Multi-Variate Timeseries Forecasting Using Complex Fuzzy Logic

Omolbanin Yazdanbakhsh
Dept. of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, Canada
yazdanba@ualberta.ca

Scott Dick
Dept. of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, Canada
sdick@ualberta.ca

***Abstract*— Complex fuzzy logic has been repeatedly used to construct very effective time-series forecasting algorithms. The great majority of these studies, however, only involve univariate time series. The only exception is one work on bivariate time series. Our objective is to investigate the network architectures and time series representations that lead to effective general multi-variate time series forecasting. Our experiments will make use of the Adaptive Neuro-Complex Fuzzy Inferential System architecture, evaluating three different approaches (single-input single-output, multiple-input single-output, and multiple-input multiple-output) on three multi-variate datasets. Our results indicate that the complex fuzzy architectures are at least as accurate as Radial Basis Function Networks and Support Vector Regression on these problems.**

Keywords—Multivariate time series forecasting; Complex fuzzy logic; neuro-fuzzy systems; machine learning.

I. INTRODUCTION

A time series is an ordered collection of repeated observations of a phenomenon over time. They are important examples of data streams, which are a major focus of the Big Data and analytics communities; one obvious example is the time series of stock quotations from financial markets. These particular time series chart the rise and fall in value of tens of trillions of dollars' worth of assets. Forecasting the future evolution of a time series (e.g. by predicting the future movement of the stock market) is thus a valuable and well-recognized problem in statistics and machine learning. Various statistical and machine learning algorithms have been proposed for the prediction of univariate time series (time series containing observations of a single quantity, e.g. [5, 9, 41, 46, 47]). Multivariate time series analysis is a generalization wherein multiple quantities are simultaneously observed. It is more challenging than univariate time series analysis, as relations between the different variates need to be considered. Again, numerous statistical and machine-learning algorithms have been developed for multivariate time series forecasting, e.g. [3, 4, 6, 12, 32, 37].

In the past decade, neuro-fuzzy systems based on Complex Fuzzy Sets and Logic (CFSL) have shown promising results in univariate time series prediction. Complex fuzzy sets (an extension of type-1 fuzzy sets wherein the codomain of the membership function is the unit disc of the complex plane) and complex fuzzy logic (an isomorphic multi-valued logic whose truth values are drawn from the unit disc) were first introduced by Ramot in [30, 31]. The Adaptive Neuro-Complex Fuzzy Inference System (ANCFIS) [8], and the family of architectures based on the Complex Neuro-Fuzzy System (CNFS) are based on ANFIS system [16], but incorporate complex fuzzy sets in their input layers, and utilize complex-valued network signals and complex fuzzy logic operators. ANCFIS was applied to univariate time series forecasting in [8, 28, 43, 44], and [21, 24-26] applied CNFS to this area. However, these systems have not previously been extended to multivariate time series forecasting; the only exception is the use of some CNFS systems for bivariate forecasting (using the so-called dual-output property) [26]. Currently, no systems based on complex fuzzy sets and logic exist for general multivariate time series forecasting.

In this paper, we report on our initial results in extending ANCFIS for multi-variate time series forecasting, focusing on three datasets (one having two variates, the others having three). We compare three different approaches: a collection of single-input single-output (SISO) ANCFIS networks, a collection of multiple-input single-output (MISO) ANCFIS networks, and a single multiple-input multiple-output (MIMO) ANCFIS network. We cross-check our results by comparing them against the well-known Radial Basis Function Network (RBFN) and Support Vector Regression (SVR) algorithms.

The remainder of the paper is organized as follow. In Section 2 we provide essential background on complex fuzzy sets and logics, the ANCFIS system, and delay embedding techniques for multivariate and univariate time series. Our experimental methodology is described in Section 3, and we discuss our experimental results in Section 4. In Section 5 we offer a summary and discussion of future work on this topic.

II. LITERATURE REVIEW

A. Complex Fuzzy Sets and Logic

Ramot et.al proposed complex fuzzy sets with complex-value membership functions of the form [31]:

$$\mu_s(x) = r_s(x).e^{(jw_s(x))}, j = \sqrt{-1} \tag{1}$$

where r_s is the magnitude and w_s is the phase of the membership grade; r_s was constrained to be ≤ 1. The first complex fuzzy logic was proposed by Ramot et.al [30], who suggested the algebraic product as a fuzzy implication. Dick [11] studied the magnitude and phase of a complex membership grade simultaneously and showed algebraic product was a candidate for the conjunction operator in complex fuzzy sets. He also suggested sinusoidal functions as a candidate for complex fuzzy membership functions, which might be able to capture periodic behaviour of phenomena. Tamir et.al [38] defined complex fuzzy sets with membership grades drawn from the unit square; these complex fuzzy sets are called “pure’ complex fuzzy sets. Pure complex fuzzy logic was studied in [38-40]. Pythagorean membership grades and complex Atanassov’s intuitionistic fuzzy sets were also proposed in [33, 42]. Different complex fuzzy operations based on Ramot’s complex fuzzy set definition were proposed in [42, 45].

B. Adaptive Neuro- Complex Fuzzy Inference System

ANCFIS, the first machine-learning realization of the complex fuzzy logic based on [11, 30] was designed in [8]. The system is based on ANFIS (Adaptive Neuro-Fuzzy Inference System), the well-known neuro-fuzzy system proposed by Jang et.al [16], with several changes. In order to mimic a complex fuzzy inference system, ANCFIS uses sinusoidal membership functions:

$$r(\theta) = d \cdot sin(a(\theta = x) + b) + c \tag{2}$$

where $r(\theta)$ is amplitude and θ is phase. The parameters $\{a, b, c, d\}$ manipulate the shape of a sine wave; a changes the frequency of the wave, b causes a phase shift, c shifts the wave vertically and d changes its amplitude. The following conditions must be satisfied for the parameters to keep the membership grade in the unit disk [8]:

$$0 \quad d + c \quad 1, \qquad 1 \quad c \quad d \quad 0 \tag{3}$$

As suggested by Dick [11], this membership function is able to capture periodic behavior of phenomena; thus, time series forecasting was the first application of ANCFIS [8, 28, 43, 44].

A second difference from ANFIS is that ANCFIS does not use the normal “lagged” representation of inputs to a time series. Rather than taking a sequence of previous observations and treating each as a measurement on orthogonal input dimensions, ANCFIS accepts a windowed segment of a time series (although this window closely relates to the lagged representation, as we discuss in Section 3). This preserves the time order and sampling rate between the observations – information that is lost in the lagged representation. The layer-1 fuzzification operation of ANCFIS is accomplished using the complex-valued convolution operator, and signals in much of the network are complex-valued.

A third difference is implemented in a new layer added before the fourth layer of the original ANFIS. In [30], Ramot et al. chose the algebraic product as a complex fuzzy implication to realize their concept of *rule interference*; the idea that, based on membership phase, the firing strengths of complex fuzzy rules may interfere constructively or destructively with each other. In ANCFIS, the dot product operation is used to implement this idea in the new fourth layer.

The ANCFIS architecture has six layers as follows [8]:

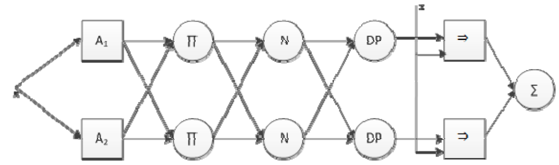

Fig. 1. ANCFIS architecture for univariate time series problems with two membership functions

- Layer 1: Membership grades are assigned to the input vectors. Each membership function, defined in (2), is sampled over one period to the length of input vector. Then, the complex convolution between the samples and the input vector is calculated. To limit the results to the unit disc, they are normalized by the Elliot function.
- Layer 2: The firing strength of each rule is calculated by the algebraic product.
- Layer 3: The firing strength of each rule is normalized.
- Layer 4: Rule interference is implemented by taking the dot product of the current rule’s firing strength and the complex sum of all other rules’ firing strengths.
- Layer 5: A linear consequent function is applied as in ANFIS, with parameters determined by a Kalman filter.
- Layer 6: All incoming signals are summed to obtain the final network output.

Following ANFIS, ANCIFS employs a hybrid learning system where in the forward pass, consequent parameters are obtained by a Kalman filter [19], and antecedent parameters are updated in the backward pass. However, while error signals are back-propagated using gradient descent until Layer 1, there is no closed-form solution of the derivative of network error with respect to the layer-1 CFS parameters. Thus, a derivative-free optimization algorithm, Variable Neighbourhood Chaotic Simulated Annealing (VNCSA) [8], is used to update the CFS parameters.

Other complex fuzzy machine-learning systems have been proposed. An online version of ANCFIS has been designed in [1]. The family of Complex Neuro-Fuzzy Systems (CNFS) architectures have been proposed in [20-27] using complex Gaussian membership functions.

C. Delay Embedding of a Time Series

Delay embeddings are a common method of creating a lagged representation of a time series. According to Taken’s Theorem [17], if enough previous observations are taken, the resulting state space is equivalent to the original state space of the system. Thus, the prediction of the time series is possible from the trajectory of the reconstructed phase space. Delay embeddings on univariate and multivariate time series differ to an extent.

1) Univariate Time Series

A delay vector of a univariate time series is given by [13]:

$$S_n = (s_{n-(m-1)\tau},\ s_{n-(m-2)\tau}, \dots, s_n) \tag{4}$$

where S_n is the delay vector with dimension of m, and τ specifies the delay between successive observations; both are determined heuristically. Candidates for τ can be the first zero of the autocorrelation function, or the first minimum of the time-delayed mutual information function [17], while dimensionality is commonly estimated by the false nearest neighbors technique [18].

2) Multivariate Time Series

A delay embedding of a multivariate time series with M variables and length of N, $X_1, X_{2,} \dots, X_N$ where $X_i = (x_{1,i}, x_{2,i}, \dots, x_{M,i})$, can be constructed as [3]:

$$V_n = \begin{array}{c} (x_{1,n}, x_{1,n-\tau_1}, \cdots, x_{1,n-(d_1-1)\tau_1}, \\ x_{2,n}, x_{2,n-\tau_2}, \cdots, x_{2,n-(d_2-1)\tau_2}, \\ \cdots \quad \cdots \quad , \\ x_{M,n}, x_{M,n-\tau_M}, \cdots, x_{M,n-(d_M-1)\tau_M}) \end{array} \tag{5}$$

where τ_i and d_i $(i = 1, 2, \dots, M)$ are the time delay and the embedding dimension of the i-th variable, respectively. To obtain the time delays, each variable of the multivariate time series is considered separately, and the same approach as the univariate time series are applied on them, mutual information and autocorrelation [2, 3, 37]. [3] proposed an approach to determine the dimensions; it works based on finding the nearest neighbor of each delay vector in a given dimension and time delay. The error between one-step ahead prediction of these points is used to specify the appropriate dimension.

III. METHODOLOGY

A. Experimental Design

We will study three alternative architectures of ANCFIS in forecasting multivariate time series. The well-known radial basis function network (RBFN) and support vector regression (SVR) algorithms are used as a cross-check on our results. The RBFN and SVR of course employ the lagged representation of the time series. The input windows in ANCFIS are also essentially the same delay vectors. Our previous work has compared multiple ways of defining input windows. The first, taking all observations within one apparent "period" of the time series worked well in [8], but can result in very large windows; we found in [43] that this leads to extremely long runtimes for ANCFIS training. We also considered using the delay parameter as the length of the input window, and finally treating delay coordinates as a down-sampled input window (this is possible because the sampling rate remains uniform and the delay vectors are time-ordered) [43]. This latter seemed best in our experiments, and so will be the approach we follow in the current paper.

We first estimate the dimension and delay of the embedding vector, following the methods summarized in Section 2. The delay of each variable is obtained individually by comparing the mutual information and autocorrelation function for it, and then manually cross-checking the phase portrait (phase portraits are a plot of the variate x(t) versus its delay x(t-n) [17]). To obtain the dimension, in case of univariate time series, we use the false nearest neighbors technique [18], and for multivariate time series, we implement Cao's approach [3] considering three nearest neighbors.

1) SISO ANCFIS

This design of ANCFIS treats each variate as an independent univariate time series. Thus, we have *N* complete and independent ANCFIS systems for an *N*-variate time series. For example, for a two-variate time series (x and y) with $\tau_1 = \tau_2 = 1$, and $d_1 = 3$ *and* $d_2 = 2$, we have the following input vectors for the two ANCFISs:

x_3, x_2, x_1, x_t	Input to the first ANCFIS system
y_2, y_1, y_t	Input to the second ANCFIS system

where x_t and y_t are the one-step-ahead targets that must be predicted. The overall performance of ANCFIS in forecasting of multivariate time series in this approach is assessed on the combined accuracy of all the systems.

2) MISO ANCFIS

This design of ANCFIS again constructs a separate ANCFIS for each variates, but the input to each is the combined delay vector for all variates; only the one-step-ahead target is different. For example, for a two-variate time series (x and y) with $\tau_1 = \tau_2 = 1$, and $d_1 = 3$ and $d_2 = 2$, we have the following input vectors to the first and the second ANCFIS system:

$x_3, x_2, x_1, x_t, y_3, y_2, \underset{\substack{\uparrow \\ Target}}{\boxed{y_t}}$	Input to the first ANCFIS system
$x_3, x_2, x_1, \underset{\substack{\uparrow \\ Target}}{\boxed{x_t}}, y_3, y_2, y_t$	Input to the second ANCFIS system

where x_t and y_t are the one-step ahead targets that must be predicted (plainly the irrelevant target will simply be ignored). Again, performance is measured by the combined accuracy of all systems.

3) MIMO ANCFIS

This ANCFIS design uses a single network with multiple output nodes. It implements the delay embedding technique for multivariate time series and predicts all the variates. For example, for a two-variate time series (x and y) with $\tau_1 = \tau_2 = 1$, and $d_1 = 3$ and $d_2 = 2$, we have the following input vector to the ANCFIS system:

$x_3, x_2, x_1, x_t, y_3, y_2, y_t$

where x_t and y_t are the one-step ahead targets that must be predicted

4) RBFN & SVR

The Radial Basis Function Network (RBFN) is a two layer neural network consisting of a hidden layer with nonlinear transformation and a linear layer output. The network follows a mapping from inputs to output according to [7]:

$$f_r(x) = \lambda_0 + \sum_{i=1}^{n_r} \lambda_i \varphi(\| x - c_i \|) \tag{6}$$

where $x \in R^n$ is the input vector, $\varphi(.)$ is a given function from $\mathbb{R}^+$ to $\mathbb{R}$, $\|.\|$ denotes the Euclidean norm, λ_i are the weights $0 \; i \; n_r$, $c_i \in R^n$ are the RBF centers, $1 \; i \; n_r$, n_r is the number of centers, and $\varphi(.)$ can be a Gaussian function [7]:

$$\varphi(x) = exp\left(\frac{-\| x - c_i \|^2}{\beta^2}\right) \tag{7}$$

where β is the spread parameter.

Support vector regression (SVR) approximates a function based on training data, $\{(x_1, y_1), \ldots, (x_l, y_l)\} \subset \mathbb{R}^d \times \mathbb{R}$, as [36]:

$$f(x) = \langle w, x \rangle + b \tag{8}$$

where $w \in \mathbb{R}^d$ indicates weights, $b \in \mathbb{R}$ is bias, and $\langle .,. \rangle$ is the dot product in $\mathbb{R}^d$. The aim is to find w and b such that the following constraint is satisfied [34, 36]:

$$minimize \;\; \frac{1}{2} \, \|w\|^2 + C \sum_{i=1}^{l} L(y(i), f(x(i), w)) \tag{9}$$

where the first part, $\|w\|^2 = \langle w, w \rangle$, ensures the smoothness of the function. The second part assures that the error between the predicted values by the linear regression, $f(x)$, and actual ones is not more than ε, a user-defined parameter. $C > 0$ is also a user-defined parameter, $y(i)$ is the desired value, and $L(.)$ is the loss function.

To study multivariate time series with RBFN and SVR, the delay embedding technique for multivariate time series is used to obtain the dimension and delay of the variables for the input vectors. For SVR, *N* different systems are designed for *N*-variate time series where each system has same input vector with a different prediction target (as in MISO ANCFIS). In RBFN, a MIMO system is designed with the same input vector as SVR, but with multiple targets. We implement SVR and RBFN with package *e1071* in the R environment [29] and MATLAB (*newrb.m*) [10], respectively.

We split each time series into 2/3 and 1/3 for the training and testing set, respectively; the data points in the training set are chronologically earlier that those in the testing set. The results of the best ANCFIS technique on each data set are compared against RBFN and SVR using a rank-based method on root mean square error (RMSE). First accuracy of each of the architectures is obtained for the multivariate time series with RMSE; for a multivariate time series with *K* variables, the final RMSE for a given approach is obtained as:

$$\text{RMSE} = \sqrt{\frac{1}{K} \sum_{i=1}^{K} MSE_i} \tag{10}$$

where MSE_i is the mean squared one-step-ahead error for the *i*-th variate in the time series. Then, the Friedman test, a well-known nonparametric statistical test, is done to find out if there is a significant difference between the designs. The Friedman statistic, *S*, is calculated as [14]

$$S = \left[\frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2\right] - 3n(k+1) \tag{11}$$

where k is the number of methods applied to *n* time series and R_j is sum of the ranks obtained by applying *j*-th method on the *n* time series. This tests the null hypothesis,

$$H_0 = [\tau_1 = \cdots = \tau_k]$$

against the general alternative:

$$H_1 : [\tau_1, \ldots, \tau_k \; not \; all \; equal]$$

at the α level of significance where τ_i is the *i*-th method effect; H_0 is rejected if $S \geq s_\alpha$ where s_α is a constant obtained by *NSM3* package in R environment [35]. Finally, if a significant difference is found, the performances of the designs will be compared using the Friedman Rank Sums multiple-comparison test [14]:

Decide: $\tau_u \neq \tau_v$ if $|R_u - R_v| \geq r_\alpha$ otherwise $\tau_u = \tau_v$, where τ_i and R_i are the effect and rank sum of the *i*-th method, respectively. r_α is a constant selected such that the experiment-wise error rate equals to α. We obtain r_α from the *NSM3* package.

5) Motel Dataset

This bivariate time series records the monthly occupancy of hotels, motels and guestrooms in Victoria, Australia over the period of Jan 1980 - June 1995 (186 data points) [15]. The first and second variate are the total number of room nights occupied and total revenues (thousands of dollars), respectively. We split the time series into 124 data points in the training set and 62 in the testing set. The delays and dimensions obtained by univariate and multivariate delay embedding techniques are indicated below:

TABLE I. DELAYS AND DIMENSIONS OF THE MOTEL DATASET BY DELAY EMBEDDING TECHNIQUES

MOTEL	Delay (τ)	Dimension (d_1, d_2)
Univariate technique	6	$d_1 = d_2 = 6$
Multivariate technique	6	$d_1 = d_2 = 2$

6) Flour Dataset

This trivariate time series records the monthly flour price on commodity exchanges in Buffalo, Minneapolis, and Kansas City over nine years (1972-1980) [15]. There are 100 observations, which we split into 70 data points in the training set and 30 in the testing set. The delays and dimensions we obtain are shown in Table II:

7) Precipitation Dataset

This time series records the monthly precipitation in the east, middle and west areas of Tennessee from 1895-1989 [48]. The length of the dataset is 1140 giving 760 data points in the training and 380 data points in the testing set. The delays and dimensions we obtain are given in Table III

TABLE II. DELAYS AND DIMENSIONS OF THE FLOUR DATASET BY DELAY EMBEDDING TECHNIQUES

FLOUR	Delay (τ)	Dimension (d_1, d_2, d_3)
Univariate technique	2	$(d_1, d_2, d_3) = (5,4,8)$
Multivariate technique	2	$(d_1, d_2, d_3) = (1,2,1)$

TABLE III. DELAYS AND DIMENSIONS OF THE PRECIPITATON DATASET BY DELAY EMBEDDING TECHNIQUES

PRECIPITATION	Delay (τ)	Dimension (d_1, d_2, d_3)
Univariate technique	3	$(d_1, d_2, d_3) = (5,5,5)$
Multivariate technique	3	$(d_1, d_2, d_3) = (4,5,3)$

IV. EXPERIMENTS

In this section, experimental results of the three alternative architectures of ANCIFS, RBFN and SVR are compared for the three multivariate time series. The RMSE for each algorithm and dataset is given in Table IV.

TABLE IV. EXPERIMENTAL RESULTS - RMSE

	SISO ANCFIS	MISO ANCFIS	MIMO ANCFIS	RBFN	SVR
Motel	0.2181	**0.1952**	0.2080	0.1836	0.1944
Flour Prices	0.1070	0.1102	**0.0963**	0.1030	0.1119
Tenn. Precip.	**0.1495**	0.1841	0.1987	0.1521	0.1522

.

Table V shows the rank of each of the approach based on RMSE in the time series and Table VI indicates the average rank and the rank sum of the methods.

TABLE V. RANK OF THE DESIGNS OVER THE THREE TIME SERIES

	ANCFIS	RBFN	SVR
Motel	3	1	2
Precipitation	1	2	3
Flour	1	2	3

TABLE VI. AVERAGE RANK AND RANK SUM OF THE DESIGNS

	ANCFIS	RBFN	SVR
Average Rank	1.66	1.66	2.66
Rank Sum	5	5	8

The Friedman statistic based on Equation 11 with k=3, n=3 is:

$$S = 2$$

The p-value for the Friedman test is p=0.5278 (%52). Therefore, we conclude that there is no significance difference between the ANCFIS, RBFN and SVR.

V. CONCLUSION

We have studied, for the first time, multivariate time series forecasting by the ANCFIS architecture. In this work, three different versions of ANCFIS in dealing with multivariate time series prediction were examined. The results were compared against two other machine-learning algorithm, RBFN and SVR. In the Flour and Precipitation time series, one of the ANCFIS designs outperforms RBFN and SVR, although in Motel, RBFN shows better performance. The study shows that ANCFIS can be an effective multi-variate forecasting tool. In future work we intend to improve the MIMO ANCFIS because we believe that designing one system to work with all the variables in multivariate time series is more efficient than the other variations of ANCFIS.

REFERENCES

[1] S. Aghakhani and S. Dick, "An on-line learning algorithm for complex fuzzy logic," in *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, 2010, pp. 1-7.

[2] S. Boccaletti, D. Valladares, L. M. Pecora, H. P. Geffert, and T. Carroll, "Reconstructing embedding spaces of coupled dynamical systems from multivariate data," *Physical Review E,* vol. 65, p. 035204, 2002.

[3] L. Cao, A. Mees, and K. Judd, "Dynamics from multivariate time series," *Physica D: Nonlinear Phenomena,* vol. 121, pp. 75-88, 1998.

[4] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural networks,* vol. 5, pp. 961-970, 1992.

[5] C. Chatfield, *Time-series forecasting*: CRC Press, 2000.

[6] S.-M. Chen and K. Tanuwijaya, "Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques," *Expert Systems with Applications,* vol. 38, pp. 10594-10605, 2011.

[7] S. Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *Neural Networks, IEEE Transactions on,* vol. 2, pp. 302-309, 1991.

[8] Z. Chen, S. Aghakhani, J. Man, and S. Dick, "ANCFIS: A neurofuzzy architecture employing complex fuzzy sets," *Fuzzy Systems, IEEE Transactions on,* vol. 19, pp. 305-322, 2011.

[9] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting,* vol. 22, pp. 443-473, 2006.

[10] H. Demuth and M. Beale, "Neural network toolbox for use with MATLAB," 1993.

[11] S. Dick, "Toward complex fuzzy logic," *Fuzzy Systems, IEEE Transactions on,* vol. 13, pp. 405-414, 2005.

[12] M. Han and Y. Wang, "Analysis and modeling of multivariate chaotic time series based on neural network," *Expert Systems with Applications,* vol. 36, pp. 1280-1290, 2009.

[13] R. Hegger, H. Kantz, and T. Schreiber, "Practical implementation of nonlinear time series methods: The TISEAN package," *arXiv preprint chao-dyn/9810005,* 1998.

[14] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric statistical methods*: John Wiley & Sons, 2013.

[15] R. J. Hyndman and M. Akram, "Time series data library," ed, 2006.

[16] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 23, pp. 665-685, 1993.

[17] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge ; New York: Cambridge University Press, 1997.

[18] M. B. Kennel, R. Brown, and H. D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A,* vol. 45, p. 3403, 1992.

[19] L. Kleeman, "Understanding and applying Kalman filtering."

[20] C. Li and F. Chan, "Complex-Fuzzy Adaptive Image Restoration–An Artificial-Bee-Colony-Based Learning Approach," in *Intelligent Information and Database Systems*, ed: Springer, 2011, pp. 90-99.

[21] C. Li and T.-W. Chiang, "Complex Fuzzy Computing to Time Series Prediction A Multi-Swarm PSO Learning Approach," in *Intelligent Information and Database Systems*, ed: Springer, 2011, pp. 242-251.

[22] C. Li and T.-W. Chiang, "Complex neuro-fuzzy self-learning approach to function approximation," in *Intelligent Information and Database Systems*, ed: Springer, 2010, pp. 289-299.

[23] C. Li and T.-W. Chiang, "Function approximation with complex neuro-fuzzy system using complex fuzzy sets–a new approach," *New generation computing,* vol. 29, pp. 261-276, 2011.

[24] C. Li and T.-W. Chiang, "Intelligent financial time series forecasting: A complex neuro-fuzzy approach with multi-swarm intelligence," *International Journal of Applied Mathematics and Computer Science,* vol. 22, pp. 787-800, 2012.

[25] C. Li, T.-W. Chiang, and L.-C. Yeh, "A novel self-organizing complex neuro-fuzzy approach to the problem of time series forecasting," *Neurocomputing,* 2012.

[26] C. Li and T. Chiang, "Complex Neuro-Fuzzy ARIMA Forecasting—A New Approach Using Complex Fuzzy Sets," 2011.

[27] C. Li, T. Wu, and F.-T. Chan, "Self-learning complex neuro-fuzzy system with complex fuzzy sets and its application to adaptive image noise canceling," *Neurocomputing,* vol. 94, pp. 121-139, 2012.

[28] J. Man, Z. Chen, and S. Dick, "Towards inductive learning of complex fuzzy inference systems," in *Fuzzy Information Processing Society, 2007. NAFIPS'07. Annual Meeting of the North American*, 2007, pp. 415-420.

[29] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, "Misc Functions of the Department of Statistics (e1071)," *TU Wien URL: http://cran.rproject.org/web/packages/e1071/e1071.pdf,* 2012.

[30] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *Fuzzy Systems, IEEE Transactions on,* vol. 11, pp. 450-461, 2003.

[31] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *Fuzzy Systems, IEEE Transactions on,* vol. 10, pp. 171-186, 2002.

[32] G. C. Reinsel, *Elements of multivariate time series analysis*: Springer Science & Business Media, 2003.

[33] A. R. Salleh, "Complex intuitionistic fuzzy sets," in *AIP Conference Proceedings*, 2012, p. 464.

[34] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *Computational Intelligence Magazine, IEEE,* vol. 4, pp. 24-38, 2009.

[35] G. Schneider, E. Chicken, and R. Becvarik, "NSM3: Functions and Datasets to Accompany Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods, Third Edition," 1.3 ed, 2015.

[36] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing,* vol. 14, pp. 199-222, 2004.

[37] L.-y. Su, "Prediction of multivariate chaotic time series with local polynomial fitting," *Computers & Mathematics with Applications,* vol. 59, pp. 737-744, 2010.

[38] D. E. Tamir and A. Kandel, "Axiomatic theory of complex fuzzy logic and complex fuzzy classes," *International Journal of Computers, Communications & Control VI (3),* 2011.

[39] D. E. Tamir, M. Last, and A. Kandel, "The Theory and Applications of Generalized Complex Fuzzy Propositional Logic," in *Soft Computing: State of the Art Theory and Novel Applications*, R. R. Yager*, et al.*, Eds., ed: Springer, 2013, pp. 177-192.

[40] D. E. Tamir, H. N. Teodorescu, M. Last, and A. Kandel, "Discrete complex fuzzy logic," in *Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American*, 2012, pp. 1-6.

[41] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega,* vol. 29, pp. 309-317, 2001.

[42] R. R. Yager and A. M. Abbasov, "Pythagorean membership grades, complex numbers, and decision making," *International Journal of Intelligent Systems,* 2013.

[43] O. Yazdanbakhsh and S. Dick, "Time-Series Forecasting via Complex Fuzzy Logic," in *Frontiers of Higher Order Fuzzy Sets*, A. Sadeghian and H. Tahayori, Eds., ed: Springer, 2014, pp. 147-165.

[44] O. Yazdanbaksh, A. Krahn, and S. Dick, "Predicting solar power output using complex fuzzy logic," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, 2013, pp. 1243-1248.

[45] G. Zhang, T. S. Dillon, K.-Y. Cai, J. Ma, and J. Lu, "Operation properties and δ-equalities of complex fuzzy sets," *International journal of approximate reasoning,* vol. 50, pp. 1227-1249, 2009.

[46] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting,* vol. 14, pp. 35-62, 1998.

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