Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 4, 787–800
DOI: 10.2478/v10006-012-0058-x

INTELLIGENT FINANCIAL TIME SERIES FORECASTING: A COMPLEX NEURO–FUZZY APPROACH WITH MULTI–SWARM INTELLIGENCE

CHUNSHIEN LI, TAI-WEI CHIANG

Laboratory of Intelligent Systems and Applications
National Central University, No. 300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan, ROC
e-mail: `jamesli@mgt.ncu.edu.tw`, `taiwei.chiang@gmail.com`

Financial investors often face an urgent need to predict the future. Accurate forecasting may allow investors to be aware of changes in financial markets in the future, so that they can reduce the risk of investment. In this paper, we present an intelligent computing paradigm, called the Complex Neuro-Fuzzy System (CNFS), applied to the problem of financial time series forecasting. The CNFS is an adaptive system, which is designed using Complex Fuzzy Sets (CFSs) whose membership functions are complex-valued and characterized within the unit disc of the complex plane. The application of CFSs to the CNFS can augment the adaptive capability of nonlinear functional mapping, which is valuable for nonlinear forecasting. Moreover, to optimize the CNFS for accurate forecasting, we devised a new hybrid learning method, called the HMSPSO-RLSE, which integrates in a hybrid way the so-called Hierarchical Multi-Swarm PSO (HMSPSO) and the well-known Recursive Least Squares Estimator (RLSE). Three examples of financial time series are used to test the proposed approach, whose experimental results outperform those of other methods.

Keywords: complex fuzzy set, complex neuro-fuzzy system, hierarchical multi-swarm particle swarm optimization, recursive least squares estimator, time series forecasting.

1. Introduction

Financial time series forecasting is an interesting research issue that has received growing attention in recent years. Accurate and reliable forecasting of future trends is a crucial undertaking in investment decision making. However, financial markets are complex and dynamic with high risk, because several complex forces affect them frequently. For the cases of financial time series, they are usually non-stationary, hence the mean, the variance and the covariance change over time. Due to many influential factors, it is hard to explore a closed form for the relation between historical data and future trends in the financial markets. Since accurate prediction of financial time series is regarded as a difficult nonlinear problem, traditional approaches that require mathematical models are not able to attack such problems with satisfactory performance.

To deal with time series forecasting problems, various intelligent approaches have been presented (Rojas *et al.*, 2008; Graves and Pedrycz, 2009; Brdyś *et al.*, 2009; Simiński, 2010; Khashei and Bijari, 2011; Vo *et al.*, 2011; Smetek and Trawinski, 2011; Tung and Quek, 2011) where fuzzy systems and neural networks have been widely investigated. Rojas *et al.* (2008) investigated a hybrid methodology that combines ANN and ARMA models for the problem of time series forecasting. Graves and Pedrycz (2009) developed a fuzzy rule based time series prediction model which builds upon an architecture of Takagi–Sugeno (T–S) rule-based models to investigate and evaluate the proposed rule-based model against commonly used time series models.

Khashei and Bijari (2011) proposed a hybridization of artificial neural networks and AutoRegressive Integrated Moving Average (ARIMA) models, in which the latter are used to identify the existing linear structure in data, and then a neural network is employed to determine a model to capture the underlying data-generating process. Tung and Quek (2011) presented a self-organising neural-fuzzy semantic network named the evolving Fuzzy Semantic Memory (eFSM) model to forecast the volatility levels of the Hang Seng Index.

Although neural networks have an excellent learning ability, it is not easy to create a meaningful explanation by human knowledge. For fuzzy systems, human experience and knowledge can be extracted with fuzzy If-Then rules, which can be easily mapped into practical application domains. Moreover, neural networks and fuzzy systems have proved to be universal approximators (Hornik *et al.*, 1989; Castro, 1995). By combining the flexible learning capability and the link-type distributed structure of neural networks and the fuzzy inference ability of a fuzzy system, Jang (1993) presented the Adaptive Network based Fuzzy Inference System (ANFIS), where the backpropagation method and the recursive least squares estimator are integrated to train the ANFIS predictive model.

Hybrid neural fuzzy models have demonstrated great performance in time series forecasting (Jang, 1993; Kim and Kasabov, 1999; Paul and Kumar, 2002; Gao and Er, 2005; Mousavi *et al.*, 2007; Deng and Wang, 2009; Graves and Pedrycz, 2009; Boyacioglu and Avci, 2010; Li and Cheng, 2011). For example, Gao and Er (2005) focused on the modeling and prediction using the Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) model with the Fuzzy Neural Network (FNN) methodology for time series forecasting. Mousavi *et al.* (2007) used Fuzzy Regression (FR) and an ANFIS for an reservoir operations optimization problem. Boyacioglu and Avci (2010) investigated the predictability of stock market return with an ANFIS to determine whether an ANFIS algorithm is capable of accurately predicting stock market return. Li and Cheng (2011) presented a self-organizing neural fuzzy system to forecast the S&P 500 time series.

In the past decade, Ramot *et al.* (2002; 2003) proposed the novel concept of Complex Fuzzy Sets (CFSs). In general, a CFS is an advanced fuzzy set whose membership function is complex-valued and defined within the unit disc of the complex plane. The complex-valued membership of a CFS is characterized by both an amplitude function and a phase function. This complex-valued property is in contrast with that of a traditional type-1 fuzzy set, whose membership function is normally defined in the unit real-valued interval [0, 1]. CFSs are not fuzzy complex numbers developed by Buckley (1989), which are ordinary type-1 fuzzy sets, whose members are complex-valued. From the perspective of the membership degree, CFSs have more degrees of freedom than standard type-1 fuzzy sets, and they can expand the capability for adaptation of an adaptive system because of their complex-valued memberships which enable novel nonlinear mappings. Although preliminary study of CFSs has been presented (Ramot *et al.*, 2002; 2003; Dick, 2005; Man *et al.*, 2007), it is still interesting to construct intuitively understandable complex-fuzzy-set based systems for applications. Zhang *et al.* (2009) focused on discussing operation properties for CFSs without conferring on complex fuzzy reasoning procedures. Chen *et al.* (2011) presented the Adaptive Neuro-Complex Fuzzy Inferential System (ANCFIS) for the problem of time series forecasting, where they implemented CFSs based on the equivalence of the CFSs phase to their support. In our previous work (Li *et al.*, 2010; Li and Chiang, 2011a; 2011b; 2011c), neuro-fuzzy computing models using complex fuzzy sets have been presented for real-world applications, with an excellent mapping ability and learning adaptability.

In this paper, with CFSs we present a Complex Neuro-Fuzzy System (CNFS) as a new computing paradigm for the problem of time series forecasting. The CNFS can extend the adaptability of traditional NFS models that use ordinary type-1 fuzzy sets (Jang, 1993; Li and Priemer, 1997; 1999; Li and Lee, 2003; Li *et al.*, 2004). We designed complex Gaussian fuzzy sets for the premises of fuzzy If-Then rules in the CNFS. For parameter learning, we devised a novel Hierarchical Multi-Swarm Particle Swarm Optimization (HMSPSO) algorithm in the way that multiple PSO swarms are arranged in a hierarchical structure to enhance the searching multiplicity and efficiency. Then, using the concept of divide-and-conquer, we developed a new hybrid learning method, called the HMSPSO-RLSE, which combines HMSPSO and the Recursive Least Squares Estimator (RLSE) algorithm in a hybrid way.

Our idea for fast learning is to separate the parameter space into two smaller ones, so that the algorithm can quickly approach an optimal (or near optimal) solution for the proposed CNFS. For the HMSPSO-RLSE learning method, HMSPSO is used to update the premise parameters of the CNFS and the RLSE is used to adjust the consequent parameters. Three examples of financial time series are used to test the proposed approach. The experimental results are compared with those of other approaches (Lu *et al.*, 2009; Tung and Quek, 2011) in terms of performance. The study has three major contributions. First, we developed the CNFS computing paradigm, where new complex Gaussian fuzzy sets are designed for fuzzy If-Then rules. Second, for fast learning we introduced the HMSPSO-RLSE hybrid method and applied it successfully to CNFS prediction models for accurate forecasting. Third, we successfully employed the proposed approach in the applications of real-world financial time series with excellent performance.

We organize the rest of the paper as follows. In Section 2, the proposed complex neuro-fuzzy system with complex fuzzy sets is specified. In Section 3, the HMSPSO-RLSE hybrid learning method is given. In Section 4, three examples of financial time series are used to test the proposed approach. In Section 5, we discuss the experimental results and the proposed approach. Finally, we conclude the paper.

2. Methodology

For time series forecasting, we present an adaptive intelligent model, called the complex neuro-fuzzy system, which is based on the theory of Neuro-Fuzzy Systems (NFSs) and that of complex fuzzy sets. Inheriting the property of universal approximation of the NFS (Hornik *et al.*, 1989; Castro, 1995), the proposed CNFS can approximate highly nonlinear functions with excellent accuracy and provide an outstanding mapping capability for time series forecasting. A CFS is an advanced fuzzy set whose membership function is characterized within the unit disc of the complex plane. That is, a CFS can be contrasted with an ordinary type-1 fuzzy set because of its complex-valued membership description in the two-dimensional unit disc of the complex plane. Therefore, a CFS can provide more degrees of freedom for learning adaptation than a traditional fuzzy set.

2.1. Complex fuzzy sets.

The theory of complex fuzzy sets (Moses *et al.*, 1999; Ramot *et al.*, 2002; 2003; Dick, 2005) has provided a new perspective in fuzzy theory research and application. Suppose we have a complex fuzzy set, *S*, whose membership function $\mu_s(h)$ is given as follows:

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

where $j = \sqrt{-1}$; h is the base variable for the complex fuzzy set; $r_s(h)$ is the amplitude function of the complex-valued membership; $\omega_s(h) \in \mathbb{R}$ is the phase function; $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ indicate the real and imaginary parts of $\mu_s(h)$, respectively. The property of sinusoidal waves is embodied in the definition of the complex fuzzy set. For the special case when $\omega_s(h)$ equals zero, a complex fuzzy set degenerates to a traditional type-1 fuzzy set. This fact indicates that the phase term is the key that distinguishes the CFS from its regular counterpart. We present a class of complex Gaussian fuzzy sets whose membership function, denoted by *cGaussian*(h, m, σ, λ), is

$$\begin{aligned}&\textit{cGaussian}(h, m, \sigma, \lambda)\\ &= r_s(h, m, \sigma, \lambda)\exp(j\omega_s(h, m, \sigma, \lambda)),\end{aligned} \tag{2a}$$

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

$$\begin{aligned}\omega_s(h, m, \sigma, \lambda) = &-\exp\Big[-0.5\Big(\frac{h-m}{\sigma}\Big)^2\Big]\\ &\times\Big(\frac{h-m}{\sigma^2}\Big)\lambda,\end{aligned} \tag{2c}$$

where $\{m, \sigma, \lambda\}$ are the parameters for the mean, spread and phase frequency factor, respectively. An illustration for a complex Gaussian fuzzy set is shown in Fig. 1.

2.2. Complex neuro-fuzzy system.

In the section, we specify the proposed complex neuro-fuzzy system, which possesses the merits of a neuro-fuzzy system and complex fuzzy sets. The proposed CNFS can achieve high performance for nonlinear functional mappings because CFSs can provide more degrees of freedom for learning adaption than ordinary type-1 fuzzy sets. It has an excellent nonlinear mapping capability for modeling and forecasting. Suppose we have a CNFS whose knowledge base is composed of *K* first-order Takagi–Sugeno fuzzy rules with *M* inputs and one output, given as follows:

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

for $i = 1, 2, \ldots, K$, where i indicates the i-th fuzzy rule; x_j and h_j are the j-th linguistic and base variables, respectively; $A_j^{(i)}(h_j)$ is the complex fuzzy set for the j-th premise condition of the i-th rule; $z^{(i)}$ is the nominal output of the i-th rule; $\{a_j^{(i)},\ j = 0, 1, \ldots, M\}$ are the consecutive parameters of the i-th rule.

A grid partition for the input space of the proposed CNFS is used. Each input linguistic variable has few linguistic values that are characterized by complex fuzzy sets. For the proposed CNFS, we use a six-layered feed-forward neural network to realize the process of complex fuzzy inference. The explanation of the six layers is specified as follows.

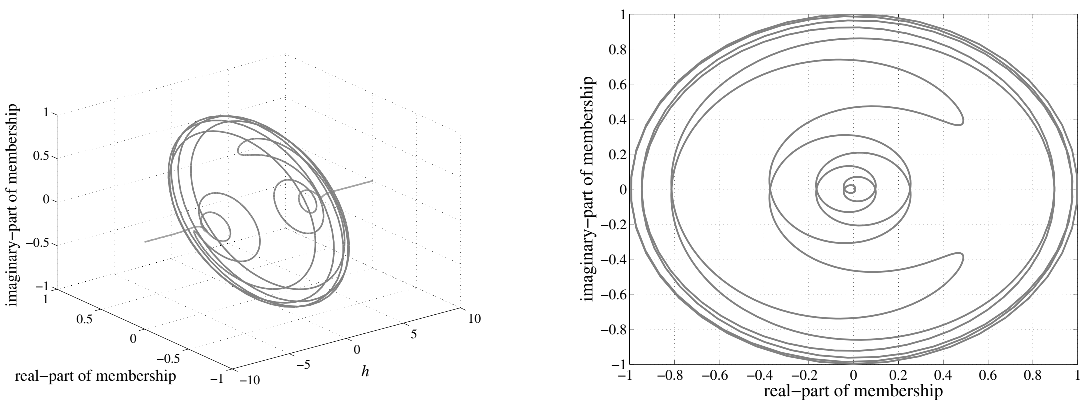
Layer 1: The input layer receives the inputs and sends them directly to the next layer. The input vector at time t is given as

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

Layer 2: The layer is called the complex-fuzzy-set layer, whose each node represents a linguistic value characterized by a CFS for the premises of the CNFS and calculates a complex-valued membership degree. The class of complex Gaussian fuzzy sets in (2a)–(2c) is used for the design of these CFSs.

Layer 3: This layer is called the fuzzy-rule layer, whose each node is used to calculate the firing strength of a fuzzy rule. In the paper, the operator of the *fuzzy-product* is used for *t*-norm calculation. For the *i*-th rule, the firing strength is expressed as follows:

$$\begin{aligned}\beta^{(i)}(t) &= \prod_{j=1}^{M} \mu_j^{(i)}(h_j(t)),\\ &= \prod_{j=1}^{M} r_j^{(i)}(h_j(t))\text{exp}(j\omega_j^{(i)}(h_j(t))),\end{aligned} \tag{5}$$



(a)

(b)

Fig. 1. Illustration of a complex Gaussian fuzzy set: 3-D view with the coordinates of the base variable, real-part membership and imaginary-part membership (a), 2-D view in the complex plane composed by the real and imaginary parts of the complex-valued membership function (b).

where $\mu_j^{(i)}(\cdot)$ is the complex-valued membership function of $A_j^{(i)}(\cdot)$ whose amplitude and phase functions are $r_j^{(i)}(\cdot)$ and $\omega_j^{(i)}(\cdot)$, respectively.

Layer 4: This layer is for the normalization of the firing strengths of fuzzy rules. For the *i*-th rule, the normalized firing strength is expressed as

$$\begin{aligned}\lambda^{(i)}(t) &= \frac{\beta^{(i)}(t)}{\sum\limits_{i=1}^{K}\beta^{(i)}(t)},\\ &= \frac{\prod\limits_{j=1}^{M} r_j^{(i)}(h_j(t))\exp(j\omega_j^{(i)}(h_j(t)))}{\sum\limits_{i=1}^{K}\prod\limits_{j=1}^{M} r_j^{(i)}(h_j(t))\exp(j\omega_j^{(i)}(h_j(t)))}.\end{aligned} \tag{6}$$

Layer 5: The layer is called the consequent layer, which is used for calculating normalized consequents. For the *i*-th rule, the normalized consequent is given as follows:

$$\begin{aligned}\xi^{(i)}(t) &= \lambda^{(i)}(t)z^{(i)}(t),\\ &= \frac{\prod\limits_{j=1}^{M} r_j^{(i)}(h_j(t))\exp(j\omega_j^{(i)}(h_j(t)))}{\sum\limits_{i=1}^{K}\prod\limits_{j=1}^{M} r_j^{(i)}(h_j(t))\exp(j\omega_j^{(i)}(h_j(t)))}\\ &\quad\times(a_0^{(i)}+\sum_{j=1}^{M}a_j^{(i)}h_j).\end{aligned} \tag{7}$$

Layer 6: This layer is called the output layer, to which the normalized consequents from Layer 5 are aggregated to produce the proposed CNFS output, given as

$$\begin{aligned}\xi^{(i)}(t) &= \lambda^{(i)}(t)z^{(i)}(t),\\ &= \frac{\prod\limits_{j=1}^{M} r_j^{(i)}(h_j(t))\exp(j\omega_j^{(i)}(h_j(t)))}{\sum\limits_{i=1}^{K}\prod\limits_{j=1}^{M} r_j^{(i)}(h_j(t))\exp(j\omega_j^{(i)}(h_j(t)))}\\ &\quad\times(a_0^{(i)}+\sum_{j=1}^{M}a_j^{(i)}h_j).\end{aligned} \tag{8}$$

Generally, owing to the nature of CFSs, the output of the CNFS is complex-valued

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

where $\xi_{\mathrm{Re}}(t)$and $\xi_{\mathrm{Im}}(t)$ are the real and imaginary parts of the CNFS output, respectively. Based on (9), the CNFS can be viewed as a complex-valued function,

$$\begin{aligned}\xi(t) &= \mathrm{F}(\mathbf{H}(t),\mathbf{W})\\ &= \mathrm{F}_{\mathrm{Re}}(\mathbf{H}(t),\mathbf{W})+j\mathrm{F}_{\mathrm{Im}}(\mathbf{H}(t),\mathbf{W}),\end{aligned} \tag{10}$$

where $\mathrm{F}_{\mathrm{Re}}(\cdot)$ and $\mathrm{F}_{\mathrm{Im}}(\cdot)$ are the real and imaginary parts of the CNFS complex-valued function, respectively; $\mathbf{H}(t)$ is the input vector to the CNFS; $\mathbf{W}$ denotes the set of the CNFS parameters, including the subset of the premise parameters (denoted by $\mathbf{W}_{\mathrm{If}}$) and the subset of the consequent parameters (denoted by $\mathbf{W}_{\mathrm{Then}}$). The set of the CNFS parameters is expressed by

$$\mathbf{W} = \mathbf{W}_{\mathrm{If}} \cup \mathbf{W}_{\mathrm{Then}}. \tag{11}$$

3. Hybrid learning

Based on the principle of divide-and-conquer, we spiritually separate the parameters of the CNFS into two smaller sets of parameters: the premise parameters (or the If-part parameters) and the consequent parameters (or the Then-part parameters). For fast learning, we devised the HMSPSO-RLSE hybrid learning method, where HMSPSO is used to update the premise parameters ($\mathbf{W}_{\text{If}}$) and the RLSE is used to update the consequent parameters ($\mathbf{W}_{\text{Then}}$).

HMSPSO is based on the method of particle swarm optimization (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995), motivated by the food searching behavior of bird flocking or fish schooling.

There are many particles in a PSO swarm. For a particle, the best location during the search process is denoted by **pbest**. The particles in the swarm compete with one another to become the best particle, whose location is denoted by **gbest**. Although PSO is a superb global search method, it converges easily towards the **gbest** in the first few iterations of the search process, and thus sinks into the trap of a local optimum (Niu *et al.*, 2007).

Several approaches in the literature have been proposed to improve the easily-trapped problem at the local minimum by the original PSO and its variants (Yuhui and Eberhart, 2001; Mansour *et al.*, 2007; Niu *et al.*, 2007), for example, through increasing the diversity of the population (Niu *et al.*, 2007) or via adjusting the parameters of PSO (Yuhui and Eberhart, 2001). In this paper, we propose HMSPSO to be a new multi-swarm-based PSO method that involves multiple PSO swarms. HMSPSO is with a multi-level hierarchical architecture to balance both the independent search by each swarm and the cooperative search among the swarms. Suppose that HMSPSO involves *m* particles in total, which comprise a battalion with an *n*-level structure. At the bottom of the multi-level structure, we create several swarms with the m particles. For each of these swarms, we can select the best in the swarm, and thus create several swarm-best particles, which form the first level. Similarly, the second level can be created. The process goes on until the battalion-best particle is placed on the top of the multi-level structure. In general, the proposed HMSPSO is described by the following equations:

$$\begin{aligned}\mathbf{V}_i(k+1) &\\ = w\mathbf{V}_i(k) &+ c_0\zeta_0(\mathbf{pbest}_i(k) - \mathbf{L}_i(k))\\ &+ \sum_{j=1}^{n}\sum_{q=1}^{r(j)} c_j\zeta_j(\mathbf{gbest}_{j,q}(k) - \mathbf{L}_i(k)),\end{aligned} \tag{12a}$$

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

where $\mathbf{V}_i(k) = [v_{i,1}(k)\ v_{i,2}(k)\cdots v_{i,Q}(k)]^{\mathrm{T}}$ is the velocity of the i-th particle in the k-th iteration for $i = 1,2,\ldots,m$; $\mathbf{L}_i(k)) = [l_{i,1}(k)\ l_{i,2}(k)\ldots l_{i,Q}(k)]^{\mathrm{T}}$ is the location of the i-th particle in the k-th iteration; w is the inertia weight; $\{c_j,\ j = 0,1,\ldots,n\}$ are the acceleration factors; $\{\zeta_j, j = 0,1,\ldots,n\}$ are random numbers between 0 and 1; $\mathbf{pbest}_i$ indicates the best position of the i-th particle during the search process; $\mathbf{gbest}_{j,q}$ indicates the best in the q-th PSO swarm on the j-th level for $j = 1,2,\ldots,n$; $r(j)$ indicates the number of swarms on the j-th level of the PSO battalion. Note that the battalion best is in the set of $\{\mathbf{gbest}_{n,q}(k),\ q = 1,2,\ldots,r(n)\}$ and so it is already involved in (12a).

To update the consequent parameters of the CNFS, we apply the well-known RLSE method, which is good at optimization of linear models. The algorithm of RLSE stems from the Least Squares Estimation (LSE) problem, whose model is

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

where y is the target; u is the input to the model; $\{f_i(u),\ i = 1,2,\ldots,m\}$ are known functions of u; $\{\theta_i,\ i = 1,2,\ldots,m\}$ are the model parameters to be estimated; ε is the model error. Note that the parameters $\{\theta_i,\ i = 1,2,\ldots,m\}$ can be viewed as the consequent parameters of the CNFS. To model an unknown target system, samples for the input-output behavior of the unknown system are collected to be the Training Data (TD), denoted by

$$\mathrm{TD} = \{(\mu_i, y_i), i = 1,2,\ldots,N\}, \tag{14}$$

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

$$\mathbf{y} = \mathbf{A}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \tag{15}$$

where $\boldsymbol{\theta} = [\theta_1\ \theta_2\ldots\theta_m]^{\mathrm{T}}$; $\mathbf{y} = [y_1\ y_2\ \ldots y_N]^{\mathrm{T}}$; $\boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2\ \ldots\varepsilon_N]^{\mathrm{T}}$; $\mathbf{A}$ is the matrix formed by $\{f_i(u_j),\ i = 1,2,\ldots,m$ and $j = 1,2,\ldots,N\}$. The optimal estimator for $\boldsymbol{\theta}$ can be obtained recursively by the following RLSE equations:

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

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

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

For parameter learning, the proposed CNFS is trained by the HMSPSO-RLSE hybrid learning method iteratively. In general, the training process goes on until any of stopping conditions is satisfied. Such stopping conditions can include the accuracy of performance better than some preset threshold and the amount of learning iterations used up. We only consider the latter as the termination condition, because the former is impractical to set in this study. The training procedure is given as follows.

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

Step 2. Update the premise parameters by HMSPSO.

Step 3. Update the consequent parameters by the RLSE, in which the row vector **b** and the vector $\boldsymbol{\theta}$ are arranged as follows. Note that $\lambda^{(i)}$ in (18) can be obtained by (6).

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

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

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

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

Step 4. Calculate the CNFS output.

Step 5. Calculate the cost in the MSE, defined as

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

Note that we involve only the real-part of the CNFS output in (21), because the time series forecasting problem is in the real-valued domain.

Step 6. Compare the costs from all the HMSPSO particles. Update **pbest** and **gbest** in the multiple swarms. If any stopping condition is satisfied, stop the algorithm and the battalion best is used for the optimal premise parameters of the CNFS. Otherwise, go back to Step 2 and continue the procedure.

4. Experiments

In the section, we use three examples of financial time series to test the proposed approach, whose results are compared with those other approaches (Lu *et al.*, 2009; Tung and Quek, 2011).

Example 1. (*Time series of the Nikkei 225 Index*) The Nikkei 225 (N225) Index is the most generally quoted average of Japanese equities, similar to the Dow Jones Industrial Average in the United States. N225 is the most frequently used and reliable indicator to investigate fluctuation of stocks in the Japanese market. From 4 October 1999 to 30 September 2004, the 1227 observations with the daily opening-price index of N225 were collected (Yahoo Finance for Nikkei 225 Index, 2011). The range of the dataset was normalized to the interval $[-1, 1]$, as was done by Lu *et al.* (2009).

Table 1. Settings of the HMSPSO-RLSE method.

HMSPSO	
Dimensions of PSO particles	18
Number of swarms	3
Number of particles per swarm	100
Number of levels	1
Particle velocity initialization	Random in $[0, 1]^{18}$
Particle position initialization	Random in $[0, 1]^{18}$
Acceleration factors, (c_0, c_1, c_2, c_3)	(2, 2, 2, 2)
Inertia weight, *w*	0.8
Maximum number of iterations	300
RLSE	
Number of consequent parameters	27
$\boldsymbol{\theta}_0$	27×1 zero vector
$\mathbf{P}_0$	$\alpha\mathbf{I}$
α	10^8
$\mathbf{I}$	27×27 identity matrix

Table 2. If-part parameters of the CNFS after learning (Nikkei 225 Index).

Fuzzy set		m	σ	λ
x_1	$A_{1,1}(h_1)$	0.2522	0.9198	0.1148
	$A_{1,2}(h_1)$	0.9365	0.0582	0.2597
	$A_{1,3}(h_1)$	0.8811	0.0480	0.8046
x_2	$A_{2,1}(h_2)$	0.6727	0.1220	0.4751
	$A_{2,2}(h_2)$	0.8583	0.1914	0.5930
	$A_{2,3}(h_2)$	0.4030	0.3710	0.6384

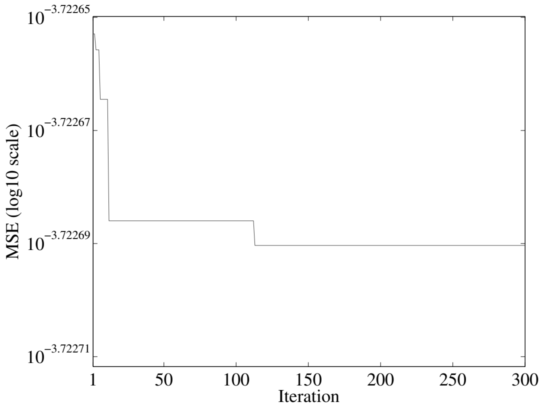


Fig. 2. Learning curve of the proposed CNFS by the HMSPSO-RLSE (Nikkei 225 Index.)

After the normalization, the dataset is denoted by $\{y(t),\ t = 1, 2, \ldots, 1227\}$, where t is the time index. For the preparation of training data, the dataset is arranged into the form of (input, target) for 1225 data pairs, which are denoted by $\{(\mathbf{H}(i), d(i)),\ i = 1, 2, \ldots, 1225\}$, where $\mathbf{H}(i) = [y(t-1)\ y(t)]^{\mathrm{T}}$; $d(i) = y(t+1)$; $t = i+1$. Note that $\mathbf{H}(i)$ is the input vector to the CNFS prediction model and $d(i)$ is the corresponding target. The first 850 data pairs were used for training and the rest for testing. For the proposed CNFS, we designed 9 T–S fuzzy rules, whose premises are characterized by complex Gaussian fuzzy sets. There are 18 premise parameters and 27 consequent parameters. For the parameter learning of the CNFS, we applied the HMSPSO-RLSE hybrid learning method, in which the premise parameters are tuned by HMSPSO and the consequent parameters are updated by the RLSE.

The settings for the HMSPSO-RLSE method are given in Table 1. The cost function was designed with the MSE. The learning curve for the proposed CNFS model is shown in Fig. 2, and the If-part and Then-part parameters of the CNFS after learning are listed in Tables 2 and 3, respectively. After learning, the prediction response in the real range and the prediction error that is the difference between the actual index and its forecast by the proposed CNFS are shown in Figs. 3(a) and 3(b), respectively. For performance comparison, two indices are used: the Root Mean Square Error (RMSE) and Mean Absolute Difference (MAD). The RMSE is the square root of the MSE and the MAD is given below:

$$\mathrm{MAD} = \frac{1}{N}\sum_{t=1}^{N} |y(t) - \mathrm{Re}(\xi(t))|. \tag{22}$$

In terms of performance the proposed approach is compared with other approaches (Lu *et al.*, 2009). The performance comparison is shown in Table 4, where the proposed approach shows superior performance to the other approaches. Moreover, in Table 4, we also compare the proposed approach to the CNFS and its neuro-fuzzy system counterpart which is designed with ordinary Gaussian fuzzy sets in (2b). Both of them were trained by the PSO-RLSE method, which is a hybrid method combining a single-swarm PSO algorithm and the RLSE algorithm (Li and Chiang, 2011b). ♦

Example 2. (*Time series of TAIEX*) In the second example, we used the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX), which is a stock market index for companies traded on the TaiWan Stock Exchange (TWSE). From 2 January 2003 to 27 February 2006, the daily closing-price data of TAIEX were collected (Yahoo Finance for Taiwan Stock Exchange Capitalization Weighted Stock Index, 2011). There are 781 samples in total. They were normalized into the range of $[-1, 1]$, denoted by $\{y(t),\ t = 1, 2, \ldots, 781\}$, where t is the time index. Then, these data were arranged in the form of (input, target) for 780 data pairs, which are denoted as $\{(\mathbf{H}(i), d(i)),\ i = 1, 2, \ldots, 780\}$, where $\mathbf{H}(i) = [y(t-1)\ y(t)]^T$; $d(i) = y(t+1)$; $t = i+1$. The first 546 data pairs were used for training and the rest for testing.

For the proposed CNFS prediction model, nine first-order T–S fuzzy If-Then rules with two inputs and one output were designed. Each input has three complex Gaussian fuzzy sets. The cost function was designed with the MSE. The HMSPSO-RLSE method, whose settings are given in Table 1, was used for the parameter learning of the CNFS. The If-part and Then-part parameters of the CNFS after learning are listed in Tables 5 and 6, respectively. The prediction response and the prediction error by the proposed CNFS are shown in Figs. 4(a) and 4(b), respectively. The learning curve for the CNFS is shown in Fig. 5.

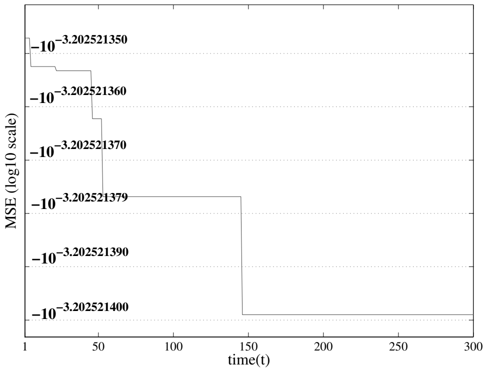


Fig. 5. Learning curve of the CNFS (TAIEX Index).

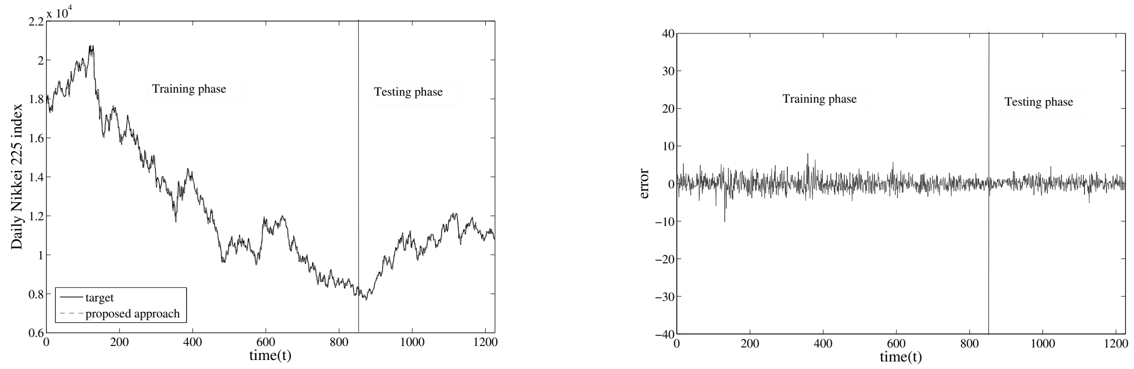
The proposed approach is compared with other approaches. It is also compared with the CNFS that was trained by the PSO-RLSE method and its NFS counterpart that was also trained by the PSO-RLSE method. The performance comparison in the RMSE and the MAD is shown in Table 7, in which the proposed approach outperforms the others (Lu *et al.*, 2009). Through the experimental results, the proposed approach has shown remarkable performance for the forecasting of TAIEX. ♦

Example 3. (*Volatility time series of the daily Hang Seng Index*) The Hang Seng Index (HSI) is the main indicator of the overall market performance in Hong Kong and it covers about 63% of the capitalization of the Hong Kong Stock Market. In this example, we test the proposed approach against the volatility of the HSI. For the five-year span from 3 January 2002 to 29 December 2006, the daily closing-price data of the HSI were collected (Yahoo Finance for Hang Seng Index, 2011). There are 1241

Table 3. Then-part parameters of the CNFS after learning (Nikkei 225 Index).

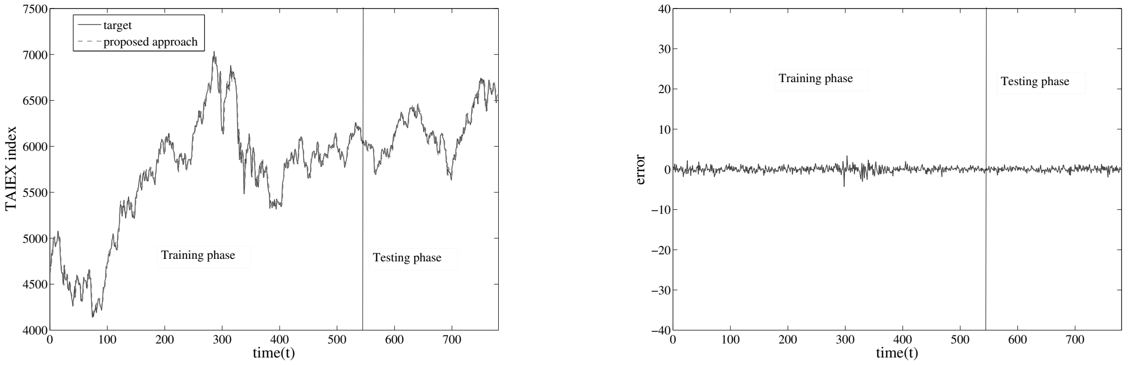
Rule no.	a_0	a_1	a_2
1	$0.0098 + 0.0159j$	$-0.0875 + 0.0068j$	$1.0688 - 0.0303j$
2	$0.0084 - 0.0148j$	$-0.0366 + 0.0187j$	$1.0224 - 0.0023j$
3	$0.0002 + 0.0002j$	$-0.0776 + 0.0040j$	$1.0745 - 0.0048j$
4	$-6.8106 - 6.3199j$	$1.0620 - 0.9738j$	$7.0414 + 8.7280j$
5	$6.2642 - 0.6098j$	$0.0151 - 0.2226j$	$-5.7874 + 1.8436j$
6	$16.5681 + 1.1824j$	$-0.8273 + 1.7113j$	$-16.3044 - 5.7277j$
7	$0.8021 - 3.2256j$	$-2.0056 - 0.2184j$	$1.8280 + 3.8855j$
8	$1.4662 + 1.3968j$	$-0.0598 + 0.3556j$	$-0.7867 - 1.6339j$
9	$2.5999 + 5.8397j$	$1.1766 - 1.7399j$	$-2.2758 - 5.3130j$

Note that after learning the Then-part parameters become complex-valued.



(a) (b)

Fig. 3. Prediction response by the CNFS with the HMSPSO-RLSE method (a), prediction error (Nikkei 225 Index) (b).



(a) (b)

Fig. 4. Prediction response by the CNFS with the HMSPSO-RLSE method (a), prediction error (TAIEX Index) (b).

trading days in total. The daily stock returns are generated by taking the logarithmic difference of the daily stock index, expressed as follows:

$$R(t) = \ln(P(t)) - \ln(P(t-1)) \times 100\%, \qquad (23)$$

where t is the time index; P is the daily closing-price index; R is the daily stock return. The daily stock returns of the HSI are shown in Fig. 6.

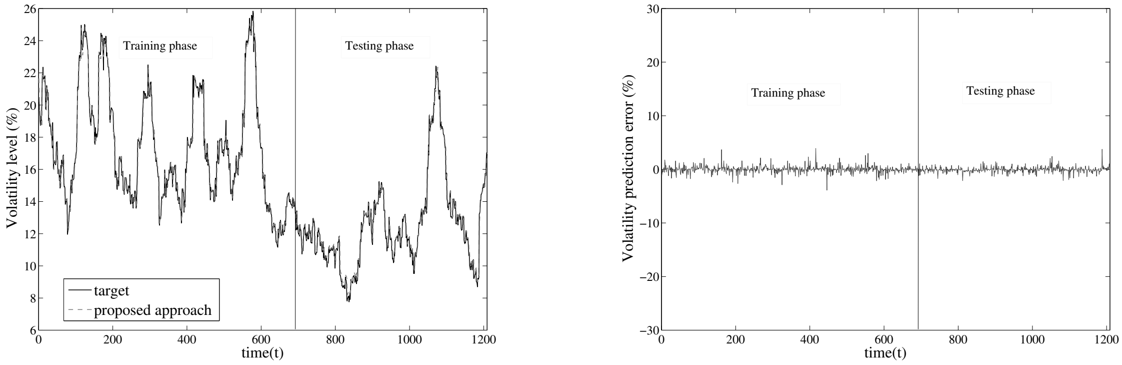
Then, with the daily stock returns, the Historical Volatility (HV) is given as

$$\mathrm{HV}(t) = \sqrt{\frac{\sum_{t=1}^{N} R(t) - \bar{R}}{N-1}} \times \sqrt{T} \times 100\%, \qquad (24)$$

where $\bar{R}$ denotes the average daily stock returns of the past N trading days; T is a normalizing factor which transfers the daily historical standard deviation into a yearly one. In this example, the 30-day ($N = 30$) historical standard deviation and the annualized normalizing factor with $T =$

Table 4. Then-part parameters of the CNFS after learning (Nikkei 225 Index).

Rule no.	a_0	a_1	a_2
1	$-0.6113 + 0.3605j$	$0.6884 + 1.2963j$	$1.1378 - 2.3062j$
2	$0.4360 - 0.3342j$	$-0.5600 - 1.1051j$	$1.1383 + 3.8616j$
3	$-12.6125 - 3.4865j$	$13.5670 - 11.1452j$	$-3.4185 + 8.6284j$
4	$2.0269 + 1.2375j$	$-0.3034 - 1.7860j$	$-1.4133 + 0.9111j$
5	$-1.4655 - 1.2695j$	$-0.5814 + 1.6826j$	$2.8811 - 5.0030j$
6	$15.0455 - 14.8307j$	$-8.7080 + 44.4367j$	$-1.2253 - 11.7603j$
7	$-0.3636 - 0.1676j$	$0.1200 + 0.1723j$	$1.2466 + 0.0380j$
8	$-0.3077 + 0.7500j$	$0.5477 - 2.1092j$	$1.8564 + 1.3990j$
9	$1.9702 + 6.7749j$	$1.7707 - 12.6058j$	$-1.2757 - 2.8104j$
Note that after learning the Then-part parameters become complex-valued.			



(a) (b)

Fig. 7. Prediction response by the CNFS with the HMSPSO-RLSE method (a), prediction error (Hang Seng Index) (b).

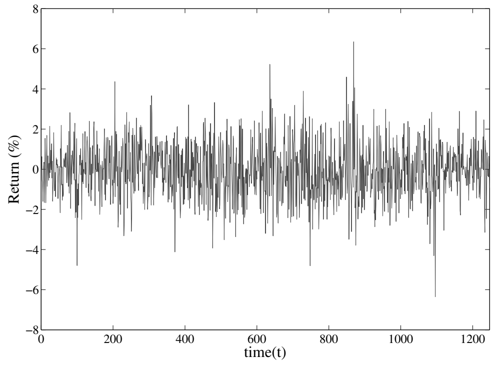


Fig. 6. Daily returns of the Hang Seng Index.

Table 5. If-part parameters of the CNFS after learning (TAIEX Index).

Fuzzy set		m	σ	λ
x_1	$A_{1,1}(h_1)$	0.4098	0.6756	0.2041
	$A_{1,2}(h_1)$	0.5692	0.4346	0.7296
	$A_{1,3}(h_1)$	0.7253	0.1628	0.9802
x_2	$A_{2,1}(h_2)$	0.5374	0.9876	0.2041
	$A_{2,2}(h_2)$	0.0359	0.3464	0.2265
	$A_{2,3}(h_2)$	0.3782	0.0591	0.8105

Table 6. Performance comparison (TAIEX Index).

Method	RMSE	MAD
Random walk (Lu *et al.*, 2009)	53.21	39.88
SVR (Lu *et al.*, 2009)	46.60	34.63
ICA-SVR (Lu *et al.*, 2009)	41.09	31.70
NFS (by the PSO-RLSE)	7.46	5.05
CNFS (by the PSO-RLSE)	5.45	4.20
CNFS (by the HMSPSO-RLSE) (proposed)	5.39	4.17

252 were set in order to make a comparison with the work of Tung and Quek (2011). There are 1211 volatility data, which are calculated from the original 1241 daily trading observations. These volatility data were then normalized into the range of $[0, 1]$, denoted as $y(t), t = 1, 2, \ldots 1211$, where t indicates the time index. The volatility data of the first three years (Jan 2002 to Dec 2004) were used for training and the data of the remaining two years (Jan 2005 to Dec 2006) were used for testing. The data were arranged in the form of (input, target) for 1209 data pairs, which are denoted by $\{(\mathbf{H}(i),\ d(i)),\ i = 1, 2, \ldots, 780\}$, where $\mathbf{H}(i) = [y(t-1)\ y(t)]^T$; $d(i) = y(t+1)$; $t = i + 1$. For the proposed CNFS model, nine first-order T–S fuzzy If-Then rules with two inputs and one output were

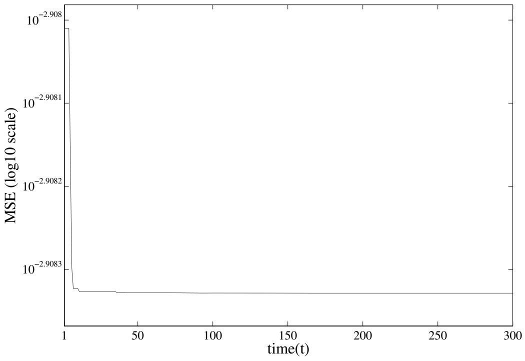


Fig. 8. Learning curve of the CNFS (Hang Seng Index).

designed. Each input has three complex Gaussian fuzzy sets.

The cost function was designed with the MSE. For parameter learning, the HMSPSO-RLSE method was used to update the free parameters of the CNFS. The settings for the HMSPSO-RLSE are the same as in Table 1. After learning, the prediction response for the volatility of HSI and the predictive error are shown in Figs. 7(a) and 7(b), respectively. The learning curve of the proposed model is shown in Fig. 8. The If-part and Then-part parameters of the CNFS after learning are listed in Tables 8 and 9, respectively. In terms of performance the proposed CNFS approach was compared with other approaches in the literature: the back-propagation neural network FFNN-BP, the Radial Basis Function (RBF) network (Moody and Darken, 1989), the Cerebellar Model Articulation Controller (CMAC) (Albus, 1975) neural network, HyFIS (Kim and Kasabov, 1999), the ANFIS (Jang, 1993), and the evolving fuzzy semantic memory (Tung and Quek, 2011).

The models that are compared with the proposed approach in this example were implemented by Tung and Quek (2011). Three performance indices are used, including the MSE, Average Relative Variance (ARV), and the Mean Absolute Percentage Error (MAPE). The ARV and the MAPE are described below:

Table 7. Performance comparison (Nikkei 225 Index).

Method	RMSE	MAD
Random walk (Lu *et al.*, 2009)	137.85	105.77
SVR (Lu *et al.*, 2009)	60.53	43.71
ICA-SVR (Lu *et al.*, 2009)	56.76	40.86
NFS (by the PSO-RLSE)	14.65	11.18
CNFS (by the PSO-RLSE)	13.25	10.52
CNFS (by the HMSPSO-RLSE) (proposed)	13.12	10.19

Table 8. If-part parameters of the CNFS after learning (Hang Seng Index).

Fuzzy set		m	σ	λ
x_1	$A_{1,1}(h_1)$	0.7076	0.1559	0.0117
	$A_{1,2}(h_1)$	0.5455	0.0231	0.7770
	$A_{1,3}(h_1)$	0.3126	0.9272	0.6648
x_2	$A_{2,1}(h_2)$	0.3236	0.6395	0.2252
	$A_{2,2}(h_2)$	0.7654	0.5014	0.9305
	$A_{2,3}(h_2)$	0.8082	0.0643	0.8986

$$\mathrm{ARV}(t) = \frac{\sum\limits_{t=1}^{N} y(t) - \mathrm{Re}(\xi(t))^2}{\sum\limits_{t=1}^{N} (y(t) - \bar{y})^2}, \tag{25}$$

$$\mathrm{MAPE}(t) = \frac{1}{N} \sum_{t=1}^{N} \left| \frac{y(t) - \mathrm{Re}(\xi(t))}{y(t)} \right| \times 100\%, \tag{26}$$

where $\bar{y} = (1/N)\sum_{t=1}^{N} y(t)$. The performance comparison is shown in Table 10, where the proposed approach shows much better performance than the other approaches (Tung and Quek, 2011) in the testing phase. The proposed approach is also compared with the CNFS that was trained by the PSO-RLSE method and the NFS that was also trained by the PSO-RLSE method. As shown in Table 10, the proposed approach outperforms the CNFS trained by the PSO-RLSE and the NFS.

5. Discussion

A novel complex fuzzy computing paradigm applying complex fuzzy sets to the problem of time series forecasting has been presented in this paper. With the synergistic merits of complex fuzzy inference and neural-network based adaptive capability, the CNFS has shown outstanding nonlinear forecasting capability through the experimental results. Based on the divide-and-conquer concept, the problem space is spiritually divided into two smaller spaces: the If-part subspace and the Then-part subspace. For the purpose of fast learning to fine tune the CNFS, we have devised the HMSPSO-RLSE hybrid learning method, which has been successfully applied to the problem of time series forecasting with three examples of financial time series.

The proposed approach has shown good results in experiments for both the learning convergence (as shown in Figs. 2, 5 and 8) and the prediction performance (as shown in Tables 4, 7 and 10). The experimental results show that the proposed approach outperforms the compared approaches in the literature (Lu *et al.*, 2009; Tung and Quek, 2011). For instance, for Example 1 (time series of the Nikkei 225 Index), as shown in Table 4, the proposed approach has the prediction performance 13.12

Table 9. Then-part parameters of the CNFS after learning (Hang Seng Index).

Rule no.	a_0	a_1	a_2
1	$-0.2518 + 0.2199j$	$-0.2155 + 0.8574j$	$1.4812 - 0.7546j$
2	$-0.6646 - 0.3928j$	$-1.0824 - 0.1470j$	$2.9048 + 0.2852j$
3	$-0.4580 + 3.8483j$	$0.6791 + 0.1423j$	$0.2826 - 4.5508j$
4	$13.1515 + 1.8947j$	$2.8715 + 7.8316j$	$-9.1240 - 0.1960j$
5	$-14.9904 + 24.8028j$	$-8.4406 - 2.7863j$	$16.7598 - 37.0894j$
6	$46.7548 - 36.3650j$	$-121.90 - 1.0158j$	$28.1925 + 51.1853j$
7	$0.1518 + 0.0656j$	$-0.0109 - 0.6614j$	$0.8988 + 0.8336j$
8	$-0.3380 + 0.1717j$	$1.1900 + 0.3128j$	$0.1590 - 0.6342j$
9	$2.4196 - 2.9736j$	$-1.4731 - 0.4107j$	$0.0583 + 3.8774j$
Note that after learning the Then-part parameters become complex-valued.			

Table 10. Performance comparison (volatility of the Hang Seng Index).

Method	Rules	Training phase			Testing phase		
		MSE	ARV	MAPE	MSE	ARV	MAPE
FFNN-BP (Tung and Quek, 2011)	20	6.287	0.644	8.145	4.351	0.331	9.305
RBF (Tung and Quek, 2011)	40	0.718	0.074	3.601	3.780	0.288	13.424
CMAC (Tung and Quek, 2011)	93036	0.387	0.042	2.112	12.539	0.474	37.212
HyFIS (Tung and Quek, 2011)	154	0.734	0.075	3.476	3.971	0.302	14.114
ANFIS (Tung and Quek, 2011)	11	0.351	0.036	2.298	5.371	0.409	14.725
eFSM (Tung and Quek, 2011)	97	0.699	0.072	3.343	1.059	0.081	4.989
NFS (by the PSO-RLSE)	9	0.405	0.031	1.784	0.279	0.036	1.355
CNFS (by the PSO-RLSE)	9	0.404	0.031	1.765	0.262	0.033	1.280
CNFS (by the HMSPSO-RLSE) (proposed)	9	0.380	0.029	1.756	0.246	0.032	1.273

in the RMSE for the testing phase. Such performance is 90% better than that of the random walk (Lu *et al.*, 2009), which is 137.85 in the RMSE, 78% better than that of the SVR (Lu *et al.*, 2009), which is 60.53 in the RMSE, 76% better than that of the ICA-SVR (Lu *et al.*, 2009), which is 56.76 in the RMSE, and 10% better than that of the NFS, which is 14.65 in the RMSE.

For Example 2 (time series of the TAIEX Index), as shown in Table 7, the proposed approach has the prediction performance 5.39 in the RMSE for the testing phase. This performance is 89% better than that of the random walk (Lu *et al.*, 2009), which is 53.21 in the RMSE, 88% better than that of the SVR (Lu *et al.*, 2009), which is 46.60 in the RMSE, 86% better than that of the ICA-SVR (Lu *et al.*, 2009), which is 41.09 in the RMSE, and 27% better than that of the NFS, which is 7.46 in the RMSE.

For Example 3 (volatility time series of the daily Hang Heng Index), as shown in Table 10, the proposed approach has the prediction performance 0.246 in the MSE for the testing phase. This performance is 94% better than that of the FFNN-BP (Tung and Quek, 2011), which is 4.351 in the MSE, 93% better than that of the RBF (Tung and Quek, 2011), which is 3.780 in the MSE, 97% better than that of the CMAC (Tung and Quek, 2011), which is 12.539 in the MSE, 93% better than that of the HyFIS (Tung and Quek, 2011), which is 3.971 in the MSE, 95% better than that of the ANFIS (Tung and Quek, 2011), which is 5.371 in the MSE, 76% better than that of the eFSM (Tung and Quek, 2011), which is 1.059 in the MSE, and 11% better than that of the NFS, which is 0.279 in the MSE.

We think the excellent performance of the proposed approach mainly comes from two factors. First, the complex-valued membership property of CFSs can enrich and augment the functional mapping ability of the CNFS for accurate forecasting. Second, the hybrid learning method can successfully search for the optimal or near-optimal solution quickly. Moreover, for the HMSPSO-RLSE hybrid learning method, we have presented HMSPSO as a new multi-swarm-based scheme for optimization. It can help avoid being trapped into a local minimum and so enhance the search ability for the optimal solution. Through the combination of HMSPSO and the RLSE, the hybrid learning method has shown very successful results. For instance, in Example 3, where the time series shows more fierce fluctuation than those in the other examples, the CNFS trained by the HMSPSO-RLSE hybrid learning method has the prediction performance of 0.246 in the MSE in testing phase, while the CNFS trained by the PSO-RLSE has 0.279 in the MSE, as shown in Table 10. With the same CNFS, the HMSPSO-RLSE shows better performance than the PSO-RLSE hybrid learning method. For ARV and the MAPE in Table 10, we observed similar results. This has illustrated that the proposed HMSPSO-RLSE hybrid learning method can efficiently increase forecasting performance.

6. Conclusion

We conclude the paper by describing two main advantages. First, the proposed CNFS approach using CFSs has been successfully presented as a new computing paradigm with good adaptation for modeling and forecasting. With the practical implementation of complex fuzzy sets into the CNFS, we have opened a new window to intelligent-system based research and applications. Second, the novel HMSPSO-RLSE hybrid learning method for fast learning has been demonstrated successfully with three examples of financial time series forecasting, where the proposed approach has shown superior performance compared to other approaches. The experimental results show that the proposed approach outperforms the compared approaches. For future work, the computing approach with complex fuzzy sets can be applied to other areas, such as modeling, controls, classification, and signal processing.

Acknowledgment

This research work is supported by the National Science Council, Taiwan (ROC), under the grant contract no. NSC99-2221-E-008-088. The authors are grateful to the anonymous reviewers for their valuable comments.

References

Albus, J.S. (1975). Data storage in the cerebellar model articulation controller (CMAC), *Journal of Dynamic Systems, Measurement and Control* **97**(2): 228–233.

Boyacioglu, M.A. and Avci, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange, *Expert Systems with Applications* **37**(12): 7908–7912.

Brdyś, M.A., Borowa, A., Idźkowiak, P. and Brdyś, M.T. (2009). Adaptive prediction of stock exchange indices by state space wavelet networks, *International Journal of Applied Mathematics and Computer Science* **19**(2): 337–348, DOI: 10.2478/v10006-009-0029-z.

Buckley, J.J. (1989). Fuzzy complex numbers, *Fuzzy Sets and Systems* **33**(3): 333–345.

Castro, J.L. (1995). Fuzzy logic controllers are universal approximators, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **25**(4): 629–635.

Chen, Z., Aghakhani, S., Man, J. and Dick, S. (2011). ANCFIS: A neurofuzzy architecture employing complex fuzzy sets, *IEEE Transactions on Fuzzy Systems* **19**(2): 305–322.

Deng, X. and Wang, X. (2009). Incremental learning of dynamic fuzzy neural networks for accurate system modeling, *Fuzzy Sets and Systems* **160**(7): 972–987.

Dick, S. (2005). Toward complex fuzzy logic, *IEEE Transactions on Fuzzy Systems* **13**(3): 405–414.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory, *Proceedings of the 6th International Symposium on Micro Machine and Human Science, MHS 1995, Nagoya, Japan*, pp. 39–43.

Gao, Y. and Er, M.J. (2005). Narmax time series model prediction: Feedforward and recurrent fuzzy neural network approaches, *Fuzzy Sets and Systems* **150**(2): 331–350.

Graves, D. and Pedrycz, W. (2009). Fuzzy prediction architecture using recurrent neural networks, *Neurocomputing* **72**(7–9): 1668–1678.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators, *Neural Networks* **2**(5): 359–366.

Jang, J.S.R. (1993). ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man, and Cybernetics* **23**(3): 665–685.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, *IEEE International Conference on Neural Networks, Perth, Australia*, pp. 1942–1948.

Khashei, M. and Bijari, M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, *Applied Soft Computing* **11**(2): 2664–2675.

Kim, J. and Kasabov, N. (1999). HYFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems, *Neural Networks* **12**(9): 1301–1319.

Li, C. and Cheng, H.-H. (2011). Intelligent forecasting of S&P 500 time series—A self-organizing fuzzy approach, *in* N.T. Nguyen, C.-G. Kim and A. Janiak (Eds.), *Intelligent Information and Database Systems*, Lecture Notes in Artificial Intelligence, Vol. 6592, Springer-Verlag, Berlin/Heidelberg, pp. 411–420.

Li, C. and Chiang, T.-W. (2011a). Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach, *in* N.T. Nguyen, C.-G. Kim and A. Janiak (Eds.) *Intelligent Information and Database Systems*, Lecture Notes in Artificial Intelligence, Vol. 6592, Springer-Verlag, Berlin/Heidelberg, pp. 242–251.

Li, C. and Chiang, T.-W. (2011b). Complex fuzzy model with PSO-RLSE hybrid learning approach to function approximation, *International Journal of Intelligent Information and Database Systems* **5**(4): 409–430.

Li, C. and Chiang, T.-W. (2011c). Function approximation with complex neuro-fuzzy system using complex fuzzy sets—A new approach, *New Generation Computing* **29**(3): 261–276.

Li, C., Chiang, T.-W., J.-W., H. and Wu, T. (2010). Complex neuro-fuzzy intelligent approach to function approximation, *3rd International Workshop on Advanced Computational Intelligence, IWACI 2010, Suzhou, China*, pp. 151–156.

Li, C. and Lee, C.-Y. (2003). Self-organizing neuro-fuzzy system for control of unknown plants, *IEEE Transactions on Fuzzy Systems* **11**(1): 135–150.

Li, C., Lee, C.-Y. and Cheng, K.-H. (2004). Pseudoerror-based self-organizing neuro-fuzzy system, *IEEE Transactions on Fuzzy Systems* **12**(6): 812–819.

Li, C. and Priemer, R. (1997). Self-learning general purpose PID controller, *Journal of the Franklin Institute* **334**(2): 167–189.

Li, C. and Priemer, R. (1999). Fuzzy control of unknown multiple-input-multiple-output plants, *Fuzzy Sets and Systems* **104**(2): 245–267.

Lu, C.-J., Lee, T.-S. and Chiu, C.-C. (2009). Financial time series forecasting using independent component analysis and support vector regression, *Decision Support Systems* **47**(2): 115–125.

Man, J.Y., Chen, Z. and Dick, S. (2007). Towards inductive learning of complex fuzzy inference systems, *Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 2007, San Diego, CA, USA*, pp. 415–420.

Mansour, M.M., Mekhamer, S.F. and El-Kharbawe, N.-S. (2007). A modified particle swarm optimizer for the coordination of directional overcurrent relays, *IEEE Transactions on Power Delivery* **22**(3): 1400–1410.

Moody, J. and Darken, C.J. (1989). Fast learning in networks of locally-tuned processing units, *Neural Computation* **1**(2): 281–294.

Moses, D., Degani, O., Teodorescu, H.N., Friedman, M. and Kandel, A. (1999). Linguistic coordinate transformations for complex fuzzy sets, *IEEE International Fuzzy Systems Conference Proceedings, FUZZ-IEEE 1999, Seoul, Korea*, pp. 1340–1345.

Mousavi, S.J., Ponnambalam, K. and Karray, F. (2007). Inferring operating rules for reservoir operations using fuzzy regression and ANFIS, *Fuzzy Sets and Systems* **158**(10): 1064–1082.

Niu, B., Zhu, Y., He, X. and Wu, H. (2007). MCPSO: A multi-swarm cooperative particle swarm optimizer, *Applied Mathematics and Computation* **185**(2): 1050–1062.

Paul, S. and Kumar, S. (2002). Subsethood-product fuzzy neural inference system (SUPFUNIS), *IEEE Transactions on Neural Networks* **13**(3): 578–599.

Ramot, D., Friedman, M., Langholz, G. and Kandel, A. (2003). Complex fuzzy logic, *IEEE Transactions on Fuzzy Systems* **11**(4): 450–461.

Ramot, D., Milo, R., Friedman, M. and Kandel, A. (2002). Complex fuzzy sets, *IEEE Transactions on Fuzzy Systems* **10**(2): 171–186.

Rojas, I., Valenzuela, O., Rojas, F., Guillen, A., Herrera, L.J., Pomares, H., Marquez, L. and Pasadas, M. (2008). Soft-computing techniques and ARMA model for time series prediction, *Neurocomputing* **71**(4–6): 519–537.

Simiński, K. (2010). Rule weights in a neuro-fuzzy system with a hierarchical domain partition, *International Journal of Applied Mathematics and Computer Science* **20**(2): 337–347, DOI: 10.2478/v10006-010-0025-3.

Smetek, M. and Trawinski, B. (2011). Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms, *New Generation Computing* **29**(3): 309–327.

Tung, W.L. and Quek, C. (2011). Financial volatility trading using a self-organising neural-fuzzy semantic network and option straddle-based approach, *Expert Systems with Applications* **38**(5): 4668–4688.

Vo, N., Quang, T., Dinh, T. and Dinh, T. (2011). Robust visual tracking using randomized forest and online appearance model, *in* N.T. Nguyen, C.-G. Kim and A. Janiak (Eds.), *Intelligent Information and Database Systems*, Lecture Notes in Artificial Intelligence, Vol. 6592, Springer-Verlag, Berlin/Heidelberg pp. 212–221.

Yahoo Finance for Hang Seng Index (2011). Website: `http://finance.yahoo.com/q?s=^HSI`.

Yahoo Finance for Nikkei 225 Index (2011). Website, `http://finance.yahoo.com/q?s=^N225`.

Yahoo Finance for Taiwan Stock Exchange Capitalization Weighted Stock Index (2011). Website, `http://finance.yahoo.com/q?s=^TWII`.

Yuhui, S. and Eberhart, R.C. (2001). Fuzzy adaptive particle swarm optimization, *Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea*, pp. 101–106.

Zhang, G., Dillon, T.S., Cai, K.-Y., Ma, J. and Lu, J. (2009). Operation properties and δ-equalities of complex fuzzy sets, *International Journal of Approximate Reasoning* **50**(8): 1227–1249.

Chunshien Li received his Ph.D. degree in electrical engineering and computer science in 1996 from the University of Illinois at Chicago, USA. He is currently an associate professor at the Department of Information Management, National Central University, Taiwan. He has been with the Department of Computer Science and Information Engineering, National University of Taiwan, and the Department of Electrical Engineering, Chang Gung University. Professor Li has also been the chief of the Information System Group of the Computer Center, and the director of the Teacher Development Center, National University of Taiwan, Taiwan. He received the Research Award from the National Science Council, Taiwan, in 1999, and the Excellent Teacher Research Award from Chang Gung University, Taiwan, in 2000. He is the author of more than 100 academic and technical publications. He has served as a referee for several international journals and as a PC member for several international academic conferences. His current research interests include computational intelligence, machine learning and optimization, neural fuzzy systems, prediction, pattern recognition, intelligent modeling and forecasting, intelligent control, and intelligent signal processing.

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

Received: 18 October 2011
Revised: 17 May 2012