an advanced fuzzy set whose membership degrees are complex-valued within the unit disc of the complex plane. With the synergetic merits of CNFS and ARIMA, CNFS-ARIMA models have excellent nonlinear mapping capability for time-series forecasting. A number of benchmark time series are used to test the proposed approach, whose results are compared with those by other approaches. Moreover, real-world financial time series, such as the National Association of Securities Dealers Automated Quotation (NASDAQ), the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX), and the Dow Jones Industrial (DJI) Average Index, are used for the proposed approach to perform the dual-output forecasting experiments. The experimental results indicate that the proposed approach shows excellent performance.**

***Index Terms*—Complex fuzzy set (CFS), complex neurofuzzy system (CNFS), hybrid learning, self-organization, time-series forecasting.**

I. Introduction

TIME-SERIES forecasting is an interesting and challenging research issue with growing attention to various areas [1]–[4]. The motivation of time-series forecasting is to explore possible functional relationships and discover statistical regularities for the statistical observations of an underlying time series so that useful decision making can be made in advance. However, accurate and reliable forecasting for the future trend is usually difficult and laborious in real-world problems. Traditional approaches that require mathematic models can hardly attack such problems with satisfactory performance. Recently, various different intelligent approaches have been presented to deal with time-series forecasting problems [1]–[12]. In this paper, for the problem of time-series forecasting, we propose a novel intelligent approach, using complex fuzzy sets (CFSs), neurofuzzy modeling, autoregressive integrated moving average (ARIMA) modeling, and machine learning. Neurofuzzy systems (NFSs) [13]–[17] are intelligent computing models. An NFS that is a synergy of a neural network (NN) and a fuzzy inference system (FIS) combines the flexible learning capability with link-type distributed structure of NN and the fuzzy inference ability of FIS. FISs can extract a human's experience and knowledge in the form of IF–THEN rules that can be easily explained by a human. NFSs have been useful for time-series forecasting [6], [13], [18], owing to their abilities to capture the underlying relationship from the observed data. Theoretically, NNs and FISs have been proved universal approximators that can approximate any continuous function with arbitrarily desired accuracy in a compact set [19], [20]. However, some related research works have brought up inconsistent results [21]–[24]. For example, Taskaya-Temizel and Casey [24] discovered that the linear autoregressive (AR) models could outperform NNs in some cases. For this phenomenon, the reason may be that the underlying data are around linear without much disorder, and thus, NNs cannot be expected to obtain better performance than linear models for linear relationships [25].

For time-series analysis and modeling, ARIMA [26], which has enjoyed fruitful applications in forecasting for economic, engineering, astronomy, and many others [3], [27]–[30], is one of the most popular approaches. For ARIMA, the model is generally referred to as an ARIMA(p, d, q) model, where (p, d, q) are nonnegative integers that refer to the order of the autoregressive, integrated, and moving average parts of the model, respectively. Lags of the differenced series appearing in the model are called "autoregressive" terms; lags of the forecast errors are called "moving average" terms (MA). A time series can be differenced so that the differenced series, which is also called an "integrated" (I) version of the time series, is stationary in a statistical sense. With different combinations, ARIMA can represent several modeling types, including AR, MA, autoregressive moving average (ARMA), autoregressive integrated, and ARIMA. For the modeling of ARIMA, homogeneous nonstationary behavior can be handled by supposing some suitable difference of the process so that the differenced version of the time series may become stationary to make forecasting more accurate. Although ARIMA has good flexibility in modeling, it has the restriction in linearity form, where a linear functional relationship for current and past values and white noise is assumed. For this reason, the traditional form of ARIMA is found to be not sufficient to cope with complex and nonlinear real-world problems with satisfactory performance. Many researchers used intelligent

Manuscript received July 12, 2011; revised January 27, 2012 and June 16, 2012; accepted October 2, 2012. Date of publication November 12, 2012; date of current version May 29, 2013. This work was supported by the National Science Council, Taiwan, under Grant NSC99-2221-E-008-088.

The authors are with the Department of Information Management, National Central University, Taoyuan 320, Taiwan (e-mail: jamesli@mgt.ncu.edu.tw; taiwei.chiang@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2012.2226890

computing methods for time-series modeling and forecasting [4], [25], [31]–[35]. In this paper, we present an innovative hybrid computing paradigm, which is denoted as CNFS-ARIMA, that integrates both complex neurofuzzy system (CNFS) and ARIMA for time-series forecasting. ARIMA models are good for linear modeling of time series, while the CNFS using CFSs is good for nonlinear function mapping. The CNFS using CFSs can achieve higher flexibility of adaption than a traditional NFS that is designed with regular type-1 fuzzy sets. The rationale of CFSs [36]–[39] has encouraged a new direction to ignite a fresh development for fuzzy systems and applications. The membership function of a type-1 fuzzy set is confined to the 1-D real-valued interval [0, 1]. In contrast, a CFS is characterized by a complex-valued membership function whose membership degrees are within the 2-D unit disc of the complex plane. Besides, CFSs are different from fuzzy complex numbers [40]–[42], whose support elements are complex-valued. The characterization of CFSs consists of an amplitude function and a phase function [36], [37], expanding the description of membership degrees from 1-D real-valued interval (for traditional type-1 fuzzy sets) to 2-D complex-valued space (for CFSs) and increasing the degrees of freedom for highly nonlinear mapping ability [43]. It is interesting that the extension by complex-valued set-element relationship with CFSs greatly enriches the state of belongingness for a set and its elements with more information for membership description to enhance the capability to catch the tie of input and output. Hence, a CNFS with CFSs may have greater adaptive capability than its NFS counterpart that uses traditional type-1 fuzzy sets. Thanks to the complex-valued property of CFSs, the proposed model can produce complex-valued output, of which the real and imaginary parts can be used for two different functional mappings, respectively. This is known as the *dual-output property*, which contrasts the CNFS with its traditional NFS counterpart whose output can handle one functional relationship only. For multivariate time-series forecasting, several approaches [44]–[47], such as NNs, radial basis function (RBF), and fuzzy-neural models, were proposed to handle time series with multiple inputs and outputs. Chakraborty *et al*. [44] presented a feed-forward connectionist network approach to multivariate time series based on past observations. Raman and Sunilkumar [45] investigated the use of artificial NN to analyze and discuss stochastic modeling of multivariate time series in the synthesis of reservoir inflow series. Nie [46] presented a fuzzy-neural approach to the prediction of nonlinear and multivariate time series. Ochoa-Rivera *et al*. [47] used an NN approach for nonlinear modeling of multivariate streamflow time series. In general, the models used in [44]–[47] inherited the flexible link-type structure of NNs and used multiple output nodes to handle multivariable time series. In these models, each output node can handle one output variable only, although multiple output nodes can be used for multivariable time series. This is completely different from the dual-output property presented in this paper that stems from the complex-valued membership functions of CFSs used for the proposed CNFS-ARIMA that can produce complex-valued output, of which the real and imaginary parts can be used for two different functional mappings, respectively. For multivariable time series, the proposed approach requires only one half of the output nodes that are by the traditional models [44]–[47]. Although the CFS theory was presented and several theoretical properties for CFSs were discussed [36]–[39], [48], [49], CFSs are still beyond human's experience [36]–[39], [48], [50]–[52], due to their complex-valued membership description. Hence, academic curiosity remains for how to design and implement an intuitively understandable complex-fuzzy-based system. Zhang *et al*. [49] discussed the δ-similarity operation properties for CFSs. Chen *et al*. [51] proposed an adaptive neurocomplex-fuzzy-inferential system (ANCFIS) for the problem of time-series forecasting, where the implementation of CFSs is based on the restricted equivalence of CFSs' phases to their support elements. Ma *et al*. [52] proposed a product–sum aggregation operator-based prediction method with CFSs for multiple periodic factor prediction problems to handle the semantic uncertainty and periodicity in the data used at the same time.

In this study, based on our previous work [43], [53]–[56], we apply a class of novel Gaussian CFSs into fuzzy IF–THEN rules of the proposed CNFS-ARIMA to study the adaptability gain for the problem of time-series forecasting. The major differences of this study and our previous research works are in the design of CFSs, the utility of ARIMA models, and the structure-learning method that is used for data-driven modeling. For the CFSs, we used a class of Gaussian CFSs whose membership degrees are more flexible by the utility of phase frequency factor λ [54] to increase their variability in the unit disc of the complex plane, which benefits to the membership description. ARIMA models are used in the consequent parts of Takagi–Sugeno fuzzy IF–THEN rules for the ability of forecasting. To construct the proposed CNFS-ARIMA properly, the modeling process is composed of two learning phases: the structure identification (or structure learning) and the parameter estimation (or parameter learning). The data-driven technique of structure learning for model structure is used to reduce human interference and, therefore, promote the efficiency of modeling, for example, in the selection of the number of fuzzy rules and the initial parameter settings of CFSs, and based on this, the phase of parameter learning follows to fine tune the model. For the phase of structure identification for the proposed computing model, a clustering method called the FCM-based splitting algorithm (FBSA) [57] is adopted to automatically settle on a proper number of clusters for a given training dataset (TD). Based on the generated clusters, fuzzy IF–THEN rules using CFSs can be created for the proposed CNFS-ARIMA. There is no fuzzy IF–THEN rule in its genesis. The FBSA clustering method is used to automatically determine the structure of the initial knowledge base that is composed of fuzzy IF–THEN rules for the proposed computing model. For the phase of parameter estimation, the well-known particle swarm optimization (PSO) algorithm and the famous recursive least squares estimator (RLSE) algorithm are integrated in a hybrid way to become the so-called PSO-RLSE method [43], [53]–[56] to fine tune the free parameters of the CNFS-ARIMA. This hybrid learning method is inspired by the divide-and-conquer concept [13], by which we separate spiritually the parameter space of the CNFS-ARIMA into two smaller spaces so that fast parameter learning for the optimal (or near optimal) solution to the proposed CNFS-ARIMA can be achieved. Originally presented by Eberhart and Kennedy [58], [59], PSO is an excellent algorithm having collective wisdom for the evolution on optimization search. The original PSO and its variants have been successfully applied in fuzzy systems and NNs in various research areas [60]–[65]. The method of RLSE [66] can do a great job on linear-model optimization problems with efficiency, and thus, it can be used for tuning the parameter of ARIMA. The performance of the hybrid PSO-RLSE learning method has been illustrated in our previous study [43], [53]–[56]. Four examples are used to test the proposed approach, whose results are compared with those by other research works. The experimental results indicate that the proposed approach shows excellent performance.

We organize the rest of this paper as follows. In Section II, the methodology of the proposed CNFS-ARIMA using CFSs is specified. In Section III, the structure identification and parameter estimation of CNFS-ARIMA are given. In Section IV, four examples for time-series forecasting are given to test the proposed approach. Experimental results are given and compared with those by other approaches. Finally, a discussion for the study is given, and the paper is concluded.

II. Methodology

A general CNFS-ARIMA computing model is composed of several fuzzy If–Then rules, where CFSs describe the antecedent parts, and ARIMA models specify the consequent parts. The membership functions of CFSs are characterized in the 2-D unit disc of the complex plane. We embed the capability of self-construction (also called self-organization or structure identification) into the proposed CNFS-ARIMA to determine adequate modeling structure automatically. This data-driven approach can reduce human intervention as much as possible and increase promising performance in data-rich applications such as time-series forecasting. Afterward, the model is fine-tuned by the so-called parameter learning (also called parameter estimation). In the following, we explain each rationale in detail.

A. Complex Fuzzy Set

The theory of CFSs [37] has pointed out a new aspect in fuzzy set theory. CFSs characterized by complex-valued membership functions can provide relatively powerful potential for learning adaptation of intelligent systems and applications. This property distinguishes CFSs from traditional type-1 fuzzy sets, whose membership grades are in the 1-D real-valued unit interval [0, 1]. Based on the complex-valued property of CFSs, the proposed CNFS-ARIMA can boost the functional mapping ability in the modeling and produce complex-valued output, whose real and imaginary parts can be used to model two different functional relationships (or two sequences of time series) simultaneously. This is called the *dual-output property*. For a CFS S, the membership function $\mu_s(h)$ can be expressed as follows:

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

where $j = \sqrt{-1}$; $h \in R$ is the base variable (or called numerical variable) in the universe of discourse U for the CFS; $r_s(h) \in [0, 1]$ is the amplitude function; $\omega_s(h) \in R$ is the phase function; and $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ indicate the real and imaginary parts, respectively. The property of the sinusoidal waves appears obviously in the definition of CFS. For the case where $\omega_s(h)$ equals 0, the CFS degenerates to a traditional type-1 fuzzy set. The amplitude term is equivalent to a regular real-valued membership function for an ordinary fuzzy set, while the phase term [37]–[39], [48] is the key feature that contrasts a CFS with an ordinary fuzzy set. Obviously, the phase function $\omega_s(h)$ that makes CFS a novel concept is a novel parameter for membership description, and it augments the set–element relationship into 2-D complex-valued space. In paired form, the CFS S can be expressed as follows:

$$S = \{((h,\ \mu_s(h))\,|h \in U\}. \qquad (2)$$

For the study, we use a class of Gaussian CFSs [54] whose membership functions, which are denoted as $c\mathrm{GMF}(h, m, \sigma, \lambda)$, are defined as follows:

$$c\mathrm{GMF}(h, m, \sigma, \lambda) = r_s(h, m, \sigma) \cdot \exp(j\omega_s(h, m, \sigma, \lambda)) \qquad (3a)$$

$$\begin{aligned}r_s(h, m, \sigma) &= \mathrm{Gaussian}(h, m, \sigma) \\ &= \exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \qquad (3b)\end{aligned}$$

$$\omega_s(h, m, \sigma, \lambda) = -\exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \cdot \left[\left(\frac{h-m}{\sigma^2}\right)\right] \cdot \lambda \qquad (3c)$$

where $\{m, \sigma, \lambda\}$ are the parameters of mean, spread, and phase frequency factor for the CFSs. The amplitude term is designed with standard Gaussian membership function. Based on the amplitude term, the phase term is devised with the first derivative of the amplitude function to make correlative relationship between them. Moreover, we use a phase frequency factor λ [54] to increase the flexibility and adaption for membership description. An illustration for a Gaussian CFS is shown in Fig. 1.

B. Autoregressive Integrated Moving Average Model

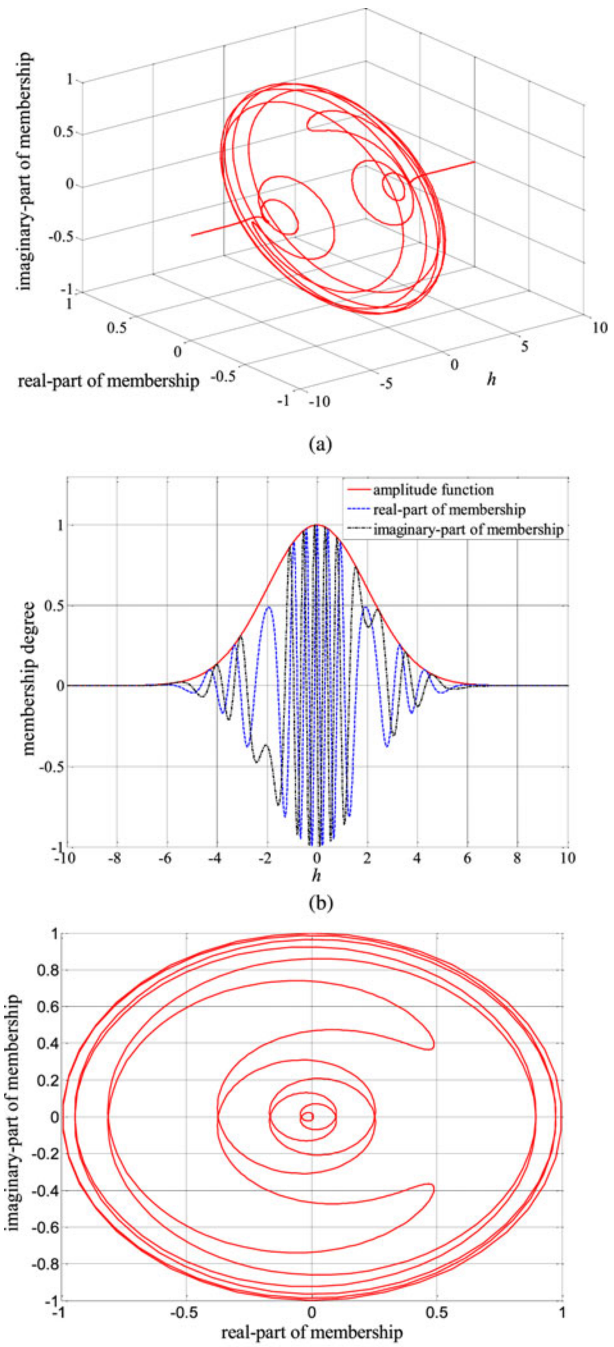
ARIMA modeling [26] is one of the most important and widely used approaches to time-series analysis and forecasting. The structure of ARIMA is supposed to be a linear functional relationship for model output, observations, and model error. For modeling, the framework of ARIMA can have the flexibility of variable structure to represent various modeling types. Suppose we have a time series, which is denoted as $\{y_\tau, \tau = 1, 2, \ldots\}$. Let us define the following notations:

$$B^k y_\tau \equiv y_{\tau-k}\ \forall k \in \{1, 2, \ldots\} \qquad (4)$$

$$\nabla^d \equiv (1-B)^d \qquad (5)$$

where τ is the time index for the series, B indicates the backward shift operator, and ∇ indicates the difference operator. The model equation of ARIMA can be expressed as follows:

$$\nabla^d y_\tau = \varphi_0 + \sum_{i=1}^{p} \varphi_i \mathrm{B}^i \nabla^d y_\tau - \sum_{j=1}^{q} \theta_j \mathrm{B}^j e_\tau + e_\tau \qquad (6)$$



(a)

(b)

(c)

Fig. 1. Gaussian CFS, where the parameters $\{m, \sigma, \lambda\}$ are set to $\{0, 2, 60\}$, and the base variable h is in the interval $[-10, 10]$. (a) Three-dimensional view in the coordinates of the base variable, real-part membership, and imaginary-part membership. (b) Real-part membership and imaginary-part membership versus base variable. (c) Real-part membership versus imaginary-part membership.

where $\{p, d, q\}$ indicate the AR order, differencing order, and MA order, respectively; $\{\varphi_i, i = 0, 1, 2, \ldots, p\}$ and $\{\theta_j, j = 1, 2, \ldots, q\}$ are model parameters; and e_τ is a random shock (model error) that is assumed to be independent and identically distributed with zero mean and a variance σ_e^2. If the differencing order $d = 0$, then an ARIMA degenerates to an ARMA. The general form of ARIMA can be expressed as follows:

$$\psi_\tau = \varphi_0 + \sum_{i=1}^{p} \varphi_i \psi_{\tau-i} - \sum_{j=1}^{q} \theta_j e_{\tau-j} + e_\tau \tag{7}$$

where $\psi_\tau \equiv \nabla^d y_\tau = (1-B)^d y_\tau$. Assume that $\{e_{\tau-j}, j = 0, 1, \ldots, q\}$ are treated as model errors in a forecasting process. Thus, a general ARIMA prediction model is given as follows:

$$\begin{aligned}\psi_\tau &= \varphi_0 + \sum_{i=1}^{p} \varphi_i B^i \psi_\tau - \sum_{j=1}^{q} \theta_j \mathrm{B}^j e_\tau \\ &\equiv \mathrm{ARIMA}(p, d, q).\end{aligned} \tag{8}$$

By introducing the complex fuzzy inference into ARIMA, we can promote the linear ARIMA modeling to become a nonlinear CNFS-ARIMA modeling. In the following, we specify the rationale of CNFS-ARIMA.

C. Complex Neurofuzzy System-Based Autoregressive Integrated Moving Average Model

Suppose we have a CNFS-ARIMA with M inputs and one output, comprising K fuzzy IF–THEN rules, where CFSs describe the antecedents (called premise parts or IF-parts), and ARIMA models specify the consequents (called THEN-parts), which are given as follows:

Rule i :

IF v_1 is ${}^i\mathrm{A}_1(h_1)$ and v_2 is ${}^i\mathrm{A}_2(h_2)$ and ...
and v_M is ${}^i\mathrm{A}_M(h_M)$

$$\text{THEN } {}^i\psi = {}^i\varphi_0 + \sum_{k=1}^{p} {}^i\varphi_k B^k \psi_\tau - \sum_{l=1}^{q} {}^i\theta_l \mathrm{B}^l e_\tau \quad \text{for } i = 1, 2, \ldots, K \tag{9}$$

where v_j is the jth input linguistic variable; ${}^i\mathrm{A}_j(h_j)$ is the CFS for the jth input linguistic variable of the ith rule for $j = 1, 2, \ldots, M$; h_j is the jth input base variable that can pick up valuation from a time series $\{y_\tau, \tau = 1, 2, \ldots\}$ and/or their variants; ${}^i\psi$ is the output of the ith rule; $\{{}^i\varphi_k, k = 0, 1, \ldots, p\}$ and $\{{}^i\theta_l, l = 1, 2, \ldots, q\}$ are the consequent parameters; $\psi_\tau = (1-B)^d y_\tau$; and e_τ is the model error. For the proposed CNFS-ARIMA, the complex fuzzy inference process can be cast into the framework of a six-layer feed-forward NN.

Layer 0: This layer is called the input layer, where each node corresponds to an input base variable. The inputs are sent to the next layer directly. Note that for the valuations of the input vector, we can pick up time-series observations and/or their variants. The input vector at time t is given as follows:

$$\mathbf{H}(t) = [h_1(t) h_2(t) \cdots h_M(t)]^{\mathrm{T}}. \tag{10}$$

Layer 1: This layer is called the complex-fuzzy-set layer, of which each node represents a linguistic value that is characterized by a CFS for which the class of Gaussian CFSs in (3a)–(3c) is used. The nodes compute the complex-valued membership degrees.

Layer 2: This layer is called the fuzzy-rule layer that is in charge of the calculation of rule firing strengths. The ith node of the layer is responsible for the firing strength ${}^{i}\beta(t)$ of the ith fuzzy rule, which is expressed as follows:

$$\begin{aligned} {}^{i}\beta(t) &= \prod_{k=1}^{M} {}^{i}\mathrm{A}_k\left(h_k(t)\right) \\ &= \prod_{k=1}^{M} {}^{i}r_k\left(h_k(t)\right)\exp\left(j\,{}^{i}\omega_k(h_k(t))\right) \end{aligned} \tag{11}$$

where the kth CFS of the ith rule is expressed as ${}^{i}\mathrm{A}_k\left(h_k(t)\right) = {}^{i}r_k\left(h_k(t)\right)\exp\left(j\,{}^{i}\omega_k(h_k(t))\right)$, for $i = 1, 2, \ldots, K$. The *fuzzy-product* operator is used for the t-norm calculation of the firing strengths.

Layer 3: This layer is for the normalization of the firing strengths. The normalized firing strength for the ith rule is expressed as follows:

$$\begin{aligned} {}^{i}\lambda(t) &= \frac{{}^{i}\beta(t)}{\sum_{i=1}^{K} {}^{i}\beta(t)} \\ &= \frac{\prod_{k=1}^{M} {}^{i}r_k\left(h_k(t)\right)\exp\left(j\,{}^{i}\omega_k(h_k(t))\right)}{\sum_{i=1}^{K}\prod_{k=1}^{M} {}^{i}r_k\left(h_k(t)\right)\exp\left(j\,{}^{i}\omega_k(h_k(t))\right)}. \end{aligned} \tag{12}$$

Layer 4: The layer is called the consequent layer, which is used to calculate the normalized consequents of the fuzzy IF–THEN rules, given as follows:

$$\begin{aligned} {}^{i}\xi(t) &= {}^{i}\lambda(t)\,{}^{i}\psi(t) \\ &= {}^{i}\lambda(t)\left({}^{i}\varphi_0 + \sum_{k=1}^{p} {}^{i}\varphi_k B^k \psi_\tau - \sum_{l=1}^{q} {}^{i}\theta_l \mathrm{B}^l e_\tau\right) \\ &\qquad i = 1, 2, \ldots, K. \end{aligned} \tag{13}$$

Layer 5: This layer is called the output layer, where the normalized consequents from Layer 4 are congregated to produce the inferred output of CNFS-ARIMA, which is given as follows:

$$\widehat{\psi}(t) = \sum_{i=1}^{K} {}^{i}\xi(t) = \sum_{i=1}^{K} {}^{i}\lambda(t)\,{}^{i}\psi(t). \tag{14}$$

In general, the CNFS-ARIMA output $\widehat{\psi}(t)$ is complex-valued, because of the complex-valued membership degrees of CFSs. We can express the output $\widehat{\psi}(t)$ as follows:

$$\begin{aligned} \widehat{\psi}(t) &= \widehat{\psi}_{\mathrm{Re}}(t) + j\widehat{\psi}_{\mathrm{Im}}(t) = |\widehat{\psi}(t)|\mathrm{exp}(j\omega_{\widehat{\psi}(t)}) \\ &= |\widehat{\psi}(t)|\mathrm{cos}(\omega_{\widehat{\psi}(t)}) + j|\widehat{\psi}(t)|\mathrm{sin}(\omega_{\widehat{\psi}(t)}) \end{aligned} \tag{15}$$

where $\widehat{\psi}_{\mathrm{Re}}(t)$ and $\widehat{\psi}_{\mathrm{Im}}(t)$ indicate the real and imaginary parts of $\widehat{\psi}(t)$, respectively. The amplitude and phase of the complex-valued output are given as follows:

$$|\widehat{\psi}(t)| = \sqrt{(\widehat{\psi}_{\mathrm{Re}}(t))^2 + (\widehat{\psi}_{\mathrm{Im}}(t))^2} \tag{16}$$

$$\omega_{\widehat{\psi}(t)} = \tan^{-1}\left(\frac{\widehat{\psi}_{\mathrm{Im}}(t)}{\widehat{\psi}_{\mathrm{Re}}(t)}\right). \tag{17}$$

Based on (15), we can regard the CNFS-ARIMA as a complex-valued function $\mathbf{F}\left(\mathbf{H}(t), \mathbf{W}\right)$, which is expressed as follows:

$$\begin{aligned} \widehat{\psi}(t) &= \mathbf{F}\left(\mathbf{H}(t), \mathbf{W}\right) \\ &= \mathbf{F}_{\mathrm{Re}}\left(\mathbf{H}(t), \mathbf{W}\right) + j\mathbf{F}_{\mathrm{Im}}\left(\mathbf{H}(t), \mathbf{W}\right) \end{aligned} \tag{18}$$

where $\mathbf{F}_{\mathrm{Re}}(\cdot)$ and $\mathbf{F}_{\mathrm{Im}}(\cdot)$ are the real and imaginary parts of $\mathbf{F}(\cdot)$, respectively; $\mathbf{H}(t)$ is the input vector to the CNFS-ARIMA; and $\mathbf{W}$ denotes the parameter set of the CNFS-ARIMA, which is composed of the subset of IF-part parameters (which is denoted as $\mathbf{W}_{\mathrm{If}}$) and the subset of THEN-part parameters (denoted as $\mathbf{W}_{\mathrm{Then}}$). Thus, the parameter set $\mathbf{W}$ is expressed as follows:

$$\mathbf{W} = \mathbf{W}_{\mathrm{If}} \cup \mathbf{W}_{\mathrm{Then}}. \tag{19}$$

To determine the initial structure of knowledge base of the CNFS-ARIMA, we use a fuzzy c-means (FCM)-based clustering method called the FCM-based splitting algorithm (FBSA) [57], by which an appropriate number of IF–THEN rules [viz., K in (9)] for the CNFS-ARIMA can be determined automatically. This process is called the structure learning (or structure identification) phase. To fine tune the CNFS-ARIMA, the parameter set $\mathbf{W}$ can be updated iteratively by a machine-learning-based method, such as PSO [58], [59] and RLSE [66]. This process is called the parameter learning (or parameter estimation) phase. In this study, we use a hybrid learning method, which is called the PSO-RLSE method, to fine tune the $\mathbf{W}_{\mathrm{If}}$ and $\mathbf{W}_{\mathrm{Then}}$ for fast learning purpose. This will become clear in the following section.

III. MACHINE LEARNING FOR THE PROPOSED COMPLEX NEUROFUZZY SYSTEM AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODEL

A. Structure Learning

The purpose of structure learning is to partition the input space appropriately according to the distribution of training data so that the initial knowledge base of the proposed CNFS-ARIMA can be established. The amount of fuzzy IF–THEN rules of a neural fuzzy model can affect application performance. In general, fuzzy rules can provide linguistic interpretation capabilities, which are fundamentally important to catch imprecision, as well as vagueness, and to represent subjective knowledge for reasoning. An appropriate amount of fuzzy IF–THEN rules for a CNFS-ARIMA model can enhance such abilities to deal with inexact information or uncertainty. In addition, the computational overhead of parameter learning is usually high if an excessive amount of fuzzy rules are involved, due to the increase of free parameters. Hence, we adopt the FBSA method [57] to automatically settle on a proper amount of clusters, on which fuzzy IF–THEN rules in (9) can be generated [16], [17]. The FBSA is a clustering algorithm that is based on the famous FCM [67], [68] and a validity function. For FCM, the fact that the number of clusters needs to give beforehand is a kind of disadvantage. The FBSA overcomes such a drawback by incrementally giving the number of clusters to FCM from small to large and then use the

Step 1. Preset C_{min} and C_{max} between which an optimal number is to be decided.
Let the cluster number be $c = C_{min}$.
Initialize the cluster centers, $\mathbf{V}(0)$.
Step 2. Apply FCM to the dataset to update the set of membership degrees in $\mathbf{U}$ and the cluster centers in $\mathbf{V}$.
Step 3. Compute *Scat*(c) in (21) and *Sep*(c) in (22).
Step 4. If $c < C_{max}$, then {compute a score *Score*(j) in (28) for each cluster, j=1,2,…,c; split the cluster with the worst score; c=c+1; go back to **Step 2**.}
Step 5. Compute validity index $V_d(\mathbf{U},\mathbf{V},c)$ in (20). Choose the number of clusters, c_f, whose $V_d(\mathbf{U},\mathbf{V},c_f)$ is the best.
Step 6. Stop.

Fig. 2. FBSA clustering algorithm.

validity function to discriminate the clustering results by FCM, and based on this, we can choose the best. Then, according to the rationale of self-organization for neural fuzzy system [16], [17], for each partitioned area (by a cluster) in the input space, we can generate a fuzzy IF–THEN rule. Suppose we have a dataset $\mathbf{X} = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$, where $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \cdots\ x_{i,M}]^{\mathrm{T}}$ is the ith datum with M dimensions. We like to divide the dataset into c clusters, whose centers are denoted by $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$, where $\mathbf{v}_i = [v_{i,1}\ v_{i,2}\ \cdots\ v_{i,M}]^{\mathrm{T}}$ is the center vector of the ith cluster. The set of membership degrees by FCM is denoted by $\mathbf{U} = \{u_{ij}, i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, c\}$, where u_{ij} is the membership degree for datum $\mathbf{x}_i$ to cluster $\mathbf{v}_j$. The FBSA algorithm is given in Fig. 2.

In Fig. 2, the FCM that is used in Step2 can be found in [68], and the *Scat*(c), *Sep*(c), and *Score*(i) are explained in the following.

For each number c from $C_{\min}$ to $C_{\max}$, the FBSA is performed to determine an appropriate number of clusters, based on the validity function given as follows:

$$V_d(\mathbf{U}, \mathbf{V}, c) = \mathit{Scat}(c) + \frac{\mathit{Sep}(c)}{\mathit{Sep}(c_{\max})} \tag{20}$$

where

$$\mathit{Scat}(c) = \frac{\frac{1}{c}\sum_{i=1}^{c} \|\sigma(\mathbf{v}_i)\|}{\|\sigma(\mathbf{X})\|} \tag{21}$$

$$\mathit{Sep}(c) = \frac{\min_{i\neq j} \|\mathbf{v}_i - \mathbf{v}_j\|^2}{\max_{i\neq j} \|\mathbf{v}_i - \mathbf{v}_j\|^2} \sum_{i=1}^{c} \left(\sum_{j=1}^{c} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \right)^{-1} \tag{22}$$

$$\boldsymbol{\sigma}(\mathbf{X}) = [\sigma(\mathbf{X})_1\ \sigma(\mathbf{X})_2\ \cdots\ \sigma(\mathbf{X})_M\ ]^{\mathrm{T}} \tag{23}$$

$$\sigma(\mathbf{X})_p = \frac{1}{n}\sum_{i=1}^{n} (x_{i,p} - \bar{x}_p) \tag{24}$$

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \tag{25}$$

$$\boldsymbol{\sigma}(\mathbf{v}_i) = [\sigma(\mathbf{v}_i)_1\ \sigma(\mathbf{v}_i)_2\ \cdots\ \sigma(\mathbf{v}_i)_M]^{\mathrm{T}} \tag{26}$$

$$\sigma(\mathbf{v}_j)_p = \frac{1}{n}\sum_{i=1}^{n} u_{ij}(x_{i,p} - v_{j,p}). \tag{27}$$

Note that Scat(c) decreases as the number of clusters increases and the clusters get more compact and that Sep(c) tends to be small when the geometry of the cluster centers is well distributed. By (20), it is possible to decide an appropriate number of clusters. In Step 4 of Fig. 2, the worst cluster among the c clusters generated so far is selected to divide into two new clusters, based on the score function given as follows:

$$\mathit{Score}(j) = \frac{\sum_{i=1}^{n} u_{ij}}{\text{cardinality of cluster } j} \text{ for } j = 1, 2, \ldots, c \tag{28}$$

where the cardinality is the number of data in cluster j, i.e., all the $\boldsymbol{x}_i$ such that $u_{ij} > u_{ik}$ for all $j \neq k$ and $k = 1, 2, \ldots, c$. Therefore, through the FBSA, the optimal number (c_f) of clusters is determined, by which the dataset can be well partitioned into the c_f clusters whose centers and spreads in standard deviation can be used as the parameters of mean and spread for Gaussian CFSs given in (3a)–(3c). The valuation of c_f is then assigned to be the amount of fuzzy rules for the CNFS-ARIMA. In the following, we continue to specify the PSO-RLSE method for parameter learning.

B. Parameter Learning

With the concept of divide-and-conquer [13], we spiritually view the parameter set $\mathbf{W}$ of CNFS-ARIMA as the union of two subsets: the subset of IF-part parameters ($\mathbf{W}_{\mathrm{If}}$) and the subset of THEN-part parameters ($\mathbf{W}_{\mathrm{Then}}$). The model of CNFS-ARIMA is fine-tuned by the PSO-RLSE hybrid method [53]–[56], where the PSO evolves $\mathbf{W}_{\mathrm{If}}$ and the RLSE updates $\mathbf{W}_{\mathrm{Then}}$. A PSO swarm is composed of many particles, each of which is viewed as a bird (or a fish) seeking for food. Assume the problem space is with Q dimensions. The update equations of PSO are expressed as follows:

$$\boldsymbol{\vartheta}_i(k+1) = w\boldsymbol{\vartheta}_i(k) + c_1\rho_1(\mathbf{pbest}_i(k) - \mathbf{L}_i(k)) + c_2\rho_2(\mathbf{gbest}(k) - \mathbf{L}_i(k)) \tag{29}$$

$$\mathbf{L}_i(k+1) = \mathbf{L}_i(k) + \boldsymbol{\vartheta}_i(k+1), \text{ for } i = 1, 2, \ldots, s_{\text{swarm}} \tag{30}$$

where i indicates the ith particle; s_{swarm} is the swarm size; k indicates the kth iteration for PSO; $\mathbf{pbest}_i(k)$ is the best location of the ith particle; $\mathbf{gbest}(k)$ is the best location of the swarm; $\boldsymbol{\vartheta}_i(k) = [\vartheta_{i,1}(k),\ \vartheta_{i,2}(k), \ldots, \vartheta_{i,Q}(k)]^T$ is the velocity of the ith particle whose location is $\mathbf{L}_i(k) = [l_{i,1}(k),\ l_{i,2}(k), \ldots, l_{i,Q}(k)]^T$; $\{w, c_1, c_2\}$ are the parameters of PSO; and $\{\rho_1, \rho_2\}$ are random numbers in [0,1]. For a general least squares estimation problem, the LSE model is specified by a linearly parameterized expression, which is given as follows:

$$d = \theta_1 f_1(u) + \theta_2 f_2(u) + \cdots + \theta_m f_m(u) + \varepsilon \tag{31}$$

where u is the input to model; d is the target; $\{f_i(u),\ i = 1, 2, \ldots, m\}$ are known functions of u; $\{\theta_i,\ i = 1, 2, \ldots, m\}$ are the model parameters to be estimated; and ε is the model error. Note that $\{\theta_i, i = 1, 2, \ldots, m\}$ can be viewed as THEN-part parameters of the proposed CNFS-ARIMA. Suppose we have a TD with n pairs denoted as follows:

$$\mathrm{TD} = \{(u_i, d_i), i = 1, 2, \ldots, n\} \tag{32}$$

where (u_i, d_i) is the ith data pair that is arranged in the form of (*input, target*). Substituting the data pairs into (31), we have a set of n equations in matrix notation

$$\mathbf{d} = \mathbf{B}\theta + \boldsymbol{\varepsilon} \tag{33a}$$

where

$$\mathbf{B} = \begin{bmatrix} f_1(u_1) & f_2(u_1) & \ldots & f_m(u_1) \\ f_1(u_2) & f_2(u_2) & \ldots & f_m(u_2) \\ \vdots & \vdots & \ldots & \vdots \\ f_1(u_n) & f_2(u_n) & \ldots & f_m(u_n) \end{bmatrix} \tag{33b}$$

$$\theta = [\,\theta_1 \quad \theta_2 \quad \cdots \quad \theta_m\,]^{\mathrm{T}} \tag{33c}$$

$$\mathbf{d} = [\,d_1 \quad d_2 \quad \cdots \quad d_n\,]^{\mathrm{T}} \tag{33d}$$

$$\boldsymbol{\varepsilon} = [\,\varepsilon_1 \quad \varepsilon_2 \quad \cdots \quad \varepsilon_n\,]^{\mathrm{T}}. \tag{33e}$$

The optimal estimate for $\boldsymbol{\theta}$ can be obtained with the following RLSE equations [66]:

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P_k b_{k+1} (b_{k+1})^{\mathrm{T}} P_k}}{1 + (\mathbf{b}_{k+1})^{\mathrm{T}} \mathbf{P}_k \mathbf{b}_{k+1}} \tag{34a}$$

$$\theta_{k+1} = \theta_k + \mathbf{P}_{k+1}\mathbf{b}_{k+1}(d_{k+1} - (\mathbf{b}_{k+1})^{\mathrm{T}}\theta_k) \tag{34b}$$

for $k = 0, 1, \ldots, (n-1)$, where [$\mathbf{b}_k^T$, d_k] is the kth row of [**B**, **d**]. To start the RLSE algorithm, we set initial θ_0 to zero and $\mathbf{P}_0 = \alpha\mathbf{I}$, where α is a large positive value, and **I** is the identity matrix. We define the cost function in the sense of mean square error (MSE) as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} e(t)(e(t))^* \tag{35}$$

where * indicates complex conjugate; $e(t) = d(t) - \hat{y}(t)$, where $d(t)$ is the target, and $\hat{y}(t)$ is the forecast by CNFS-ARIMA. Note that the root mean square error (RMSE) can be obtained easily by taking RMSE $=\sqrt{\mathrm{MSE}}$. The procedure of the PSO-RLSE hybrid method is given in the following.

Step 1: Start PSO for IF-part parameters ($\mathbf{W}_{\mathrm{If}}$).

Step 2: Calculate the firing strengths of fuzzy rules, $\{{}^i\lambda, i = 1, 2, \ldots, K\}$.

Step 3: Update THEN-part parameters ($\mathbf{W}_{\mathrm{Then}}$) by RLSE. For the implementation of PSO-RLSE, the vectors $\mathbf{b}$ and θ for RLSE are arranged as follows:

$$\mathbf{b}_{k+1} = [{}^1\mathbf{bb}\,(k+1) \quad {}^2\mathbf{bb}\,(k+1) \cdots {}^K\mathbf{bb}\,(k+1)] \tag{36}$$

$${}^i\mathbf{bb}\,(k+1) = [{}^i\lambda \quad {}^i\lambda h_1\,(k+1) \cdots {}^i\lambda h_M\,(k+1)] \tag{37}$$

$$\theta_k = [{}^1\boldsymbol{\tau}_k \quad {}^2\boldsymbol{\tau}_k \cdots {}^K\boldsymbol{\tau}_k] \tag{38}$$

$${}^i\boldsymbol{\tau}_k = [{}^i a_0\,(k) \quad {}^i a_1\,(k) \cdots {}^i a_M\,(k)] \tag{39}$$

for $i = 1, 2, \ldots, K$ and $k = 0, 1, \ldots, (n-1)$. Note that θ_k is updated recursively and sequentially by the training data pairs and that θ_n updated by the last data pair is used for THEN-part parameters.

Step 4: With TD, obtain forecasts by CNFS-ARIMA; calculate model errors and cost.

Step 5: For all the PSO particles, calculate their costs by repeating Steps 2–4.

Step 6: Update **pbest** for every particle and **gbest** for the PSO swarm. Update $\mathbf{W}_{\mathrm{If}}$ by **gbest**.

Step 7: Check termination condition to stop; otherwise, go back to *Step 2* and continue the procedure.

IV. EXPERIMENTATION

In order to clearly represent the structural order of the proposed model, we introduce the notation of CNFS(K)-ARIMA (p, d, q), meaning that the CNFS-ARIMA model has K fuzzy IF–THEN rules whose consequents have the form of ARIMA (p, d, q), where (p, d, q) indicates the order of ARIMA. The fuzzy IF–THEN rules in (9) can be rewritten as follows:

$$\text{Rule } i\text{: IF } x_1 \text{ is } {}^i\mathrm{A}_1\,(h_1) \text{ and } x_2 \text{ is } {}^i\mathrm{A}_2\,(h_2) \text{ and } \ldots \text{and } x_M \text{ is } {}^i\mathrm{A}_M\,(h_M)$$
$$\text{THEN } {}^i\psi = {}^i\mathrm{ARIMA}(p, d, q), \text{for } i = 1, 2, \ldots, K \tag{40}$$

where K can be determined automatically by the FBSA clustering method for structure learning that is done once only on the TD; ${}^i\mathrm{ARIMA}(p, d, q)$ is the ARIMA model with the order of (p, d, q) for the ith fuzzy rule. The parameter estimation is performed by the PSO-RLSE method iteratively.

A. Example 1—Star Brightness Time Series

The sequence of the time series is composed of the daily brightness observations at midnight of a variable star for 600 successive days. The time series is scaled by the factor of 1/30. In this example, the time series is normalized into the range [0, 1] and denoted as $\{y(t), t = 1, 2, \ldots, 600\}$, where t is the generic time index. The data of the time series are arranged in the form of (*input, target*) for 597 data pairs, which are denoted as $\{(\mathbf{H}(i), d(i)), i = 1, 2, \ldots, 597\}$

$$\mathbf{H}(i) = [y(t-2),\ y(t-1),\ y(t)]^{\mathrm{T}} \tag{41}$$

$$d(i) = y(t+1) \tag{42}$$

where $t = i + 2$; $\mathbf{H}(i)$ is the input vector to the proposed model from the ith pair, and $d(i)$ is the corresponding target. The method of lags is used for input vector. The size of training data pairs is set to 300. For structure learning, the FBSA clustering method is applied to the TD to determine an appropriate number of fuzzy rules for the CNFS-ARIMA prototype, and based on this, several CNFS-ARIMA models with different orders are to be created by parameter learning. Structure learning by the FBSA is done once only on the TD. The settings of the FBSA method are given in Table I.

The results by the FBSA are shown in Fig. 3 and Table II, 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 that have three inputs and one output. Each input has four CFSs. For parameter learning, the cost function in MSE is used, where only the real part of the proposed model output is involved, due to the consideration that the observations of the time series are real-valued. The settings

TABLE I
SETTINGS OF THE FBSA CLUSTERING METHOD

Parameter	Value	Remark
C_{min}	2	Min number of clusters
C_{max}	10	Max number of clusters
m	2	FCM fuzzifier
ε_{FCM}	10^{-9}	Threshold of termination for FCM

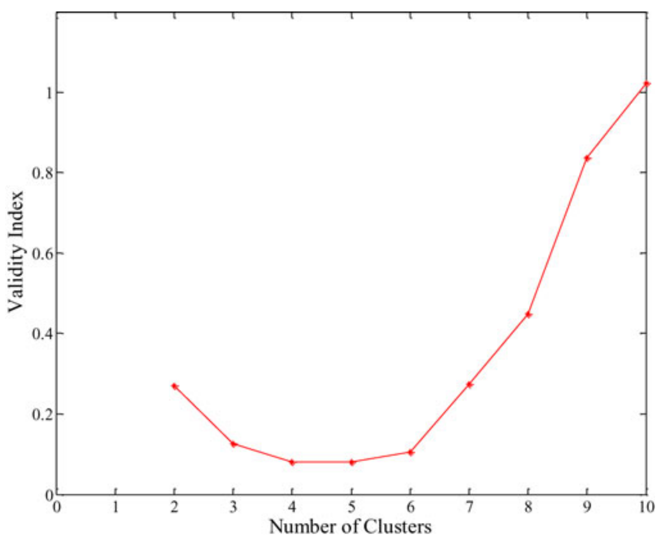


Fig. 3. Validity index curve of clusters. (Star brightness time series.)

TABLE II
VALIDITY INDICES FOR DIFFERENT NUMBERS OF CLUSTERS (STAR BRIGHTNESS TIME SERIES)

Number of clusters	Validity index
2	0.2692
3	0.1256
4	**0.0762**
5	0.0808
6	0.1060
7	0.2729
8	0.4477
9	0.8362
10	1.0223

of the PSO-RLSE method are given in Table III. With Gaussian CFSs given in (3a)–(3c), the antecedents of the fuzzy IF–THEN rules of the proposed models are described.

With the PSO-RLSE leaning method, these CNFS-ARIMA models (shown in Table III) are trained, each of which is trained for 20 repeated trials. The prediction models are compared with other approaches [5], [69], where the first 300 samples were used for the training data, and the reaming samples were used for testing. Nondimensional error index (NDEI) [51], which is also called the normalized root mean square error (NRMSE) [70], is defined as the RMSE divided by the standard deviation of a

TABLE III
SETTINGS OF THE PSO-RLSE METHOD (STAR BRIGHTNESS TIME SERIES)

Prediction model	CNFS(4)-ARIMA(3,0,0) CNFS(4)-ARIMA(3,1,0) CNFS(4)-ARIMA(3,2,0)	CNFS(4)-ARIMA(3,0,1) CNFS(4)-ARIMA(3,1,1) CNFS(4)-ARIMA(3,2,1)
PSO		
Dimensions of PSO particle	36	
Swarm size	200	
Initialization of particle positions	Random in $[0, 1]^{36}$	
Initialization of particle velocities	Random in $[0, 1]^{36}$	
Learning rate (c_1,c_2)	(2,2)	
Inertia weight w	0.9	
Max training iterations	200	
RLSE		
Number of consequent-part parameters	16	20
$\boldsymbol{\theta}_0$	16×1 zero vector	20×1 zero vector
$\mathbf{P}_0$	$\alpha\mathbf{I}$	
α	10^8	
$\mathbf{I}$	16×16 identity matrix	20×20 identity matrix

target time series, which is given as follows:

$$\mathrm{NDEI} = \frac{\sqrt{\mathrm{MSE}}}{\sigma} = \frac{\sqrt{\left(\sum_{t=1}^{n}\left(d(t) - \mathrm{Re}(\xi(t))\right)^2 / n\right)}}{\sigma} \tag{43}$$

where $d(t)$ and $\mathrm{Re}(\xi(t))$ are the target and forecast by CNFS-ARIMA, respectively; and σ is the standard deviation of the time series. For performance comparison, we summarize the results in Table IV. The CNFS(4)-ARIMA(3,0,0) is compared with its counterpart NFS(4)-ARIMA(3,0,0), whose antecedents are designed with traditional Gaussian fuzzy sets in (3b). A learning curve for the proposed CNFS(4)-ARIMA(3,2,0) is shown in Fig. 4. For one of the 20 trials, the response by the proposed CNFS(4)-ARIMA(3,2,0) is shown in Fig. 5.

In Table IV, the proposed approach has shown much better performance than the compared approaches in terms of mean and standard deviation of MSE. Among the proposed prediction models, the CNFS(4)-ARIMA(3,2,0) has the best performance, whose prediction MSE is 1.74×10^{-4} for the training phase, and 1.94×10^{-4} for the testing phase. For the comparison of the CNFS(4)-ARIMA(3,0,0) and the NFS(4)-ARIMA(3,0,0), the former is better than the latter in the testing phase, in terms of accuracy and much smaller standard deviation. The experimental results indicate that the proposed models show excellent performance, in terms of performance accuracy and small standard deviation.

For further performance comparison, the proposed CNFS(4)-ARIMA(3,2,0) is compared with ANCFIS [51] that used the first 480 training samples for training and the remaining 120 samples for testing. On the same basis of training and testing samples, the CNFS(4)-ARIMA(3,2,0) is trained and tested, whose results

TABLE IV
PERFORMANCE COMPARISON (20 REPEATED TRIALS) (STAR BRIGHTNESS TIME SERIES)

Method	Training phase		Testing phase		# of rules
	MSE (std)	NDEI (std)	MSE (std)	NDEI (std)	
TSK-NFIS [69]	3.14×10^{-4} (-)	5.90×10^{-2} (-)	3.31×10^{-4} (-)	6.09×10^{-2} (-)	-
AR [69]	3.14×10^{-4} (-)	5.81×10^{-2} (-)	3.22×10^{-4} (-)	6.01×10^{-2} (-)	-
NAR [69]	3.20×10^{-4} (-)	5.96×10^{-2} (-)	3.12×10^{-4} (-)	5.92×10^{-2} (-)	-
Neural Net [69]	3.01×10^{-4} (-)	5.78×10^{-2} (-)	3.11×10^{-4} (-)	5.91×10^{-2} (-)	-
ANN-ARMA [5]	-	-	0.7534 (-)	-	-
NFS(4)- ARIMA(3,0,0)	1.97×10^{-4} (4.25×10^{-6})	4.68×10^{-2} (1.01×10^{-3})	3.03×10^{-4} (3.18×10^{-5})	5.83×10^{-2} (6.12×10^{-3})	4
CNFS(4)-ARIMA(3,0,0)	1.93×10^{-4} (7.57×10^{-7})	4.63×10^{-2} (1.81×10^{-4})	2.17×10^{-4} (3.15×10^{-6})	4.94×10^{-2} (7.17×10^{-4})	4
CNFS(4)-ARIMA(3,0,1)	1.82×10^{-4} (2.39×10^{-6})	4.50×10^{-2} (5.91×10^{-4})	2.00×10^{-4} (9.21×10^{-6})	4.74×10^{-2} (2.19×10^{-3})	4
CNFS(4)-ARIMA(3,1,0)	1.93×10^{-4} (7.49×10^{-7})	4.63×10^{-2} (1.80×10^{-4})	2.17×10^{-4} (1.15×10^{-6})	4.95×10^{-2} (2.60×10^{-4})	4
CNFS(4)-ARIMA(3,1,1)	1.87×10^{-4} (1.89×10^{-7})	4.56×10^{-2} (5.75×10^{-4})	2.50×10^{-4} (1.66×10^{-6})	5.31×10^{-2} (3.53×10^{-4})	4
CNFS(4)-ARIMA(3,2,0)	**1.74×10^{-4} (1.89×10^{-7})**	**4.39**$\times10^{-2}$ (4.77×10^{-6})	**1.94×10^{-4} (1.66×10^{-6})**	**4.67×10^{-2} (3.99×10^{-4})**	**4**
CNFS(4)-ARIMA(3,2,1)	1.77×10^{-4} (5.16×10^{-7})	4.43×10^{-2} (1.29×10^{-5})	1.96×10^{-4} (4.85×10^{-6})	4.70×10^{-2} (1.16×10^{-3})	4

Note that the results by these models are on the basis that the first 300 samples were used for training and the remaining 300 samples were used for testing.

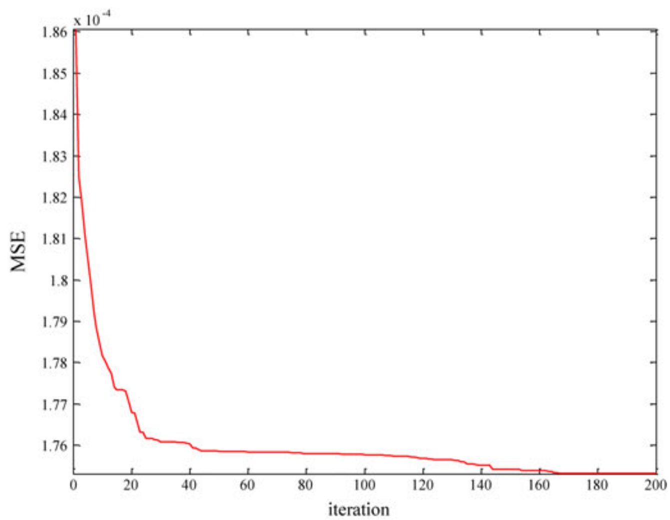


Fig. 4. Learning curve of CNFS(4)-ARIMA(3,2,0). (Star brightness time series.)

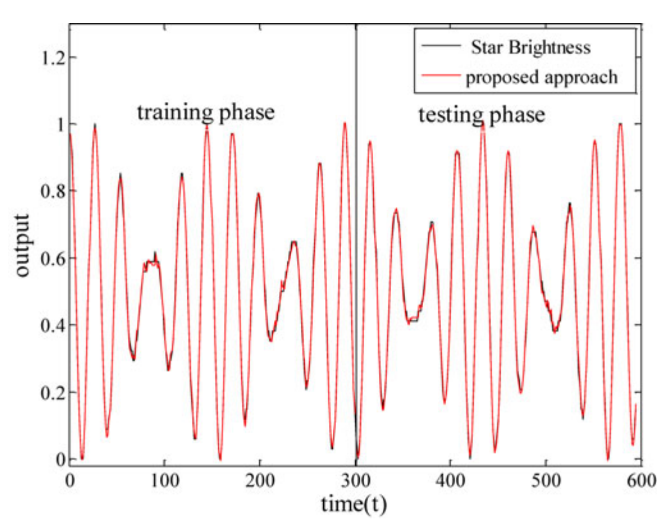


Fig. 5. Prediction response by CNFS(4)-ARIMA(3,2,0). (Star brightness time series.)

TABLE V
PERFORMANCE COMPARISON IN MSE AND NDEI
(STAR BRIGHTNESS TIME SERIES)

Method	MSE	NDEI	# of rules
RBF	0.0025	0.1289	-
SVR	2.50×10^{-4}	0.0129	-
ANFIS	2.51×10^{-4}	0.0130	4
ANCFIS [51]	5.61×10^{-5}	0.0029	-
CNFS(4)-ARIMA(3,2,0) (proposed)	3.64×10^{-5}	0.0023	4

Note that the results by these models are on the basis that the first 480 samples were used for training and the remaining 120 samples were used for testing.

are also compared with those by the approaches of adaptive neurofuzzy inference system (ANFIS) [13], RBF network [71], and support vector regression (SVR) [72], [73]. We implemented ANFIS and RBF with the fuzzy logic toolbox and NN toolbox of MATLAB, respectively, and SVR with the LIBSVM toolbox [74]. The performance comparison in MSE and NDEI are given in Table V, where the proposed approach outperforms the compared approaches. The proposed CNFS(4)-ARIMA(3,2,0) shows much better performance in Table V than those in Table IV, because several more samples were used in training, through which more information was involved for the model to explore the underlying relationship and increase the forecasting performance. In Table V, the proposed CNFS-ARIMA is superior to the compared ANCFIS, ANFIS, RBF, and SVR.

In order to see performance effects by the structure- and parameter-learning methods, we did several more experiments, on which different combinations were tested both with and without FBSA, RLSE, and PSO-RLSE. Moreover, to test performance effects by a different clustering method, we also did several more experiments on which the Gustafson–Kessel FCM (G-KFCM) method [75] was used to replace the FBSA clustering method for structure learning, after which, the PSO-RLSE method was still used for parameter learning. The G-KFCM is the method that extends the standard FCM algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes on a dataset. The results of these experiments are summarized in Table VI.

B. Example 2—Mackey–Glass Chaos Time Series

The well-known Mackey–Glass chaotic time series, which is governed by the following differential equation [76], is frequently used as a benchmark for research:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{44}$$

where $\tau = 17$; $x(0) = 1.2$; and $x(t) = 0$ for $t < 0$. The time step is given as 0.1 s. The prediction form by the proposed approach is given as follows:

$$x(t+\bar{C}) = f(x(t), x(t-\Delta), \ldots, x(t-(\bar{D}-1)\Delta)) \tag{45}$$

where t is the generic time index ($t = 118$–1117); $\bar{C} = \Delta = 6$; and $\bar{D} = 4$. Normalized to the range of the unit interval [0, 1], the 1024 data of the time series are denoted as $\{x(i),\ i = 100, 101, \ldots, 1123\}$. The data are arranged in the form of (*input, target*) for 1000 data pairs, which are denoted as $\{(\mathbf{H}(k), d(k)),\ k = 1, 2, \ldots, 1000\}$

$$\mathbf{H}(k) = [x(i-18),\ x(i-12),\ x(i-6),\ x(i)]^{\mathrm{T}} \tag{46}$$

$$d(k) = x(i+6) \tag{47}$$

where $i = k + 117$; and the lags and lag temperature are selected the same as those by ANFIS [13]. The first 500 data pairs are used for training, and the remaining data pairs are used for testing. First, for structure learning, we apply the FBSA clustering method to the training set for a CNFS-ARIMA prototype. The settings of FBSA are given in Table I, except for $C_{\max} = 20$. Eight clusters are determined to be the best. For parameter learning, the cost function is designed with RLSE, and the settings of the PSO-RLSE method are given in Table VII. Gaussian CFSs are used for the antecedents of the proposed models. Real-part model outputs are used. For performance comparison, four performance indices, i.e., MSE, RMSE, NDEI, and normalized MSE (NMSE), are used. The definition of NMSE is given as follows:

$$\mathrm{NMSE} = \frac{\sum_{t=1}^{n}\left(d(t) - \mathrm{Re}(\xi(t))\right)^2}{\sigma^2 n} \tag{48}$$

where σ is the standard deviation of the time series. Based on the model prototype obtained by structure learning, several models of CNFS(8)-ARIMA(p, d, q) with different orders are trained by parameter learning, whose results are compared with those by other approaches [13], [32], [51], [69], [77]–[84]. The learning curve is shown in Fig. 6 for the proposed CNFS(8)-ARIMA(4,1,0), whose prediction response is given in Fig. 7. The performance comparison is given in Table VIII, where the proposed CNFS(8)-ARIMA(4,1,0) shows much better performance than the compared approaches and comparable with the ANCFIS [51], which also used the concept of CFSs. This again shows that CFSs can provide excellent adaptive capability for modeling and forecasting.

TABLE VI
PERFORMANCE COMPARISON FOR DIFFERENT COMBINATIONS WITH AND WITHOUT FBSA, G-KFCM, RLSE, AND PSO-RLSE (STAR BRIGHTNESS TIME SERIES)

Method	Structure learning	Parameter learning	MSE
NFS(4)-ARIMA(3,0,0)	FBSA	RLSE	6.51×10^{-2}
NFS(4)-ARIMA(3,0,1)	FBSA	RLSE	7.05×10^{-2}
NFS(4)-ARIMA(3,1,0)	FBSA	RLSE	6.08×10^{-2}
NFS(4)-ARIMA(3,1,1)	FBSA	RLSE	4.81×10^{-2}
NFS(4)-ARIMA(3,2,0)	FBSA	RLSE	4.53×10^{-2}
NFS(4)-ARIMA(3,2,1)	FBSA	RLSE	5.20×10^{-2}
NFS(4)-ARIMA(3,0,0)	-	PSO-RLSE	3.18×10^{-4}
NFS(4)-ARIMA(3,0,1)	-	PSO-RLSE	3.11×10^{-4}
NFS(4)-ARIMA(3,1,0)	-	PSO-RLSE	3.15×10^{-4}
NFS(4)-ARIMA(3,1,1)	-	PSO-RLSE	3.13×10^{-4}
NFS(4)-ARIMA(3,2,0)	-	PSO-RLSE	2.87×10^{-4}
NFS(4)-ARIMA(3,2,1)	-	PSO-RLSE	3.00×10^{-4}
NFS(4)-ARIMA(3,0,0)	FBSA	PSO-RLSE	3.01×10^{-4}
NFS(4)-ARIMA(3,0,1)	FBSA	PSO-RLSE	2.79×10^{-4}
NFS(4)-ARIMA(3,1,0)	FBSA	PSO-RLSE	2.95×10^{-4}
NFS(4)-ARIMA(3,1,1)	FBSA	PSO-RLSE	2.81×10^{-4}
NFS(4)-ARIMA(3,2,0)	**FBSA**	**PSO-RLSE**	$\mathbf{2.67\times10^{-4}}$
NFS(4)-ARIMA(3,2,1)	FBSA	PSO-RLSE	2.76×10^{-4}
CNFS(4)-ARIMA(3,0,0)	FBSA	RLSE	6.51×10^{-2}
CNFS(4)-ARIMA(3,0,1)	FBSA	RLSE	6.88×10^{-2}
CNFS(4)-ARIMA(3,1,0)	FBSA	RLSE	5.86×10^{-2}
CNFS(4)-ARIMA(3,1,1)	FBSA	RLSE	4.05×10^{-2}
CNFS(4)-ARIMA(3,2,0)	FBSA	RLSE	3.75×10^{-2}
CNFS(4)-ARIMA(3,2,1)	FBSA	RLSE	4.72×10^{-2}
CNFS(4)-ARIMA(3,0,0)	-	PSO-RLSE	2.64×10^{-4}
CNFS(4)-ARIMA(3,0,1)	-	PSO-RLSE	2.77×10^{-4}
CNFS(4)-ARIMA(3,1,0)	-	PSO-RLSE	2.56×10^{-4}
CNFS(4)-ARIMA(3,1,1)	-	PSO-RLSE	2.91×10^{-4}
CNFS(4)-ARIMA(3,2,0)	-	PSO-RLSE	2.48×10^{-4}
CNFS(4)-ARIMA(3,2,1)	-	PSO-RLSE	2.58×10^{-4}
CNFS(4)-ARIMA(3,0,0)	G-KFCM	PSO-RLSE	2.03×10^{-4}
CNFS(4)-ARIMA(3,0,1)	G-KFCM	PSO-RLSE	2.12×10^{-4}
CNFS(4)-ARIMA(3,1,0)	G-KFCM	PSO-RLSE	2.07×10^{-4}
CNFS(4)-ARIMA(3,1,1)	G-KFCM	PSO-RLSE	2.36×10^{-4}
CNFS(4)-ARIMA(3,2,0)	G-KFCM	PSO-RLSE	1.87×10^{-4}
CNFS(4)-ARIMA(3,2,1)	G-KFCM	PSO-RLSE	1.88×10^{-4}
CNFS(4)-ARIMA(3,0,0)	FBSA	PSO-RLSE	2.10×10^{-4}
CNFS(4)-ARIMA(3,0,1)	FBSA	PSO-RLSE	1.98×10^{-4}
CNFS(4)-ARIMA(3,1,0)	FBSA	PSO-RLSE	2.10×10^{-4}
CNFS(4)-ARIMA(3,1,1)	FBSA	PSO-RLSE	2.38×10^{-4}
CNFS(4)-ARIMA(3,2,0)	**FBSA**	**PSO-RLSE**	$\mathbf{1.81\times10^{-4}}$
CNFS(4)-ARIMA(3,2,1)	FBSA	PSO-RLSE	1.91×10^{-4}

Note that the results by these models are on the basis that the first 300 samples were used for training and the remaining 300 samples were used for testing.

TABLE VII
SETTINGS OF THE PSO-RLSE METHOD (MACKEY–GLASS CHAOTIC TIME SERIES)

Prediction model	CNFS(8)-ARIMA(4,0,0) CNFS(8)-ARIMA(4,1,0) CNFS(8)-ARIMA(4,2,0)	CNFS(8)-ARIMA(4,0,1) CNFS(8)-ARIMA(4,1,1) CNFS(8)-ARIMA(4,2,1)
PSO		
Dimensions of PSO particle	96	
Swarm size	300	
Initialization of particle positions	Random in $[0, 1]^{96}$	
Initialization of particle velocities	Random in $[0, 1]^{96}$	
Learning rate (c_1,c_2)	(2,2)	
Inertia weight w	0.9	
Max training iterations	300	
RLSE		
Number of consequent-part parameters	25	30
$\boldsymbol{\theta}_0$	25×1 zero vector	30×1 zero vector
$\mathbf{P}_0$	$\alpha\mathbf{I}$	
α	10^8	
I	25×25 identity matrix	30×30 identity matrix

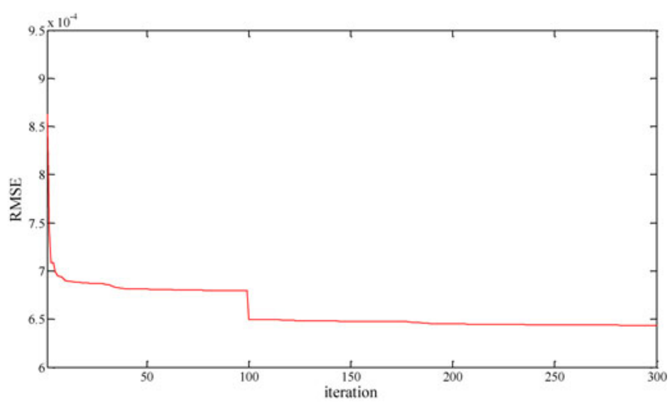


Fig. 6. Learning curve of CNFS(8)-ARIMA(4,1,0). (Mackey–Glass chaotic time series.)

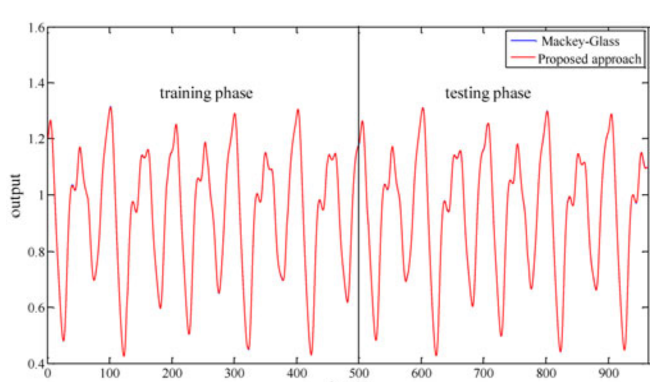


Fig. 7. Prediction response by CNFS(8)-ARIMA(4,1,0). (Mackey–Glass chaotic time series.)

TABLE VIII
PERFORMANCE COMPARISON (MACKEY–GLASS CHAOTIC TIME SERIES)

Method	MSE	RMSE	NDEI	NMSE	# of rules
FNT [32]		2.7×10^{-3}	-	-	-
LLWNN + hybrid [81]	-	3.6×10^{-3}	-	-	-
SVR [82]	-	-	-	4.5×10^{-3}	-
Bagging SVR [82]	-	-	-	2.0×10^{-3}	-
Boosting SVR 1 [82]	-	-	-	8.2×10^{-4}	-
Boosting SVR 2 [82]	-	-	-	8.0×10^{-4}	-
Cascaded-Correlation NN [83]	-	-	0.06	-	-
Back-Prop NN [83]	-	-	0.02	-	-
Sixth-order Polynomial [83]	-	-	0.04	-	-
Linear Predictive Method [83]	-	-	0.55	-	-
TSK-NFIS [69]	2.18×10^{-5}	-	0.0406	-	-
AR model [69]	3.2×10^{-5}	-	0.0492	-	-
NAR [69]	2.89×10^{-5}	-	0.0466	-	-
Neural Network [69]	3.21×10^{-5}	-	0.0488	-	-
SuPFuNIS [78]	-	-	0.014	-	15
G-FNN [79]	-	5.6×10^{-3}	-	-	10
ILA [80]	-	6.6×10^{-3}	-	-	12
AR model [13]	-	-	0.19	-	-
ANFIS [13]	-	1.5×10^{-3}	0.007	-	16
HyFIS [70]	-	2.9×10^{-3}	0.0101	-	16
RBF based AFS I [77]		1.14×10^{-2}	-	-	23
RBF based AFS II [77]		1.28×10^{-2}	-	-	21
RBF based AFS III [77]		1.31×10^{-2}	-	-	13
NEFPROX [84]		3.32×10^{-2}	-	-	129
ANCFIS [51]	3.10×10^{-7}	-	0.0027	7.29×10^{-6} *	-
CNFS(8)-ARIMA(4,1,0) (proposed)	4.05×10^{-7}	6.4×10^{-4}	0.0031	9.61×10^{-6}	8
CNFS(8)-ARIMA(4,2,0) (proposed)	5.84×10^{-7}	7.6×10^{-4}	0.0037	1.37×10^{-5}	8

* Note that in [51] the data of NMSE was given as 1.37×10^{-6}, which is inconsistent with its NDEI (0.0027). The inconsistency can be checked by the definitions of NMSE and NDEI in eqs. (50) and (51) of [51]. According to the data 0.0027 of NDEI, the NMSE is calculated to be 7.29×10^{-6}.

C. Example 3—Dual Time Series of Daily National Association of Securities Dealers Automated Quotation Composite Index

In this example, we use the real-world time series of National Association of Securities Dealers Automated Quotation

(NASDAQ) composite index to illustrate *the dual-output property* of the proposed CNFS-ARIMA approach, by which the daily opening and closing indices of NASDAQ are predicted simultaneously. Note that the proposed CNFS-ARIMA can produce a complex-valued output, where the real and imaginary parts can be used for the daily opening and closing indices of NASDAQ, respectively. NASDAQ is an important stock index in the U.S. stock markets. NASDAQ is composed of the stocks of hundreds of high-tech companies, such as Microsoft, Intel, Yahoo, and many others, covering the latest technology fields of software, computers, telecommunications, and various high-techs. Hence, NASDAQ has been viewed as one of the most important indices for the global industries and markets. From January 3, 2007 to December 20, 2010, 1000 data from daily opening and closing indices of NASDAQ [85] were collected and normalized to the unit interval [0, 1]. These data are denoted as $\{(y_1(t), y_2(t)), t = 1, 2, \ldots, 1000\}$, where $y_1(t)$ indicates the daily opening index; $y_2(t)$ represents the daily closing index; and t is the generic time index. The first 500 data of the dual sequences are used for training, and the remaining data are used for testing. They are arranged in the form of (*input, target*) for 998 data pairs, which are denoted as $\{(\mathbf{H}(k), d(k)), k = 1, 2, \ldots, 998\}$

$$\mathbf{H}(k) = [y_1(t-1),\ y_1(t),\ y_2(t-1),\ y_2(t)]^{\mathrm{T}} \quad (49)$$

$$d(k) = y_1(t+1) + jy_2(t+1) \quad (50)$$

where $t = k + 1$; $j = \sqrt{-1}$; and $d(k)$ is the complex-valued target of the kth data pair. Each component of the input $\mathbf{H}(k)$ is mapped to its own universe, on which CFSs exist and the complex-valued membership degrees are computed. These complex-valued membership degrees are surely within the unit disc of the complex plane, as shown in Fig. 1(c). Hence, the condition of those corner cases (that include the complex values $1 + 1j$, $1 - 1j$, $-1 - 1j$ and $-1 + 1j$) or other complex values outside the unit disc cannot happen in the calculation of complex-valued membership degrees in the proposed approach. Complex-valued membership degrees have to follow the restriction that they have to be within the unit disc of the complex plane. However, such a restriction is neither for the model output nor for the target.

For structure learning, the FBSA method whose settings are given in Table I is used for the training set to generate a model prototype, based on which several proposed models with different orders are experimented. Five clusters are determined. The cost function is designed with MSE for parameter learning, where the PSO-RLSE method is used whose settings are the same as given in Table VII, except the number of rules is five, and the dimensions of PSO particles are 60. For performance comparison, the proposed models are compared with other approaches, such as ANFIS [13], RBF network [71], and SVR [72], [73]. ANFIS and RBF were implemented by MATLAB; SVR was implemented by the LIBSVM tool package [74]. We compare these models in two ways. In the first case, each of ANFIS, RBF, and SVR is set up with two models, where each is with single output. One is for the opening index, and the other is for the closing index. In the second case, each of the ANFIS and RBF models is set up with two outputs for dual time-series forecasting of the opening and closing indices of NASDAQ concurrently. The performance comparison is shown in Table IX, where the results are restored to their original ranges.

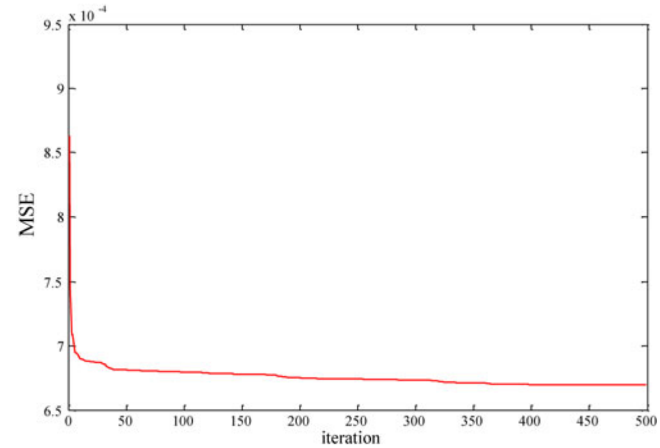


Fig. 8. Learning curve of CNFS(5)-ARIMA(4,2,0). (NASDAQ dual time series.)

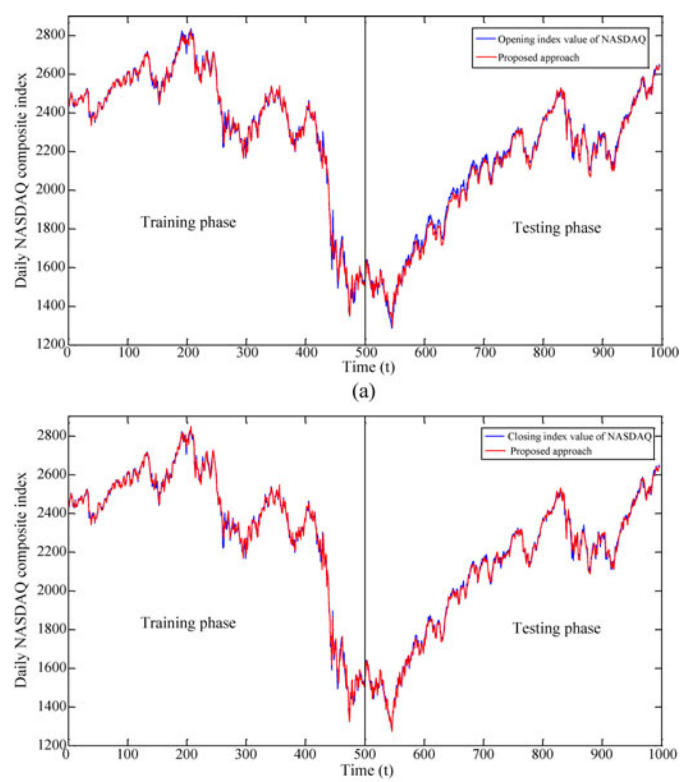


Fig. 9. Dual prediction responses by the proposed CNFS(5)-ARIMA(4,2,0) for NASDAQ dual time series (in real ranges). (a) Daily opening index. (b) Daily closing index.

The learning curve is shown in Fig. 8 for the proposed CNFS(5)-ARIMA(4,2,0), whose dual responses are given in Fig. 9(a) and (b). Through the experimental results, we have successfully demonstrated the proposed approach for the dual-output property that is induced by CFSs, as stated previously.

TABLE IX
PERFORMANCE COMPARISON (NASDAQ DUAL TIME SERIES)

Method	RMSE			
	Training phase		Testing phase	
	Opening index	Closing index	Opening index	Closing index
SVR (two models, each with single output)	35.18	35.24	37.23	40.24
ANFIS (two models, each with single output)	37.83	38.66	38.80	42.36
ANFIS (one model with two outputs)	62.75	71.51	72.52	85.08
RBF (two models, each with single output)	37.59	33.89	37.52	44.08
RBF (one model with two outputs)	178.57	179.87	261.37	258.89
CNFS(5)-ARIMA(4,0,0)	31.52	37.13	48.15	46.92
CNFS(5)-ARIMA(4,0,1)	37.91	39.78	47.02	60.52
CNFS(5)-ARIMA(4,1,0)	31.14	36.19	42.67	41.82
CNFS(5)-ARIMA(4,1,1)	33.75	40.56	40.75	44.36
CNFS(5)-ARIMA(4,2,0)	**21.56**	**20.81**	**32.52**	**33.70**
CNFS(5)-ARIMA(4,2,1)	31.08	30.23	44.28	41.06

D. Example 4—Dual Time Series of Taiwan Stock Exchange Capitalization Weighted Stock Index and Dow Jones Industrial Average Index

We use this example to illustrate the dual-output capability of the proposed approach for two different sequences of time series in dissimilar stock markets. We apply the proposed approach to two real-world time series: Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) and Dow Jones Industrial Average Index (DJI). The proposed CNFS-ARIMA approach was tested to predict daily closing indices of TAIEX and DJI concurrently. For TAIEX and DJI, their daily closing indices [86], [87] were collected for six years from 1999 to 2004. The range of dataset is normalized to the interval [0, 1]. For each year, the first ten-month data are used for training, and the last two-month data are used for testing. The proposed approach is compared with other methods [10], [88]–[92], and performance comparison is conducted year by year. The data are arranged in the form of (*input, target*) data pairs, which are denoted as $\{(\mathbf{H}(k), d(k)), k = 1, 2, \ldots, n\}$, where n is the number of data pairs, given as follows:

$$\mathbf{H}(k) = [y_1(t-1),\ y_1(t),\ y_2(t-1),\ y_2(t)]^{\mathrm{T}} \tag{51}$$

$$d(k) = y_1(t+1) + jy_2(t+1) \tag{52}$$

where $t = k + 1$ is the generic time index; $j = \sqrt{-1}$; $d(k)$ is the complex-valued target of the kth data pair; $y_1(t)$ indicates TAIEX index; and $y_2(t)$ indicates DJI index. For structure learning, the FBSA method whose settings are given in Table I is used for the training set. Four clusters are determined, based on which four fuzzy IF–THEN rules whose antecedents are described by Gaussian CFSs are generated for the CNFS(4)-ARIMA(4,2,0). For parameter learning, the PSO-RLSE method is applied, whose settings are given in Table VII, except the number of rules is four, and the dimensions of PSO particles are 48. The cost function is designed with RMSE. After parameter learning, the proposed model can produce complex-valued outputs, whose real and imaginary parts are used for the forecasts of TAIEX and DJI indices, respectively. These forecasts are restored to their real ranges for performance comparison. The proposed models are compared with many other approaches in the literature, as well as the famous approaches of ANFIS [13], RBF network [71] and SVR [72], [73]. ANFIS and RBF were implemented by MATLAB; SVR was implemented by the LIBSVM tool package [74]. For ANFIS, RBF, and SVR, we compare these models in two ways. In the first case, each of ANFIS, RBF, and SVR is set up with two models, where each is with single output. One is for TAIEX, and the other is for DJI. In the second case, each of ANFIS and RBF models is set up with two outputs for dual time-series forecasting of the TAIEX and DJI concurrently. For TAIEX time series, the performance comparison is shown in Table X. For DJI, the performance comparison is summarized in Table XI. Through the experimental results, the proposed approach has shown excellent forecasting performance. For the year 2004, the dual prediction responses and errors by the proposed approach are shown in Fig. 10 for the TAIEX and DJI indices.

V. DISCUSSION AND CONCLUSION

We have presented the proposed complex neurofuzzy computing approach that combines the theories of ARIMA and CNFS to mold the so-called CNFS-ARIMA for prediction. To self-construct the proposed CNFS-ARIMA automatically, we have introduced machine learning methods in the study that include the FBSA clustering method for structure learning to automatically determine the initial knowledge base of CNFS-ARIMA and the PSO-RLSE method for fast parameter learning to fine tune the CNFS-ARIMA. The proposed approach has revealed the advantage of the *dual-output property* that can be used to study two different time series concurrently. The proposed approach has been successfully applied to the problem of time-series forecasting with four examples in which the proposed approach has been compared with several other approaches in the literature. The experimental results indicate that the proposed approach has shown excellent performance for time-series forecasting. The excellent performance by the proposed approach stems from the excellent mapping ability of CNFS-ARIMA,

TABLE X
PERFORMANCE COMPARISON IN RMSE (TAIEX TIME SERIES)

Method \ Year	1999	2000	2001	2002	2003	2004	Average RMSE
Multivariate heuristic fuzzy time-series model [89] (Use TAIEX & NASDAQ)	-	158.7	136.49	95.15	65.51	73.57	105.88
Multivariate heuristic fuzzy time-series model [89] (Use TAIEX & Dow Jones)	-	165.8	138.25	93.73	72.95	73.49	108.84
Multivariate heuristic fuzzy time-series model [89] (Use TAIEX & M_{1b})	-	169.19	133.26	97.1	75.23	82.01	111.36
Multivariate heuristic fuzzy time-series model [89] (Use TAIEX, NASDAQ & Dow Jones)	-	157.65	131.98	93.48	65.51	73.49	104.42
Multivariate heuristic fuzzy time-series model [89] (Use TAIEX, NASDAQ & M_{1b})	-	155.51	128.44	97.15	70.76	73.48	105.07
Multivariate heuristic fuzzy time-series model [89] (Use TAIEX, NASDAQ, Dow Jones & M_{1b})	-	154.42	124.02	95.73	70.76	72.35	103.46
Fuzzy time series model (U_FTS Model) [88],[90]-[91]	120	176	148	101	74	84	117.4
Univariate conventional regression model (U_R Model) [90]-[91]	164	420	1070	116	329	146	374.2
Univariate neural network model (U_NN Model) [90]-[91]	107	309	259	78	57	60	145.0
Univariate neural network-based fuzzy time series model (U_NN_FTS Model) [90]-[91]	109	255	130	84	56	116	125.0
Univariate neural network-based fuzzy time series model with substitutes (U_NN_FTS_S Model) [90]-[92]	109	152	130	84	56	116	107.8
Bivariate conventional regression model (B_R Model) [90]-[91]	103	154	120	77	54	116	107.8
Bivariate neural network model (B_NN Model) [90]-[91]	112	274	131	69	**52**	61	116.4
Bivariate neural network-based fuzzy time series model (B_NN_FTS Model) [90]-[91]	108	259	133	85	58	67	118.3
Bivariate neural network-based fuzzy time series model with substitutes(B_NN_FTS_S Model) [90]-[91]	112	131	130	80	58	67	96.4
Fuzzy time series model with FLRGs (Use TAIEX & Dow Jones) [10]	115.47	127.51	121.98	74.65	66.02	58.89	94.09
Fuzzy time series model with FLRGs (Use TAIEX & NASDAQ) [10]	119.32	129.87	123.12	71.01	65.14	61.94	95.07
Fuzzy time series model with FLRGs (Use TAIEX & M_{1b}) [10]	120.01	129.87	117.61	85.85	63.1	67.29	97.29
Fuzzy time series model with FLRGs (Use TAIEX, NASDAQ & Dow Jones) [10]	116.64	123.62	123.85	71.98	58.06	57.73	91.98
Fuzzy time series model with FLRGs (Use TAIEX, NASDAQ & M_{1b}) [10]	114.87	128.37	123.15	74.05	67.83	65.09	95.56
Fuzzy time series model with FLRGs (Use TAIEX, NASDAQ, Dow Jones & M_{1b}) [10]	112.47	131.04	117.86	77.38	60.65	65.09	94.08
SVR (two models, each with single output)	116.11	151.09	162.46	67.72	59.47	58.81	102.61
ANFIS (two models, each with single output)	101.93	131.97	147.36	70.17	72.61	65.33	98.23
ANFIS (one model with two outputs)	125.24	147.10	151.62	78.27	81.69	70.54	109.08
RBF (two models, each with single output)	119.81	161.51	134.32	65.15	60.41	102.86	107.34
RBF (one model with two outputs)	129.79	181.18	137.58	78.54	115.92	126.48	128.25
CNFS(4)-ARIMA(4,2,0) (proposed)	**100.01**	**122.58**	**115.82**	**64.34**	**57.69**	**55.56**	**86.00**

where CFSs are important constituents for such achievements. CFSs can provide more degrees of freedom of adaption flexibility for learning of the proposed models than regular fuzzy sets. Thanks to the characteristics of complex-valued memberships, more affluent information by complex-valued membership degrees can be supplied into the proposed CNFS-ARIMA for excellent functional mapping by which accurate prediction can be achieved. Moreover, the dual-output property of the proposed approach is induced by such complex-valued membership characteristics. Based on the concept of divide and conquer, the parameter space of CNFS-ARIMA is spiritually divided into two smaller spaces: the IF-part subspace and the THEN-part subspace. The PSO-RLSE method [43], [53]–[56] was used to update the IF-part and THEN-part parameters of the proposed CNFS-ARIMA models. This method has worked efficiently in this study. For example, the learning curve of CNFS(4)-ARIMA(3,2,0) that is shown in Fig. 4 has recorded the parameter learning process. By the PSO-RLSE method, the IF-part parameters and THEN-part parameters were updated in a hybrid and fast way so that the MSE reached 1.86×10^{-4} at the first

TABLE XI
PERFORMANCE COMPARISON IN RMSE (DJI TIME SERIES)

Method \ Year	1999	2000	2001	2002	2003	2004	Average RMSE
SVR (two models, each with single output)	109.05	134.78	101.44	117.95	82.76	71.49	102.91
ANFIS (two models, each with single output)	111.20	135.76	105.56	111.69	72.09	68.00	100.72
ANFIS (one model with two outputs)	120.15	138.71	128.20	142.05	90.37	83.69	117.20
RBF (two models, each with single output)	128.38	143.13	106.33	131.24	97.58	81.79	114.74
RBF (one model with two outputs)	149.96	158.41	181.79	136.28	154.14	148.11	154.78
CNFS(4)-ARIMA(4,2,0) (proposed)	**102.05**	**130.69**	**103.06**	**103.42**	**70.70**	**66.55**	**96.08**

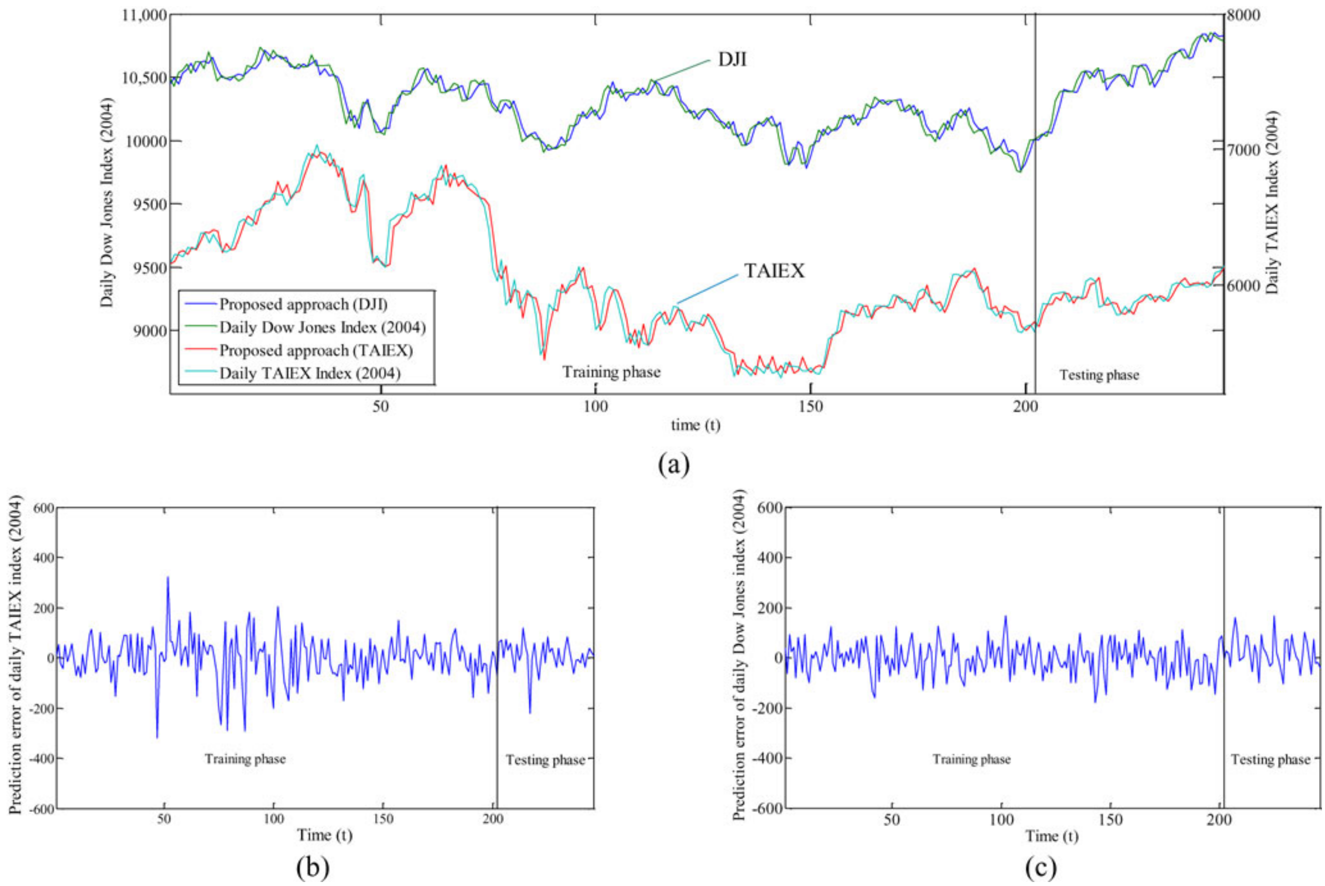


Fig. 10. Dual prediction responses by the proposed CNFS(4)-ARIMA(4,2,0). (a) Real-part prediction response for TAIEX (2004) and imaginary-part prediction response for DJI (2004). (b) Prediction error for TAIEX. (c) Prediction error for DJI.

iteration, 1.76×10^{-4} at the 20th iteration, and 1.74×10^{-4} at the final iteration. This implies that the PSO-RLSE can quickly reach the near-optimal region and look for the optimal solution. Such a PSO-RLSE learning process is usually dependent on the search efficacy by PSO, because according to the "gbest" particle for the IF-part parameters, RLSE can always find the corresponding solution for the THEN-part parameters efficiently.

In Example 1 for star brightness time series, the proposed CNFS(4)-ARIMA(3,2,0) performs better than the compared approaches [5], [51], [69]. As shown in Table IV, the proposed approach has the prediction performance 1.94×10^{-4} in the MSE for the testing phase. Such a performance is 99% better than the artificial neural network ARMA model [5], whose performance is 0.7534 in MSE, and 38% better than the nonlinear autoregressive model [69] and the Neural Net model [69] whose performances are 3.12×10^{-4} and 3.11×10^{-4} in MSE, respectively. Moreover, the proposed CNFS(4)-ARIMA(3,0,0) (with Gaussian CFSs) was compared with the traditional NFS(4)-ARIMA(3,0,0) (with regular Gaussian fuzzy sets). For 20 repeated trials, the former has the prediction performance of 2.17×10^{-4} (3.15×10^{-6}) in mean (standard deviation) of MSE for the testing phase, while the latter has 3.03×10^{-4} (3.18×10^{-5}) in mean (standard deviation) of MSE. In addition, the proposed model shows much smaller standard deviation and is 28% better in MSE than the compared NFS. This result shows that the proposed approach with CFSs has better adaption capability and more stable fidelity with much smaller standard deviation in performance accuracy than its traditional counterpart. Furthermore, to see performance effects by the structure- and parameter-learning methods, we did several more experiments on which different combinations were tested with and without FBSA, G-KFCM [75], RLSE, and PSO-RLSE. These results are summarized in Table VI for star brightness time series. With FBSA and PSO-RLSE for structure and parameter learning phases, the proposed CNFS(4)-ARIMA(3,2,0) is compared with the traditional NFS(4)-ARIMA(3,2,0) (with regular Gaussian fuzzy sets). The former has the prediction performance of 1.81×10^{-4} in MSE for the testing phase, which is 32% better than the latter, which has 2.67×10^{-4} in MSE. This encourages our thought that the property of CFSs for the proposed approach can augment the functional mapping ability for forecasting accuracy. For the models that only use FBSA and RLSE, their performances are worse than the models that use the PSO-RLSE learning method. In general, the PSO-RLSE algorithm has strong power to promote the performances by either the NFS-ARIMA or CNFS-ARIMA models. Based on the same PSO-RLSE hybrid learning method, the proposed CNFS(4)-ARIMA(3,2,0) model that used FBSA for structure learning shows 27% better performance than CNFS(4)-ARIMA(3,2,0) model that did not use FBSA for structure learning. Furthermore, we also used the G-KFCM clustering method to replace the FBSA method for the structure learning phase. In Table VI, for the CNFS(4)-ARIMA(3,2,0), the proposed method (FBSA + PSO-RLSE) shows the best performance, which is slightly better than the compared method (G-KFCM + PSO-RLSE). We believe that both clustering methods (i.e., G-KFCM and FBSA) can properly divide the training data (distributed in the input space) into a appropriate number of clusters, and based on this, fuzzy IF–THEN rules can be generated for CNFS-ARIMA. In addition, the PSO-RLSE hybrid learning method has a strong power to fine tune the models for good performance. For both methods, their performances are comparable.

In Example 2, for the Mackey–Glass chaotic time series, as shown in Table VIII, the proposed CNFS(8)-ARIMA(4,1,0) with eight fuzzy rules has the prediction performance of 4.05 $\times 10^{-7}$ in MSE, 6.4×10^{-7} in RMSE, 0.0031 in NDEI, and 9.61×10^{-6} in NMSE. This performance is 76% better than the FNT [32], whose performance is 2.7×10^{-3} in RMSE, which is 99% better than Boosting SVR 2 [82], whose performance is 8.0 $\times 10^{-4}$ in NMSE, 74% better than the SuPFuNIS method [78], whose performance is 0.014 in NDEI, and comparable with the ANCFIS [51], which also used the concept of CFSs.

In Examples 3 and 4, the proposed approach has successfully illustrated the *dual-output* performance, as shown in Figs. 9 and 10. The proposed CNFS-ARIMA approach (with CFSs) has shown such a new dual-output property, which is distinguished from other neurofuzzy approaches (with regular fuzzy sets). In Example 3, for dual time series of daily NASDAQ composite index, as shown in Table IX, the proposed CNFS(5)-ARIMA(4,2,0) has the prediction performance of 32.52 in RMSE for opening index of NASDAQ in the testing phase. Compared with those by the ANFIS, RBF, and SVR that used two models (each with single output), this performance is 16% better than ANFIS, whose performance is 38.80 in RMSE, 12% better than the SVR, whose performance is 37.23 in RMSE, and 13% better than the RBF model, whose performance is 37.52 in RMSE. Similar results can be observed for the closing index of NASDAQ. Moreover, in Table IX, the ANFIS and RBF that used one model with two outputs to deal with the NASDAQ dual time series perform much worse than the similar ANFIS and RBF that used two models, whose two single outputs were used to handle the dual time series, respectively. The RBF model with single output has the prediction performance of 37.52 in RMSE for the opening index of NASDAQ in the testing phase. However, the RBF model with two outputs substantially deteriorates to 261.37 in RMSE. The ANFIS models with single and two outputs, respectively, have similar results as those by RBF.

In Example 4, for the dual time series of the TAIEX and DJI indices, as shown in Tables X and XI, the proposed approach has the prediction performance of 86 in mean RMSE for the TAIEX index in the testing phase. Such a performance is 9% better than the bivariate NN-based fuzzy time-series model with substitutes (B_NN_FTS_S Model) [90], [91], whose performance is 107.8 in mean RMSE, and 7% better than the fuzzy time-series model with fuzzy logic relationship groups (Use TAIEX, NASDAQ, Dow Jones, and M_{1b}) [10], whose performance is 94.08 in mean RMSE. Similarly, for the closing index of the DJI, the proposed approach performs better than the compared approaches, as shown in Table XI. For TAIEX and DJI in Example 4, the proposed approach with dual-output property performs better than ANFIS and RBF that used two models, of which each was with single output and one model with two outputs, as shown in Tables X and XI.

The advantages of the proposed approach are threefold. First, we have successfully applied the rationale of CFSs to the CNFS, which opens a new window to intelligent system-based research works and applications. Second, the CNFS using CFSs and ARIMA models is combined for their merits to mold the proposed CNFS-ARIMA, which has been successfully applied to the problem of time-series forecasting. This promising innovation has shown a successful realization of a complex-fuzzy-based computing paradigm for modeling and forecasting. Finally, we have devised the proposed CNFS-ARIMA approach to have the abilities of self-organization and self-learning. The proposed CNFS-ARIMA can construct its knowledge base by the FBSA clustering method and then fine tune the CNFS-ARIMA by the PSO-RLSE method [43], [53]–[56] to become as optimal as possible. Moreover, the proposed approach using CFSs has the unique dual-output property, which has been shown successfully in the applications of real-world dual time-series forecasting. Through performance comparison, the proposed approach has shown superior performance to other approaches in the literature. The experimental results indicate that the proposed approach has shown excellent performance.

ACKNOWLEDGMENT

The authors would like to express their sincere thanks to the anonymous reviewers for their valuable comments.

References

[1] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 246–254, Jun. 2009.

[2] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models," *IEEE Trans. Energy Convers.*, vol. 24, no. 3, pp. 775–782, Sep. 2009.

[3] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[4] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2664–2675, 2011.

[5] I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L. J. Herrera, H. Pomares, L. Marquez, and M. Pasadas, "Soft-computing techniques and ARMA model for time series prediction," *Neurocomputing*, vol. 71, no. 4–6, pp. 519–537, 2008.

[6] M. A. Boyacioglu and D. Avci, "An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7908–7912, 2010.

[7] S.-T. Li and Y.-C. Cheng, "A stochastic HMM-based forecasting model for fuzzy time series," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 5, pp. 1255–1266, Oct. 2010.

[8] W.-K. Wong, E. Bai, and A. W. Chu, "Adaptive time-variant models for fuzzy-time-series forecasting," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1531–1542, Dec. 2010.

[9] D. Ruta, B. Gabrys, and C. Lemke, "A generic multilevel architecture for time series prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 3, pp. 350–359, Mar. 2011.

[10] S.-M. Chen and C.-D. Chen, "TAIEX forecasting based on fuzzy time series and fuzzy variation groups," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 1–12, Feb. 2011.

[11] Y.-C. Cheng and S.-T. Li, "Fuzzy time series forecasting with a probabilistic smoothing hidden Markov model," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 2, pp. 291–304, Apr. 2012.

[12] B. Wang, S. Wang, and J. Watada, "Fuzzy-portfolio-selection models with value-at-risk," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 4, pp. 758–769, Aug. 2011.

[13] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[14] C. Li and R. Priemer, "Self-learning general purpose PID controller," *J. Franklin Inst.*, vol. 334, no. 2, pp. 167–189, 1997.

[15] C. Li and R. Priemer, "Fuzzy control of unknown multiple-input-multiple-output plants," *Fuzzy Sets Syst.*, vol. 104, no. 2, pp. 245–267, 1999.

[16] C. Li and C.-Y. Lee, "Self-organizing neuro-fuzzy system for control of unknown plants," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 1, pp. 135–150, Feb. 2003.

[17] C. Li, C.-Y. Lee, and K.-H. Cheng, "Pseudoerror-based self-organizing neuro-fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 6, pp. 812–819, Dec. 2004.

[18] P. C. Nayak, K. P. Subheer, D. M. Rangan, and K. S. Ramasastri, "A neuro-fuzzy computing technique for modeling hydrological time series," *J. Hydrol.*, vol. 291, no. 6, pp. 52–66, 2004.

[19] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[20] J. L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 25, no. 4, pp. 629–635, Apr. 1995.

[21] C. Brace, J. Schmidt, and M. Hadin, "Comparison of the forecasting accuracy of neural networks with other established techniques," in *Proc. First Int. Forum Appl. Neural Netw. Power Syst.*, 1991, pp. 31–35.

[22] W. R. Foster, F. Collopy, and L. H. Ungar, "Neural network forecasting of short noisy time series," *Comput. Chem. Eng.*, vol. 16, no. 4, pp. 293–297, 1992.

[23] J. Denton, "How good are neural networks for causal forecasting?," *J. Bus. Forecasting Methods Syst.*, vol. 14, no. 2, pp. 17–20, 1995.

[24] T. Taskaya-Temizel and M. C. Casey, "A comparative study of autoregressive neural network hybrids," *Neural Netw.*, vol. 18, no. 5–6, pp. 781–789, 2005.

[25] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[26] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2008.

[27] S. L. Ho and M. Xie, "The use of ARIMA models for reliability forecasting and analysis," *Comput. Ind. Eng.*, vol. 35, no. 1–2, pp. 213–216, 1998.

[28] V. Ş. Ediger and S. Akar, "ARIMA forecasting of primary energy demand by fuel in Turkey," *Energy Policy*, vol. 35, no. 3, pp. 1701–1708, 2007.

[29] M.-C. Tan, S. C. Wong, J.-M. Xu, Z.-R. Guan, and P. Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 60–69, Mar. 2009.

[30] P. Chen, T. Pedersen, B.-J Birgitte, and Z. Chen, "ARIMA-based time series model of stochastic wind power generation," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 667–676, May 2010.

[31] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[32] Y. Chen, B. Yang, J. Dong, and A. Abraham, "Time-series forecasting using flexible neural tree model," *Inf. Sci.*, vol. 174, no. 3–4, pp. 219–235, 2005.

[33] A. Jain and A. M. Kumar, "Hybrid neural network models for hydrologic time series forecasting," *Appl. Soft Comput.*, vol. 7, no. 2, pp. 585–592, 2007.

[34] C. Hamzaçebi, D. Akay, and F. Kutay, "Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3839–3844, 2009.

[35] L. Yu, S. Wang, and K. K. Lai, "A neural-network-based nonlinear metamodeling approach to financial time series forecasting," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 563–574, 2009.

[36] D. Moses, O. Degani, H. N. Teodorescu, M. Friedman, and A. Kandel, "Linguistic coordinate transformations for complex fuzzy sets," in *Proc. IEEE Int. Fuzzy Syst. Conf.*, Aug. 1999, vol. 3, pp. 1340–1345.

[37] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 171–186, Apr. 2002.

[38] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 450–461, Aug. 2003.

[39] S. Dick, "Toward complex fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 405–414, Jun. 2005.

[40] J. J. Buckley, "Fuzzy complex numbers," *Fuzzy Sets Syst.*, vol. 33, no. 3, pp. 333–345, 1989.

[41] J. J. Buckley and Y. Qu, "Fuzzy complex analysis—I: Differentiation," *Fuzzy Sets Syst.*, vol. 41, no. 3, pp. 269–284, 1991.

[42] J. J. Buckley, "Fuzzy complex analysis—II: Integration," *Fuzzy Sets Syst.*, vol. 49, no. 2, pp. 171–179, 1992.

[43] C. Li and T.-W. Chiang, "Complex neuro-fuzzy self-learning approach to function approximation," *Lect. Notes Artif. Intell.*, vol. 5991, pp. 289–299, 2010.

[44] K. Chakraborty, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural Netw.*, vol. 5, no. 6, pp. 961–970, 1992.

[45] H. Raman and N. Sunilkumar, "Multivariate modelling of water resources timeseries using artificial neural networks," *Hydrol. Earth Syst. Sci.*, vol. 40, no. 2, pp. 145–163, 1995.

[46] J. Nie, "Nonlinear time-series forecasting: A fuzzy-neural approach," *Neurocomputing*, vol. 16, pp. 63–76, 1997.

[47] J. C. Ochoa-Rivera, R. García-Bartual, and J. Andreu, "Multivariate synthetic streamflow generation using a hybrid model based on artificial neural networks," *Hydrol. Earth Syst. Sci.*, vol. 6, no. 4, pp. 641–654, 2002.

[48] D. Ramot, M. Friedman, G. Langholz, R. Milo, and A. Kandel, "On complex fuzzy sets," in *Proc. 10th IEEE Int. Conf. Fuzzy Syst.*, 2001, pp. 1160–1163.

[49] G. Zhang, T. S. Dillon, K.-Y. Cai, J. Ma, and J. Lu, "Operation properties and δ-equalities of complex fuzzy sets," *Int. J. Approx. Reason.*, vol. 50, no. 8, pp. 1227–1249, 2009.

[50] D. Qiu, L. Shu, and Z.-W. Mo, "Notes on fuzzy complex analysis," *Fuzzy Sets Syst.*, vol. 160, no. 11, pp. 1578–1589, 2009.

[51] Z. Chen, S. Aghakhani, J. Man, and S. Dick, "ANCFIS: A neurofuzzy architecture employing complex fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 305–322, Apr. 2011.

[52] J. Ma, G. Zhang, and J. Lu, "A method for multiple periodic factor prediction problems using complex fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 32–45, Feb. 2012.

[53] C. Li, T.-W. Chiang, J.-W. Hu, and T. Wu, "Complex neuro-fuzzy intelligent approach to function approximation," in *Proc. 2010 3rd Int. Workshop Adv. Comput. Intell.*, 2010, pp. 151–156.

[54] C. Li and T.-W. Chiang, "Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach," *Lect. Notes Artif. Intell.*, vol. 6592, pp. 242–251, 2011.

[55] C. Li and T.-W. Chiang, "Function approximation with complex neuro-fuzzy system using complex fuzzy sets—A new approach," *New Gener. Comput.*, vol. 29, no. 3, pp. 261–276, 2011.
[56] C. Li and T.-W. Chiang, "Complex fuzzy model with PSO-RLSE hybrid learning approach to function approximation," *Int. J. Intell. Inf. Database Syst.*, vol. 5, no. 4, pp. 409–430, 2011.
[57] H. Sun, S. Wang, and Q. Jiang, "FCM-based model selection algorithms for determining the number of clusters," *Pattern Recognit.*, vol. 37, no. 10, pp. 2027–2037, 2004.
[58] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, 1995, pp. 39–43.
[59] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 4, pp. 1942–1948.
[60] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proc. Congr. Evol. Comput.*, 2001, vol. 1, pp. 101–106.
[61] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.
[62] A. Chatterjee, K. Pulasinghe, K. Watenabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1478–1489, Dec. 2005.
[63] Y.-X. Liao, J.-H. She, and M. Wu, "Integrated hybrid-PSO and fuzzy-NN decoupling control for temperature of reheating furnace," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2704–2714, Jul. 2009.
[64] C.-F. Juang, C.-M. Hsiao, and C.-H. Hsu, "Hierarchical cluster-based multispecies particle-swarm optimization for fuzzy-system optimization," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 14–26, Feb. 2010.
[65] R. P. Prado, S. Garcia-Galan, J. E. M. Exposito, and A. J. Yuste, "Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 6, pp. 1083–1097, Dec. 2010.
[66] L. Ljung and E. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[67] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Cybern. Syst.*, vol. 3, no. 3, pp. 32–57, 1973.
[68] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Norwell, MA: Kluwer, 1981.
[69] D. Graves and W. Pedrycz, "Fuzzy prediction architecture using recurrent neural networks," *Neurocomputing*, vol. 72, no. 7–9, pp. 1668–1678, 2009.
[70] J. Kim and N. Kasabov, "HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems," *Neural Netw.*, vol. 12, no. 9, pp. 1301–1319, 1999.
[71] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.
[72] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.
[73] N. I. Sapankevychand and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, May 2009.
[74] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
[75] D. E. Gustafson and W.C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decis. Control*, 1979, pp. 761–766.
[76] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
[77] K. B. Cho and B. H. Wang, "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction," *Fuzzy Sets Syst.*, vol. 83, no. 3, pp. 325–339, 1996.
[78] S. Paul and S. Kumar, "Subsethood-product fuzzy neural inference system (SuPFuNIS)," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 578–599, May 2002.
[79] Y. Gao and M. J. Er, "NARMAX time series model prediction: Feedforward and recurrent fuzzy neural network approaches," *Fuzzy Sets Syst.*, vol. 150, no. 2, pp. 331–350, 2005.
[80] X. Deng and X. Wang, "Incremental learning of dynamic fuzzy neural networks for accurate system modeling," *Fuzzy Sets Syst.*, vol. 160, no. 7, pp. 972–987, 2009.
[81] Y. Chen, B. Yang, and J. Dang, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, no. 4–6, pp. 449–465, 2006.
[82] Y.-F. Deng, X. Jin, and Y.-X. Zhong, "Ensemble SVR for prediction of time series," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2005, vol. 6, pp. 3528–3534.
[83] R. S. Crowder, "Predicting the Mackey–Glass time series with cascade-correlation learning," in *Proc. Summer School Connectionist Models*, 1990, pp. 117–123.
[84] D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation," *Fuzzy Sets Syst.*, vol. 101, no. 2, pp. 261–271, 1999.
[85] (2011). Website of Yahoo Finance: *NASDAQ Composite Index* [Online]. Available: http://finance.yahoo.com/q?s=^IXIC
[86] (2011). Website of Yahoo Finance: *Dow Jones Industrial Average Index* [Online]. Available: http://finance.yahoo.com/q?s=^DJI
[87] (2011). Website of Yahoo Finance: *Taiwan Stock Exchange Capitalization Weighted Stock Index* [Online]. Available: http://finance.yahoo.com/q?s=^TWII
[88] S.-M. Chen, "Forecasting enrollments based on fuzzy time series," *Fuzzy Sets Syst.*, vol. 81, no. 3, pp. 311–319, 1996.
[89] K.-H. Huarng, T. H.-K. Yu, and Y. W. Hsu, "A multivariate heuristic model for fuzzy time-series forecasting," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 4, pp. 836–846, Aug. 2007.
[90] T. H.-K. Yu and K.-H. Huarng, "A bivariate fuzzy time series model to forecast the TAIEX," *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2945–2952, 2008.
[91] T. H.-K. Yu and K.-H. Huarng, "Corrigendum to "A bivariate fuzzy time series model to forecast the TAIEX" [Expert Systems with Applications 34 (4) (2010) 2945–2952]," *Expert Syst. Appl.*, vol. 37, no. 7, p. 5529, 2010.
[92] K. Huarng and T. H.-K. Yu, "The application of neural networks to forecast fuzzy time series," *Physica A*, vol. 363, no. 2, pp. 481–491, 2006.

Chunshien Li received the Ph.D. degree in electrical engineering and computer science from the University of Illinois at Chicago in 1996.

He is currently an Associate Professor with the Department of Information Management, National Central University, Taoyuan, Taiwan. He was with the Department of Computer Science and Information Engineering, National University of Tainan, Tainan, Taiwan, and the Department of Electrical Engineering, Chang Gung University, Taoyuan. He was also the Chief of the Information System Group of the Computer Center, and the Director of the Teacher Development Center, National University of Tainan. He has been the first author of more than 100 academic and technical publications. His biography has been included in Marquis Who's Who in Science and Engineering, in Medicine and Healthcare, in Asia, and in the World. 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.

Dr. Li received the Research Award from the National Science Council, Taiwan, in 1999 and the Research Award of Excellent Teacher from Chang Gung University in 2000. He has served as a referee of paper review for several IEEE journals and international journals and a program committee member for several international academic conferences.

Tai-Wei Chiang received the B.S. degree from Nanhua University, Chiayi, Taiwan, in 2006 and the M.S. degree from National Formosa University, Yunlin, Taiwan, in 2008, both in information management. He is currently working toward the Ph.D. degree with the Department of Information Management, National Central University, Taoyuan, Taiwan.

His research interests include fuzzy logic, neural networks, intelligent systems, machine learning, learning algorithms, and neurofuzzy inference systems.