International Journal of Innovative Computing, Information and Control
Volume **14**, Number **5**, October **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 1767–1786

A FUZZY MODEL FOR CHAOTIC TIME SERIES PREDICTION

Matiwos Nigusie Alemu

Department of Information Engineering
Università degli Studi di Brescia
Po Box: 442,25133, Piazza del Mercato, Brescia 15-25121, Italy
godlmati@gmail.com

Received August 2017; revised February 2018

Abstract. *In this paper, a novel fuzzy time series predictor for prediction of chaotic time series data is proposed. Like the basic FTS prediction algorithms, the proposed FTS prediction method contains the four regular steps and an additional step. The proposed fuzzy predictor (PFP) uses fuzzy c-means clustering algorithms to define linguistic variables. Linguistic variables, in literature, are mostly represented by triangular membership functions, modified triangular membership functions or trapezoidal membership functions. Here, a method to find the optimum shape between the three kinds of membership functions is applied. TS fuzzy rules and inference model is used for prediction. Unlike other models, PFP has a T-norm selector. T-norm selector, during training phase, searches the best T-norm operator which results in minimum absolute error. Unlike most fuzzy time series predictors, PFP has an output validator. Output validator checks the validity of the output on occasions when PFP produces an invalid output. Invalid output occurs when the output lies outside the universe of discourse, U. In this case, PFP replaces the result with an output having inputs with similar characteristics to inputs that produce invalid output. To find the best much the minimum of n-dimension Euclidian distance between input training data points is used. The performance of PFP is compared with other models in the literature using the Mackay-Glass time series and Box-Jenkins gas furnace data. Simulation results show the best performance in terms of MSE. This accuracy was achieved by carefully searching optimum shape of MF and the best T-norms in addition to adopting the preferred methods at each 4 steps of basic time series prediction algorithm.*

Keywords: TS fuzzy inference, Fuzzy time series prediction, T-norms, Fuzzy rule-based systems

1. **Introduction.** Conventional time series is a sequential set of data measured over time [1]. Fuzzy time series (FTS), first proposed by Q. Song and B. S. Chissom, extends concepts of conventional time series to that of fuzzy sets [2-4]. FTS is formally defined as [5]:

Definition 1.1. (FTS): *Let $Y(t)$ $(t = \ldots, 0, 1, 2, \ldots)$, a subset of R, be the universe of which fuzzy sets $\mu_i(t)$ $(i = 1, 2, \ldots)$ are defined and let $F(t)$ be a collection of $\mu_i(t)$ $(i = 1, 2, \ldots)$. Then, $F(t)$ is called a fuzzy time series on $Y(t)$ $(t = \ldots, 0, 1, 2, \ldots)$.*

From this definition one can note that the main difference between conventional time series and FTS is that observations in conventional time series are real valued numbers, while in FTS observations are linguistic variables. The purpose of FTS is to predict future values via extraction of patterns in the fuzzy set. Suppose only one FTS past value is used, and then the FTS prediction model is called first order FTS [5].

DOI: 10.24507/ijicic.14.05.1767

Definition 1.2. (First order FTS model): *Suppose that $F(t)$ is caused by $F(t-1)$ only, represented as $F(t-1) \rightarrow F(t)$. This relation is expressed as: $F(t) = F(t-1) \circ R(t,t-1)$, where $R(t,t-1)$ is a fuzzy relationship between $F(t-1)$ and $F(t)$, and $F(t) = F(t-1) \circ R(t,t-1)$ is called the first order model of $F(t)$.*

The fuzzy relationship defined by $R(t,t-1)$ can be dependent or independent of time [6]. If $R(t,t-1)$ is independent of t, then $F(t)$ is called time invariant FTS. Otherwise, it is called time variant FTS [7].

Definition 1.3. (Time Variant and Time Invariant Fuzzy Time Series): *Suppose that $R(t,t-1)$ is a first-order model of $F(t)$. If for any t, $R(t,t-1)$ is independent of t, i.e., $R(t,t-1) = R(t-1,t-2)$, then $F(t)$ is called a time invariant fuzzy time series; otherwise, it is called a time variant fuzzy time series.*

If n past values are used for predicting the next value, then the FTS model is called n^{th} order FTS [7].

Definition 1.4. (n^{th} order FTS): *If $F(t)$ is caused by $F(t-1), \ldots, F(t-n)$ for $n \geq 0$, then the fuzzy relationship is expressed as $F(t-1), F(t-2), \ldots, F(t-n) \rightarrow F(t)$. It is called n^{th} order FTS.*

If the time series in Definition 1.1 has finite mean and variance that does not change overtime, then the time series is stationary.

Definition 1.5. (Stationary Time Series): *A time series $Y(t)$ $(t = \ldots, 0, 1, 2, \ldots)$ is said to be stationary if,*

a) $E[Y(t)] = \alpha$, $(t = \ldots, 0, 1, 2, \ldots)$

b) $E[Y(t)^2] < \infty$, $(t = \ldots, 0, 1, 2, \ldots)$

c) $\gamma[(Y(s,t)] = \gamma$, $Y(s+h, t+h)$ $\forall s, t, h \in \{\ldots, 0, 1, 2, \ldots\}$

In the above standard stationary time series definition, $Y(t)$ must have the features: finite variation, constant first moment and the second moment depending only on the difference $(t-s)$. For non-stationary time series, the probability distribution characteristics given in a), b) and c) change with time. Because of this change of parameters, predicting non-stationary time series is a difficult task.

In this paper a fuzzy inference model to predict chaotic or non-stationary time series is discussed. The rest of the paper is organized as follows. In Section 2 available prediction methods in literature are discussed. In Section 3 detailed explanations of fuzzy time series prediction models and algorithms along with their comparisons are presented. In Section 4 the proposed fuzzy time series prediction model is explained followed by presentation of simulation results. In Section 5 simulation results are analyzed, and finally, conclusion are given in Section 6.

2. **Predictive Models in Literature.**

In literature predictive models can be either traditional predictive models or intelligent soft computing methods [8]. Auto regressive (AR), moving average (MA), Box-Jenkins's method (ARIMA) models are traditional predictive models. Traditional models are not suitable for predicting non-stationary time series data which are characterized by highly nonlinear and chaotic nature [9]. For such cases intelligent soft computing techniques are used. ANN models are one of soft computing techniques that are universally employed in time series predictions. Properties such as self-organizing, data drivability, self-study, self-adaptability and associability make them acceptable predictor [10]. The cons of ANN rise from the black box nature of ANN, greater computational burden, proneness to over fitting, and the empirical nature of model development [11]. To reduce such effects a hybrid model consisting of combination of ANN and evolutionary or other varieties of optimization algorithms are being applied. For instance, [4,13,14] investigate use of genetic algorithm for reducing local minima problems of feed forward and recurrent neural network models. In [15] the same approach is adopted for wavelet neural networks (WNN). Besides GA other optimization algorithms are available. In [16] particle swarm optimization is used to train quantile regression neural network models. Hybrid models with more than 2 independent models are also investigated. In [17] two hybrid prediction models, each with 3-stage sub models, are proposed. The first predictor uses chaos theory, followed by multi-layer perceptron (MLP) and multi-objective particle swarm optimization (MOPSO) while the second model replaces the last stage with elitist non-dominated sorting genetic algorithm (NSGA-II). Hybrid models are proved to be effective, but they are complex computationally and did not completely eliminate ANN drawbacks, rather reduce it. To solve big data problems efficiently quantum inspired algorithms are being developed recently. One of such is quantum neural network (QNN) which is information processing architecture inspired by quantum mechanics. [18] proposes a QNN architecture incorporating Schrodinger wave equation referred to as recurrent quantum neural network (RQNN) and applies it for filtering electroencephalogram (EEG) signals. In the work of [19] the concept is further extended for the self-organized neural network resulting in self-organizing quantum neural network. The application of QNN is studied in [20] for chaotic time series prediction. However, QNN models are in infant stage and need to be further investigated.

3. **Fuzzy Time Series Predictor.**

This section presents the basic FTS prediction algorithm which contains four steps initially suggested by Q. Song and B. S. Chissom [2-4].

3.1. **Step 1: Universe of discourse definition and interval partition.**

Step 1 consists of two processes which are universe of discourse, U, definition and interval partitioning.

3.1.1. *Universe of discourse definition.* Universe of discourse is defined as:

$$U = [(D_{\min} - c_1), (D_{\max} + c_2)] \tag{1}$$

where $D_{\min}$ and $D_{\max}$ are minimum and maximum values of training data, and c_1 and c_2 are proper positive constants. c_1 and c_2 should be selected carefully, which is because testing data are expected to lie on U. Following are some typical choices of values for the constants c_1 and c_2.

a. Arbitrary Positive Number: [2-4,21-26] choose c_1 and c_2 to be arbitrary positive numbers without explaining any persuasive reason to the question why to choose arbitrary number [27].

b. Standard Deviation Based Selection: To ensure future testing results fall in the universe of discourse, U, statistical distribution of training data such as standard deviation can be used as done in [27,28]. Standard deviation σ measures spread of training data with respect to the mean value. Thus, universe of discourse is defined as [28]:

$$U = [(D_{\min} - \sigma), (D_{\max} + \sigma)] \tag{2}$$

where $D_{\min}$, $D_{\max}$ and σ are the minimum, maximum and standard deviation of training data respectively.

3.1.2. *Interval partition.* Let N be the total number of training data: interval partition involves the partition of this N training data into k intervals [29]. The length of each interval plays an important role for prediction accuracy [24]. When length of intervals is too large, there will be no fluctuation in FTS. On the other hand, when the length is too small, the meaning of FTS will diminish. In general, training data can be partitioned in to equal or unequal intervals. Following are common methods of interval partitioning.

1) **Equal Interval Partitioning.** The simplest way to partition U into k interval is to divide it equally [2-4,21-26]. For $U = [(D_{\min} - c_1), (D_{\max} + c_2)] = [L, R]$, n intervals are defined as [7]:

$$u_i = [L + (i - 1)l, L + il], \quad i = 1, 2, \ldots, n \tag{3}$$

where l is interval length and $n = (R - L)/l$ [27].

2) **Unequal Interval Partitioning**

i. **Distribution and Average Based:** [23] proposes interval partitioning method based on distribution and average based length of training data. These two methods consider fluctuations of FTS. Distribution based length is calculated according to distribution of first difference of training data. Average based length is set to the largest length that is smaller than half the first differences.

ii. **Minimum Entropy Based Approach:** The entropy of a probability distribution is a measure of uncertainty of a distribution. A key goal of entropy minimization analysis is to determine the quantity of information in a given data set. To subdivide the data into k intervals, a threshold line between classes of data is determined by entropy minimization screening method. Then, segmentation process is started, first into two classes then into k number of fuzzy sets [27].

iii. **Modified Cumulative Probability Distribution Approach (MCPDA) [28]:** Partition U into several intervals based on cumulative probability distribution of each linguistic variable.

iv. **Heuristic Search Algorithms:** Universe of discourse U on Equation (1) can be divided into k intervals as:

$$U_1 = [(D_{\min} - c_1), x_1],\ U_2 = [x_1, x_2],\ \ldots,\ U_k = [x_{k-1}, (D_{\max} + c_2)] \tag{4}$$

Search algorithms aim at finding the best values of x_i $(i = 1, 2, \ldots, k - 1)$ which minimizes prediction error. One can apply genetic algorithm as in [24,26], simulated annealing (SA) as in [30], imperialist competitive algorithm (ICA) as in [31]. SA finds interval coordinates x_i $(i = 1, \ldots, k-1)$ from initial random coordinates. ICA is a global search heuristic which mimics imperialism and imperialistic competition process as a source of inspiration [31].

v. **Clustering Algorithms:** Here, training data is partitioned into k interval using clustering algorithms. The most popular clustering algorithms are K means and FCM.

a. **K Means Clustering Algorithm:** K means clustering algorithm partitions N data points into k disjoint subsets containing N_j data points that minimize the sum of squares criteria given by:

$$J = \sum_{j=1}^{k} \sum_{n \in N_j} |x_n - \mu_j|^2 \tag{5}$$

where x_n is a vector representing the n^{th} data point and μ_j is geometric centroid of the data points in N_j. In [32] K means clustering algorithm is used to partition U.

b. FCM: Fuzzy c-means clustering algorithm (FCM), developed by Dunn in 1973 and improved by Bezdek in 1981, is unsupervised clustering algorithm. FCM is the most successful clustering algorithm [33]. Hence, FCM can be used to determine centers of intervals as in [29,33]. In FCM algorithm, each data point belongs to a cluster with degree specified by a membership grade. Given k number of clusters and $X = \{x_1, x_2, \ldots, x_n\}$ a collection of data points, FCM partitions given data into clusters by minimizing the within group sums of squared error function [34]:

$$J_m(M,V) = \sum_{j=1}^{n}\sum_{i=1}^{k}(\mu_{ij})^m \|x_j - v_i\|^2, \quad 1 \leq m \leq \infty \tag{6}$$

where $J_m(M,V)$ is the sum of squared error for the set of fuzzy clusters represented by the membership matrix M, and the associated set of cluster centers V. $\|x_k - v_i\|^2$ is the distance between data x_k and i^{th} cluster center v_i. m governs the influence of membership grades. The partition becomes fuzzier with increasing m and it is proven that FCM algorithm converges for any $m \in (1,\infty)$ [5]. The necessary conditions for Equation (6) to reach minimum are:

$$\mu_{ik} = \left(\sum_{j=1}^{c}\left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|}\right)^{\frac{2}{(m-1)}}\right)^{-1} \quad \forall i,\ \forall k \tag{7}$$

$$v_i = \frac{\sum_{k=1}^{n}(\mu_{ik})^m x_k}{\sum_{k=1}^{n}(\mu_{ik})^m} \tag{8}$$

where μ_{ik} is the membership degree of the k^{th} data with respect to i^{th} cluster and v_i is the i^{th} cluster center. The steps of FCM algorithm are [35]:

```
Begin
Input: Initial cluster center vector $V$ with cardinality $C$
Output: A set of cluster centers $V$ and MF matrix $\mu_{ik}$
While ($\Delta E \leq \varepsilon$ OR $E < \delta$)
    For No_Training_data = 1 to $N$
        Calculate membership matrix $\mu_{ik}$ by using Equation (7)
    End For
    For No_Cluster_Center = 1 to $C$
        Update Cluster center $V$ by using Equation (8)
    End For
    For No_Training_data = 1 to $N$
        For No_Cluster_Center = 1 to $C$
            Calculate the within group sums of squared error
            using Equation (6)
        End For
    End For
    Compute $\Delta E = |E_{Previous} - E_{Current}|$
End While
end
```

where N is number of training data, E is the within group sums of squared error, δ is a threshold minimum error, $\Delta E = |E_{Previous} - E_{Current}|$ and ε is a threshold number that is used to compare E improvement over the previous iteration.

3.1.3. *Interval partition methods: A discussion.* Though equal interval partitioning is the simplest method, it has several drawbacks. The first is lack of strong persuasive reason that explains the question why to divide U into equal interval [27]. Besides, it does not consider distribution of real data. Thus, it fails to generate good result in cases where the distribution of continuous values is not uniform [33]. Distribution and average based [23], minimum entropy-based approach [27] and MCPDA [28] have a reason on how to partition U. However, in terms of accuracy, these methods are outperformed by search algorithms and FCM [24,26,29-33], GA [24,26], SA [30] and ICA [31] are heuristic. Heuristic algorithms are designed to give an acceptable answer for a typical problem. As such they do not guarantee optimum answer all the time [40]. Besides this, search algorithms are slow and take a lot of execution time. Clustering algorithms are reasonable and accurate method to partition universe of discourse since they take distribution of real data points into account [29,33]. In K means clustering algorithm each data only belongs to one cluster, while in FCM data can belong to more than one cluster. FCM takes uncertainty into account by giving degree of membership of each data to each cluster. Therefore, FCM is suitable for capturing real world clustering problems [37]. Therefore, FCM is the preferred method to partition universe of discourse.

3.2. **Step 2: Defining fuzzy sets and fuzzifying time series.** Recall in Definition 1.1 that, an FS A in U is characterized by an MF, $\mu_A(x)$, which associates with each point in U a real number in the interval $[0, 1]$. Thus, in Step 2 MFs corresponding to k linguistic variables are defined. Q. Song and B. S. Chissom first define a fixed fuzzy set for each interval. [2-5,24,26] use this kind fuzzy set definition. For example, if there are five intervals, then for the first interval, fuzzy set A_1 may be defined as:

$$\mu_1 = 0.8/A_1 + 0.2/A_2 + 0/A_3 + 0/A_4 + 0/A_5$$

This type of fuzzy set definition assigns a fixed grade of membership for all crisp values. The other kind of fuzzy set definition is to define triangular membership function on each interval [27,30]. For example, given 4 intervals with cluster center:

$$u_1 = [(D_{\min} - c_1), x_1],\ u_2 = [x_1, x_2],\ u_3 = [x_3, x_4],\ u_4 = [x_4, (D_{\max} + c_2)]$$

then MF is defined by Figure 1.

The problem of this type of traditional triangular MF definition lies on the first and last fuzzy sets. For crisp values less than x_1 and greater than x_4, grade of membership decreases. Because there is no FS beyond the first and last, those values should have a value of 1. [28] proposes a modified triangular membership function shown in Figure 2. The other kind of MF is trapezoidal membership function proposed by [27]. This kind

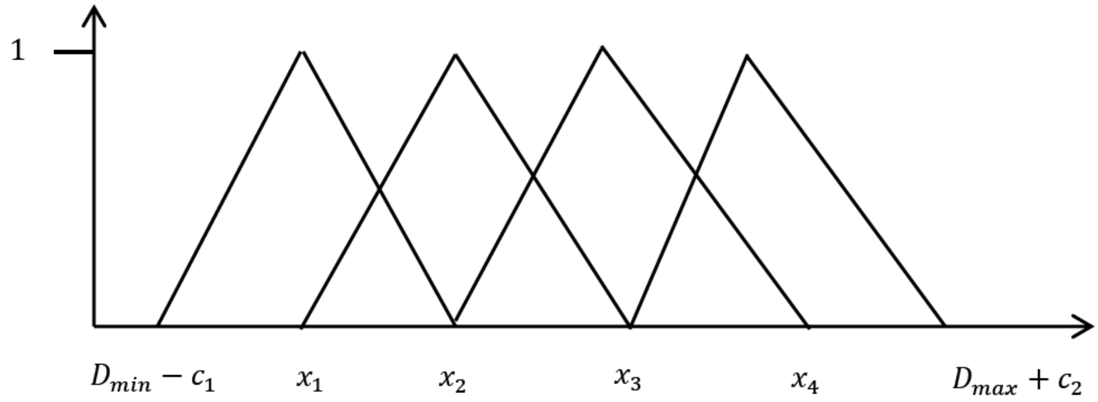

FIGURE 1. Triangular membership function

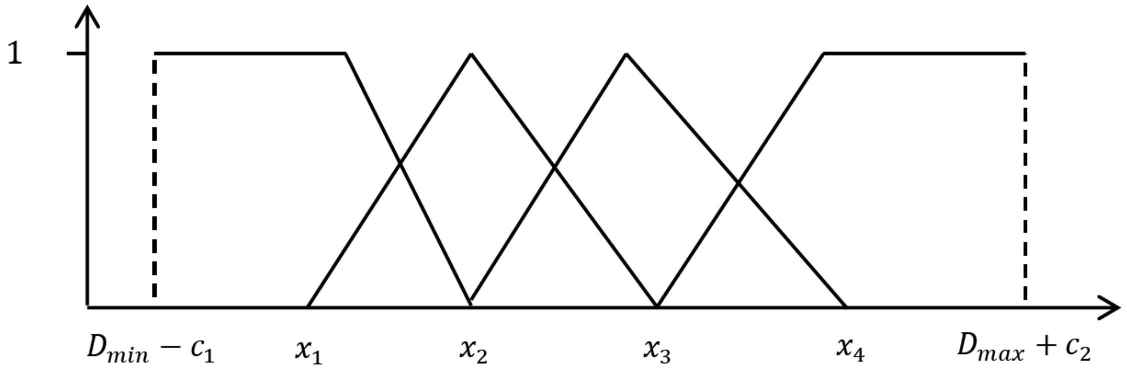

Figure 2. Modified triangular membership function

of definition is like triangular MF except the shape is trapezoid, therefore suffers similar problem as triangular MF. Modified triangular MF and trapezoidal MF are the most widely used fuzzy set definitions. In this paper, a method to find the optimum shape of MF between triangular and modified triangular MF is applied.

3.3. **Steps 3 and 4: Prediction model.**

The most popular prediction models for FTS are the Q. Song and B. S. Chissom type of prediction model and prediction model that uses TS fuzzy inference models.

3.3.1. *Q. Song and B. S. Chissom prediction.* When Q. Song and B. S. Chissom first propose FTS, they also introduce a method of forecasting students' enrollment. The prediction model is given as [2-4]:

$$A_i = A_{i-1} \circ R \tag{9}$$

where A_{i-1} is the enrollment of year $i-1$ in terms of FS and A_i is the forecasted enrollment of year i in terms of FS, and R is a fuzzy relation which indicates fuzzy relationships between time series. The operator $\circ$ is a max-min composition operator.

The derivation of fuzzy relation is a very tedious work and the max-min composition operator will take a large amount of time when the fuzzy relation R is very big [21]. Hence, Chen in [21] replaces max-min composition with simplified arithmetic operations, as a result, proposes more efficient method. [22-27,33,37,38] use Chen's algorithms with some minor modification. For instance, [26] builds weighted fuzzy rules by computing cardinality of each fuzzy relation. [38] proposes weighted fuzzy rules by calculating the count of each fuzzy relation.

3.3.2. *TS-fuzzy inference model.* Inference is the process of obtaining new knowledge from existing one. In classical logic there are two inference methods: the modus phones and modus tollens [39]. In modus phones if there is a rule or implication "*if x is A then y is B*" and for the fact "*x is A*", then one can infer that "*y is B*". In modus tollens for the same rule and the fact that "*y is not B*", one can infer "*x is not A*". Modus tollens require knowledge of consequent which is difficult in real world application [40]. In FL the modus phones inference is defined in such a way that it allows an inference when the fact is only partially known, i.e., to a certain degree of membership. For instance, for fuzzy rule: if x is A, then y is B and for the fact that $\mu_A(x) = a$, $0 \leq a \leq 1$, then fuzzy inference answers the question $\mu_B(y) = ?$.

In general, for a system having n fuzzy rules, the following scheme represents modus phones [41]:

Rule 1: if x_1 *is* A_{11} *and* x_2 *is* A_{21} *and* $\ldots$, *then* y_1 *is* B_{11} *and* y_2 *is* B_{21} *and* $\ldots$
Rule 2: if x_1 *is* A_{12} *and* x_2 *is* A_{22} *and* $\ldots$, *then* y_1 *is* B_{12} *and* y_2 *is* B_{22} *and* $\ldots$
$\vdots$
Rule n: if x_1 *is* A_{1n} *and* x_2 *is* A_{2n} *and* $\ldots$, *then* y_1 *is* B_{1n} *and* y_2 *is* B_{2n} *and* $\ldots$
For the fact: if x_1 *is* A_{11} *and* x_1 *is* A_{21}

One can conclude that: y_1 is B_{11} and y_1 is B_{21} and $\ldots$.
where x_1 is input variable of the system, y_1 is output variable of the system and A_{ij} is the i^{th} fuzzy set of rule j. Consider rules of the form:

$$\textit{if } f(x_1 \textit{ is } A_1 \textit{ and}, \ldots, x_k \textit{ is } A_k) \textit{ then } y = g(x_1, \ldots, x_k) \tag{10}$$

then, the fuzzy inference is called TS fuzzy model, first proposed by T. Takagi and M. Sugeno [42]. In Equation (10), y is variable of the consequence whose value is inferred, i.e., output variable of the system. $x_1, \ldots, x_k$ are inputs to the system, $A_1, \ldots, A_k$ are fuzzy sets, f is logical function that connects proposition of the premise and g is a crisp mathematical function. When g is constant, the resulting fuzzy inference system is called zero order TS fuzzy model. When g is first order polynomial, the resulting fuzzy inference system is called first order TS fuzzy model. A first order TS model has rule of the form [40]:

$$\textit{if } x_1 \textit{ is } A_1 \textit{ and } \ldots x_n \textit{ is } A_l \textit{ then } y = p_0 + p_1 * x_1 + \cdots + p_n * x_n \tag{11}$$

In TS fuzzy model the consequence is a crisp output. The truth value of consequent proposition $y = g(x_1, \ldots, x_k)$ is calculated from aggregate of truth values in the antecedent proposition. Given a system modeled by TS fuzzy system, how to adjust parameters of consequent is discussed next.

3.3.3. *Least square learning.* Take a system modeled by first order TS fuzzy rules. Let Y_d be $N \times 1$ vector of output data, X be $N \times M$ input data and Y be output of the fuzzy system. Each X_i is $1 \times M$ data the system inputs to produce output Y_i:

$$Y_d = [y_1, y_2, \ldots, y_N]^T \tag{12}$$

Out of R number of TS fuzzy rules, the i^{th} TS rule and crisp output corresponding to i^{th} rule is given by:

$$\begin{aligned} \textit{Rule}_i &= \textit{if } x_1 \textit{ is } \mu_1 \textit{ and } \ldots \textit{ and } x_M \textit{ is } \mu_M \\ \textit{then } y_i &= p_{i0} + p_{i1} * x_1 + \cdots + p_{iM} * x_M \end{aligned} \tag{13}$$

Equivalently, y_i can be written as,

$$y_i = \mu_{i1}(p_{10} + p_{11} * x_1 + \cdots + p_{1M} * x_M) + \cdots + \mu_k(p_{k0} + p_{k1} * x_1 + \cdots + p_{kM} * x_M) \tag{14}$$

Each rule is evaluated to have a truth value μ_i and satisfy the condition: $\sum_{i=1}^{k} \mu_i = 1$. For N number of input output training data, the task of LS learning is estimating consequent parameters: $p_{10}, p_{11}, \ldots, p_{1M}, \ldots, p_{k0}, p_{k1}, \ldots, p_{kM}$. Collecting all i^{th} output in Equation (14) and rearranging terms, the output of the fuzzy system Y is $N \times 1$ matrix given by [39]:

$$Y = A * K \tag{15}$$

where A is $N \times k * (1 + M)$ matrix given by:

$$A = \begin{bmatrix} \mu_{11} & \mu_{11}x_1 & \mu_{11}x_2 & \cdots & \mu_{11}x_M & \cdots & \mu_{1k} & \mu_{1k}x_1 & \mu_{1k}x_2 & \cdots & \mu_{1k}x_M \\ \mu_{21} & \mu_{21}x_1 & \mu_{21}x_2 & \cdots & \mu_{21}x_M & \cdots & \mu_{2k} & \mu_{2k}x_1 & \mu_{2k}x_2 & \cdots & \mu_{2k}x_M \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mu_{N1} & \mu_{N1}x_1 & \mu_{N1}x_2 & \cdots & \mu_{N1}x_M & \cdots & \mu_{Nk} & \mu_{Nk}x_1 & \mu_{Nk}x_2 & \cdots & \mu_{Nk}x_M \end{bmatrix} \tag{16}$$

Here, μ_{ij} is truth value of j^{th} rule corresponding to i^{th} output. And K is $k * (1 + M) \times 1$ vector of unknown coefficients of consequent parameters given by:

$$K = [p_{10} \quad p_{11} \quad p_{12} \quad \ldots \quad p_{1M} \quad \ldots \quad p_{k1} \quad p_{k2} \quad p_{k3} \quad \ldots \quad p_{kM}]^T$$

Usually the number of input-output patterns used in training is greater than the number of consequent parameters, i.e., $N \geq k * (1 + M)$. Hence, the problem of finding unknown coefficients K is an over determined problem in which exact solution may not even exist. However, list square estimate of K, K^*, that minimize the squared error $\|Y - Y_d\|^2$ can be found using pseudo inverse technique [43]:

$$K^* = \left(A^T A\right)^{-1} A^T Y_d \tag{17}$$

where A^T is the transpose of A, and $\left(A^T A\right)^{-1} A^T$ is the pseudo inverse of A if $\left(A^T A\right)$ is non-singular.

3.3.4. *Prediction models: A comparison.* The main drawback of Q. Song and B. S. Chissom prediction model and its varieties is that it fails or produces large error for larger number of input-output data. Therefore, it is inappropriate for chaotic time series data [32]. It also lacks the ability of tuning parameters for learning. TS fuzzy inference model is appropriate for prediction of large number of input-output data. TS fuzzy model combines the advantages of classical linear regression models from the coefficients of consequent parameters and accepts uncertainty and non-linearity via linguistic variables in the antecedent of TS fuzzy rules [32]. Consequent parameters are also tunable, e.g., LS learning can be used to tune consequent parameters [39,43]. Therefore, TS-fuzzy model is the preferred FTS prediction model for chaotic time series prediction [39,40,42,43].

4. **Proposed Fuzzy Time Series Predictor.**

Like basic FTS predictor, the proposed FTS prediction method (PFP) contains four regular steps and an extra additional step. In this section detailed explanation of PFP is presented.

4.1. **Step 1: Universe of discourse, U, definition and interval partitioning.**

The first step is to define universe of discourse U. U is defined based on Equation (2) as:

$$U = [(D_{\min} - \sigma), (D_{\max} + \sigma)]$$

where $D_{\min}$, $D_{\max}$ and σ are minimum, maximum and standard deviation of training data respectively. Then training data is partitioned into k interval to define LVs. As discussed in Section 3.1.3. FCM is the preferred method to partition training data. Hence, FCM is used for interval partitioning. See Section 3.1.2 for FCM algorithm.

4.2. **Step 2: Defining fuzzy sets and fuzzifying time series.**

The outputs of FCM invoked in Step 1 are k cluster centers. In Step 2, depending on the number of cluster centers, k LVs are defined. Linguistic variables, in literature, are represented by triangular MF, modified triangular MF or trapezoidal MF. Here, the optimum shape of MFs is selected by linearly searching the width of trapezoidal MFs. The method used by PFP to find the optimum shape of MF between triangular MF, modified triangular MF and trapezoidal MF is discussed next.

Optimum shape of membership function. Let the number of LVs be 4, i.e., $k = 4$. The PFP searches the best shape by sequentially or linearly searching discrete value of width at upper base of a trapezoidal MF. The best shape is the one which yields the smallest mean square error during training. Figure 3 shows modified triangular MF for two different width values. When the first width $width_1$ is changed to $width_2$, so are points a, b, c and d changed to a', b', c' and d'. These points define the slops that make up legs of the trapezoid. When both $width_1$ and $width_2$ both equal 0, the trapezoidal MF is reduced to modified triangular MF. Width can take positive value starting from zero to the maximum allowed value. The limit in the maximum value is due to the condition for all values of x, the sum of degree of MFs a particular value, i.e., $x \in U$ can take should not exceed 1. In other words, any values of width must satisfy the condition:

$$\sum_{i=1}^{4} \mu_i(x) = 1, \quad \forall x \in U \tag{18}$$

Consider the following observations to determine the maximum value of width. From Figure 3, point a can take any value in between cluster center two (c_2) and cluster center one (c_1). When a is at c_1, $width_1$ attains maximum allowed value and is equal to the difference of c_2 and c_1. Similarly, d can take values from center of cluster 3 (c_3) to the center of cluster four (c_4). When d is at c_4, $width_2$ is maximum allowed value and is equal to the difference between c_3 and c_4. However, for points b and c the maximum width is determined by the point where the two meet. a and b meet at half the difference of cluster center 3 (c_3) to that of cluster center 2 (c_2), i.e., $\frac{c_3-c_2}{2}$.

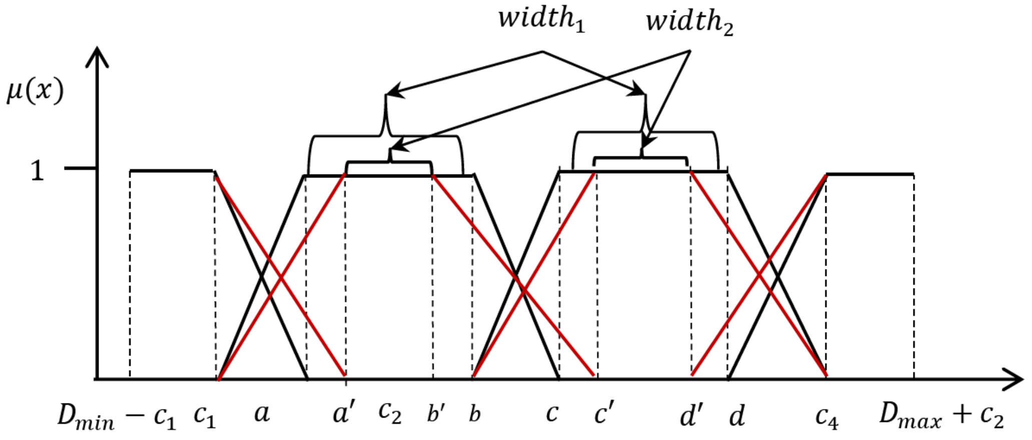

FIGURE 3. Search process for optimum shape of MF

Varying $width_1$ and $width_2$ starting from centers (i.e., from $width_1 = width_2 = 0$), the maximum width is determined by whether point a reaches c_1 first or d reaches c_4 first or b and c meet first. The first to happen from the three determines the maximum value of width. For four linguistic variables the maximum width is formally defined as:

Definition 4.1. (Allowed maximum width for 4 fuzzy set): *For four fuzzy set width of trapezoidal membership function can take values which lie in the interval* [0-*maximum width*]*, where maximum width is given by:*

$$Max\ Width = \begin{cases} 2 * \min(\Delta_1, \Delta_3) & if\ \Delta_1\ or\ \Delta_3 < \dfrac{\Delta_2}{2} \\ \Delta_2 & otherwise \end{cases} \tag{19}$$

Δ_1 = *cluster center* 2 − *cluster center* 1, Δ_2 = *cluster center* 3 − *cluster center* 2 *and* Δ_3 = *cluster center* 4 − *cluster center* 3.

Equation (19) is valid only for 4 linguistic variables. For N numbers of fuzzy sets similar logic can be applied. Allowed maximum width is defined as:

Definition 4.2. (Allowed maximum width for *N* fuzzy set): *For N fuzzy set width of upper side of trapezoidal membership function take values which lie in the interval* [0-*maximum width*]*, where maximum width is given by:*

$$Max\ Width = \begin{cases} 2 * \min(\Delta_1, \Delta_n) & if\ \Delta_1\ or\ \Delta_n < \dfrac{\Delta_i}{2}\ \ \forall i \in [2, n-1] \\ \min(\Delta_2, \Delta_3, \ldots, \Delta_{n-1}) & otherwise \end{cases} \tag{20}$$

where Δ_1 = *cluster center* 2 − *cluster center* 1*,* Δ_i = *cluster center* (i) − *cluster center* $(i-1)$*,* $i = [2, 3, \ldots, n-1]$.

The PFP searches the optimum MF between triangular MF, modified triangular MF and trapezoidal MF. As discussed in Section 3.2 another approach to find the best coordinates, a, b, c and d, is to use search algorithms. Search algorithms find the best value of these coordinates. Here in PFP, the best coordinates are searched by controlling only one valued variable: width of trapezoidal MF. Therefore, searching optimum shape of MF takes no more than a loop. Compared to search algorithms this makes the proposed method fast.

4.3. **Steps 3 and 4: prediction model.** The PFP uses first order TS model for prediction. First order TS rules are of the form (Equation (11)):

$$if\ x_1\ is\ \mu_1\ and\ \ldots x_n\ is\ \mu_l\ then\ y = p_0 + p_1 * x_1 + \cdots + p_n * x_n$$

where x_i, $i = 1, 2, \ldots, n$ are input variables, μ_l is degree of membership of x to the FS A_l and p_j, $j = 0, 1, \ldots, n$ are consequent parameters. The consequent parameters are obtained using LS learning algorithm (Section 3.3). TS fuzzy rules are extracted from training data. The maximum number of rules is related to the order of FTS and number of MFs defined in Step 2. The maximum number of rules is given as:

$$\text{Max no of rules} = (\text{No of fuzzy sets})^n \tag{21}$$

where n is the order of FTS. However, all rules may not present in the training data, in such cases the number of rules is less than the maximum number of rules given in Equation (20). Recall in Section 3.3.2 that T-norms are used in antecedent evaluation. Most FTS predictors [21-32] use the default minimum T-norm operator. PFP has a T-norm selector which selects the best T-norm operator. The best operator is the one with minimum absolute error during training phase. Table 1 shows T-norm operators used by T-norm selector of the PFP. In the testing phase the best T-norm is used.

4.4. **Output validator.** The output validator is an extra step to the common fuzzy time series predictor algorithm. This step is introduced to avoid sudden spike errors in the testing phase. On some occasions PFP may produce an invalid output. Invalid output occurs when PFP predicts an output which lies outside universe of discourse, U. In such cases, the output should be corrected to a reasonable value. When the output is invalid, PFP replaces the result with training data which have similar characteristics to inputs that produce invalid output. To find the best match, Euclidean distance is used. For example, if invalid output is predicted for the inputs, x_1, x_2, x_3 and x_4, then from n training data the output validator searches minimum distances given by

$$\left(x_1 - x_i^t\right)^2 + \left(x_2 - x_{i+1}^t\right)^2 + \left(x_3 - x_{i+2}^t\right)^2 + \left(x_4 - x_{i+3}^t\right)^2, \quad i = 1, 2, \ldots, (N-3) \tag{22}$$

where x_i^t is the i^{th} training data. The invalid output is replaced by the output of the

Table 1. T-norms used by T-norm selector of PFP

Minimum	$T_M(\mu_A(x),\mu_B(x)) = \min(\mu_A(x),\mu_B(x))$
Drastic Product	$T_D(\mu_A(x),\mu_B(x)) = \begin{cases} \min(\mu_A(x),\mu_B(x)) & \text{if } \max(\mu_A(x),\mu_B(x)) = 1 \\ 0 & \text{otherwise} \end{cases}$
Algebraic Product	$T_{AP}(\mu_A(x),\mu_B(x)) = \mu_A(x) * \mu_B(x)$
Bounded Difference or Lukasiewicz and Operator	$T_L(\mu_A(x),\mu_B(x)) = \max(0, \mu_A(x) + \mu_B(x) - 1)$
Einstein Product	$T_E(\mu_A(x),\mu_B(x)) = \dfrac{\mu_A(x) * \mu_B(x)}{2 - (\mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x))}$
Hamacher Product	$T_H(\mu_A(x),\mu_B(x)) = \dfrac{\mu_A(x) * \mu_B(x)}{\mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x)}$
Yager Intersection	$T_Y(\mu_A(x),\mu_B(x)) = 1 - \min\left\{1, \left((1-\mu_A(x))^b + (1-\mu_B(x))^b\right)^{\frac{1}{p}}\right\}, \ P \geq 1$
Dubois and Prade	$T_{DP}(\mu_A(x),\mu_B(x)) = \dfrac{\mu_A(x) * \mu_B(x)}{\max\{\mu_A(x), \mu_B(x), \alpha\}}, \ \alpha \in [0,1]$

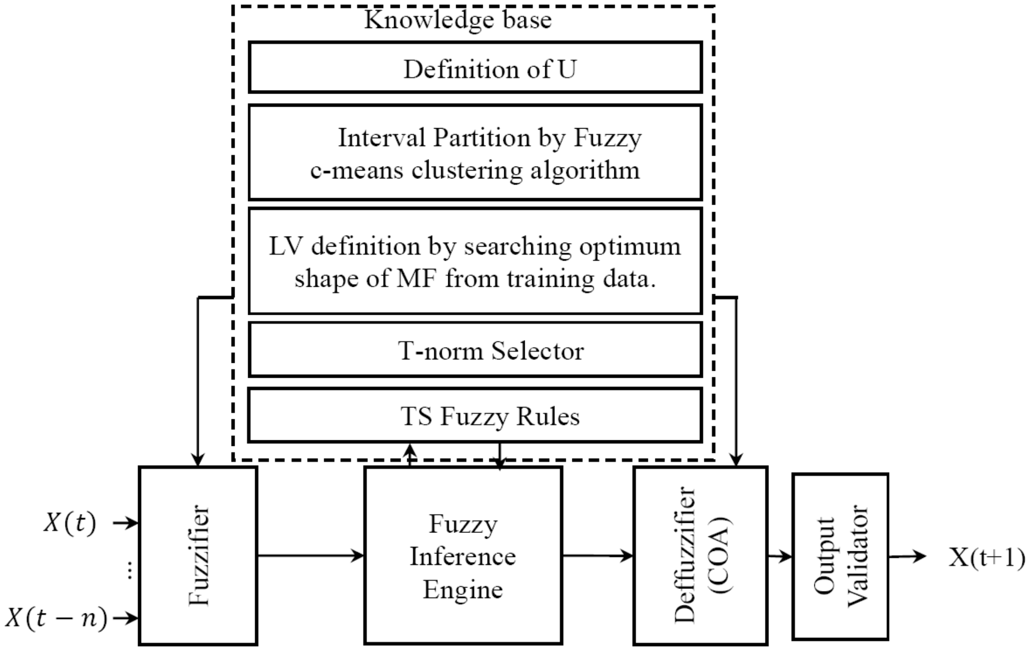

Figure 4. Proposed fuzzy predictor

predictor with inputs: x_i^t, x_{i+1}^t, x_{i+2}^t, x_{i+3}^t, where i is the index of training data that result in minimum distance among n number of input data.

4.5. **PFP summary.** Figure 4 summarizes PFP for n order FTS. The knowledge base contains the prediction algorithm database which includes universe of discourse U defined by Equation (2) in Step 1, linguistic variables defined in Step 2 and the fuzzy rules defined in Step 3. Linguistic variables are defined after linear search of optimum membership function. The optimum shape occurs in between the triangular MF, modified triangular MF or trapezoidal MF. The fuzzifier, fuzzy inference engine and deffuzzifier perform the whole prediction mechanism in Step 3 and Step 4.

TS fuzzy model is used as inference engine. In the inference engine, the truth value of consequent proposition is calculated from aggregate of truth values in the antecedent proposition. The aggregate of antecedent proposition is calculated using the best T-norm. The T-norm selector, during the training phase, selects the best T-norm operator with minimum absolute error. PFP evaluates the best one from 8 T-norm operators. For TS type rules, the crisp output is equivalent to defuzzification performed by using centroid of area. Finally, output validator checks the validity of the output.

5. **Simulation Results.**

5.1. **Performance evaluation matrix.**

PFP algorithm is evaluated using well-known performance evaluation metrics: mean square error and root mean square error which are defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y(i) - Y_p(i)\right)^2 \tag{23}$$

where $Y(i)$ is actual output, $Y_p(i)$ is the predicted output and N is the no of data points. Root mean square is defined as the square root of mean square error.

$$RMSE = \sqrt{MSE} \tag{24}$$

However, MSE and RMSE are summaries of error distribution for a specific model. Moreover, MSE and RMSE do not address the problem of quantifying a given model regarding its sensitivity to initial parameterization [44]. For models like neural network and fuzzy systems, different results are obtained at each run of algorithm. This is because of random initialization of certain parameters in training process. Hence, performance metrics from [44] that account this effect are applied for evaluation. First the proposed algorithm is trained and then tested multiple times. For every training run i of training algorithm, absolute mean error $E(i)$ and standard deviation $Std(i)$ are obtained for N test data as follows [45]:

$$E(i) = \frac{1}{N}\sum_{j=1}^{N}\left|\left(Y_p^i(j) - Y(j)\right)\right| \tag{25}$$

$$Std(i) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(Y_p^i(j) - Y(j)\right)^2} \tag{26}$$

where $Y^i(j)$ is the j^{th} output obtained by the i^{th} run and $Y(j)$ is the j^{th} actual output. The performance evaluation algorithm is given as [44]:

```
1) For $i = 1$ to $M$
2) Train the system using training data set
3) Test system using test data set
4) Calculate the absolute mean prediction error $E(i)$
5) Calculate the standard deviation $Std(i)$
6) Next $i$
```

Depending on the above algorithm, the following performance metrics are defined.

A. The Timeliness. The Timeliness is given by global mean of all the M values of $E(i)$:

$$Timeliness = \overline{E} = \frac{1}{M}\sum_{i=1}^{M} E(i) \tag{27}$$

where M is the number of runs, and $E(i)$ is absolute mean error Equation (25) at run i. The perfect predictor score is $Timeliness = 0$. For small value of Timeliness the probability to have a prediction close to real value is significant. On the contrary, if the Timeliness value is large, the probability to have a wrong prediction is very high.

B. The Precision. The Precision is the global mean of all the M values of $Std(i)$:

$$Precision = \overline{Std} = \frac{1}{M}\sum_{i=1}^{M} Std(i) \tag{28}$$

where $Std(i)$ is standard deviation of each running test i. $Precision = 0$ is the perfect score. For a small value of Precision, the probability to have predictions grouped together can be significant. On the contrary, if the Precision value is high, the predictions are dispersed.

C. The Repeatability. The Repeatability is given by the standard deviation of both $E(i)$ and $Std(i)$.

$$Repeatability = \frac{\sigma(Std) + \sigma(E)}{2} \tag{29}$$

where $\sigma(Std)$ and $\sigma(E)$ represent the standard deviation of the M values of $E(i)$ and $Std(i)$ values respectively.

$$\sigma(Std) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\overline{Std} - Std(i)\right)^2} \tag{30}$$

$$\sigma(E) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\overline{E} - E(i)\right)^2} \tag{31}$$

The perfect score is $Repeatability = 0$. This parameter indicates how close the different values of $E(i)$ and $Std(i)$ are grouped or clustered together. For small values of $\sigma(Std)$ and $\sigma(E)$, it means that at each running time i, the model gives the same performance on test set. Repeatability parameter reveals the random initialization influence of some learning parameters. The training process is completely repeatable for small values of Repeatability.

D. The Accuracy. Accuracy is defined as:

$$Accuracy = \frac{1}{Repeatability + \ Timeliness + \ Precision} \tag{32}$$

If a model has a good Timeliness, Precision and is completely Repeatable, then the prediction given in that model is very close to real data. The prediction confidence is very high. A big value of the Accuracy parameter gives a great confidence of prediction.

5.2. **Mackey-Glass time series results.** Mackey-Glass (MG) represents a model for white blood cell production in leukemia patients. MG time series is the solution of differential equation:

$$\frac{dx(t)}{dt} = \frac{\alpha x(t-\tau)}{1 + x^{10}(t-\tau)} - \beta x(t) \tag{33}$$

The series is chaotic which makes it universally acceptable representation of nonlinear oscillations of many physiological processes [47]. For this reason, it is widely used for testing the performance of prediction models [32,45,48]. Fourth-order Rung-Kutta method with time step size 0.1 is used to obtain time series values at integer points. The initial parameters are set to:

$$\tau = 17,\ \alpha = 0.2 \text{ and } \beta = 0.1$$

Input-output is constructed using embedded theorem. The retardation of the series for chaotic time series is described as [46]:

$$X(t) = [x(t-(E_m-1)t_d), \ldots, x(t-t_d), x(t)] \tag{34}$$

where $X(t)$ is the embedded vector, $x(t)$ is the value of the sequence at time t, t_d denotes retardation of the time series and E_m denotes the embedding dimension. The embedded vector $X(t)$ is used to predict sequence values at $x(t+v)$, where v is the prediction step. Here v, t_d and E_m are set to: $v = 1$, $t_d = 1$ and $E_m = 4$.

Embedded vector or input to PFP is set to: $X(t) = [x(t-3), x(t-2), x(t-1), x(t)]$. Similar to [32] 1000 data from $x(124)$ to $x(1123)$ are selected for simulation. The first 500 data are selected for training and remaining for testing. Input output data has the form: $x(t-3)$, $x(t-2)$, $x(t-1)$, $x(t)$; $x(t+1)$. From the training phase the following parameters are obtained.

- The optimum shape of MF occurs at width value 0.08.
- The best operator with minimum MSE from Table 1 is the Yager operator.

Hence, these values are used to test the performance of PFP. The predicted and real value of the MG time series is shown in Figure 5. The difference between predicted and real data is so small that it is not visible with naked eye, unless zoomed. The zoomed portion of one of hills in MG data shows minimum difference between real and predicted

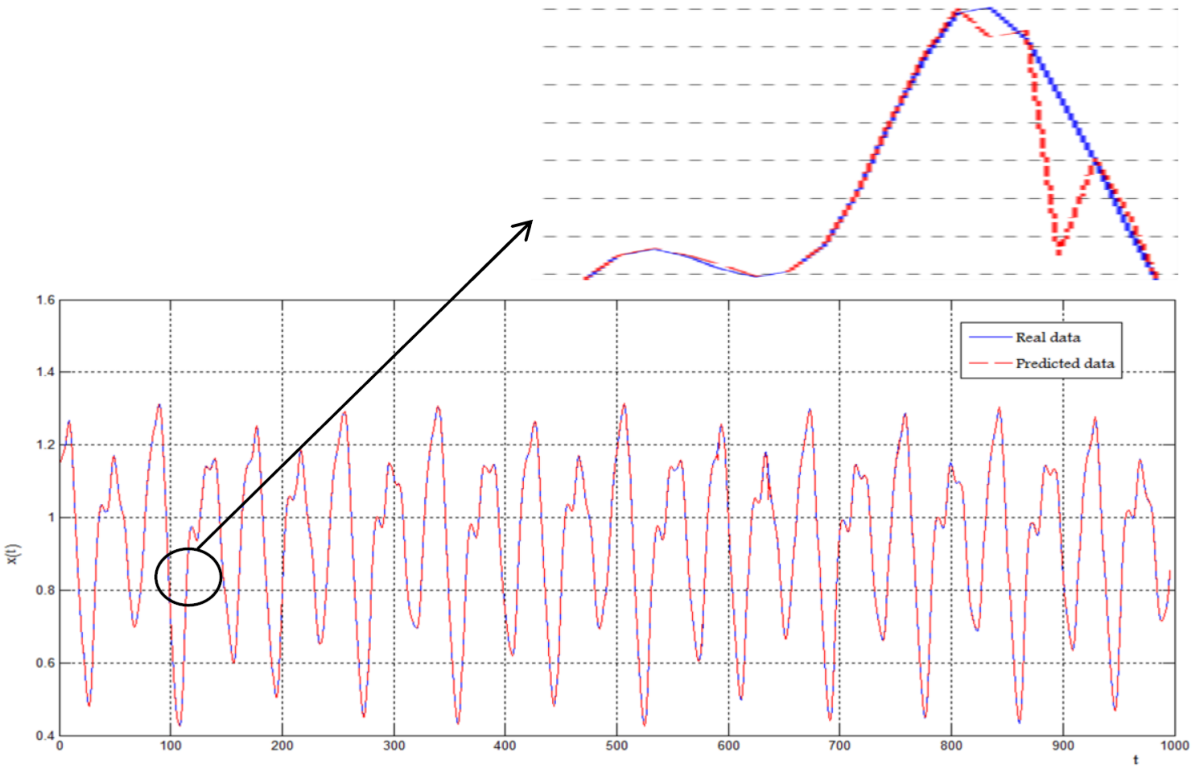

FIGURE 5. Real and predicted output of MG time series

values. For $t < 500$, i.e., for the training data error is almost zero, while for testing data error is less than 0.001 except for some points. The maximum and minimum errors for testing data are 0.0849 and 3.5124×10^{-7}.

Table 2 shows MSE and RMSE of training and testing data. Performance comparison of PFP with other models listed in [32] is shown in Table 3.

Table 2. MG time series results

Performance Evaluation Metrics	Training data	Testing data	Perfect Predictor Score
MSE	6.6120×10^{-7}	8.1314×10^{-4}	0
RMSE	1.1444×10^{-5}	0.0038	0
Timeliness	–	3.1892×10^{-4}	0
Precision	–	2.1753×10^{-5}	0
Repeatability	–	1.2051×10^{-6}	0
Accuracy	–	2925.0472	∞

Table 3. Performance comparison with other models listed in [32]

No of fuzzy set	Root Mean Square Error (RMSE)				
	WANG	MCM-1	MCM-2	Young-Chul	PFP
4	–	–	–	–	0.0038
5	–	–	–	0.0085	–
7	0.0372	0.0374	0.0340	–	–
11	0.0253	0.0265	0.0235	–	–
15	0.0191	0.0197	0.0187	–	–
19	0.0161	0.0162	0.0159	–	–
23	0.0135	0.0142	0.0131	–	–
27	0.0115	0.0115	0.0113	–	–
31	0.0108	0.0108	0.0105	–	–

PFP has high accuracy compared to other models in [32]. This is achieved using 4 linguistic variables. The number of rules is related to the number of fuzzy sets (Equation (20)). From Table 3 one can see that the number of fuzzy set for the PFP is 4, while for others it is more than 4. Hence, the PFP uses lower number of rules than other models. The accuracy is achieved by carefully searching optimum shape of MF and the best T-norms in addition to adopting the preferred methods at each 4 steps of basic time series prediction algorithm.

5.3. **Box-Jenkins gas furnace data simulation.** Box-Jenkins dataset represents CO_2 concentration as output $y(t)$ in terms of input gas flow rate $u(t)$ from a combustion process of a methane-air mixture [46]. The Box-Jenkins dataset consists of 296 input-output measurements. In order to compare PFP to [45] and other models listed in [45], $u(t-1)$, $u(t-2)$, $u(t-3)$ and $y(t-1)$, $y(t-2)$, $y(t-3)$ are selected as input variable. Separate LVs are defined for both input gas flow rate $u(t)$ and CO_2 concentration $y(t)$. The cluster centers are obtained by FCM. For CO_2 concentration linguistic variables, low and high CO_2 concentrations are defined. Similarly, for input gas flow rate low and high flow rates are defined. Cluster centers for input gas flow rate $u(t)$ are -1.0123 and 0.7704.

Cluster center of input CO_2 concentration are 50.8298 and 56.2945. The number of rules using Equation (20) is 64. Results of PFP do not vary from one run to another. Hence, the performance is evaluated using MSE and RMSE. The MSE is 2.5115×10^{-4}, while RMSE is 0.0158. The real and predicted outputs are plotted in Figure 6. The predicted and real outputs are so similar that, even if zoomed, it is difficult to see the difference. Table 4 shows performance comparison of PFP to other models. All data except the PFP is obtained from [45]. The PFP model has the lowest MSE compared to other models. The PFP learns only one time, while other models require 100 epochs to train. The number of epochs affects the speed of the algorithm. As a result, the PFP is a fast predictor compared to other models listed in the table. In general, the PFP has the lowest MSE only learning for an epoch.

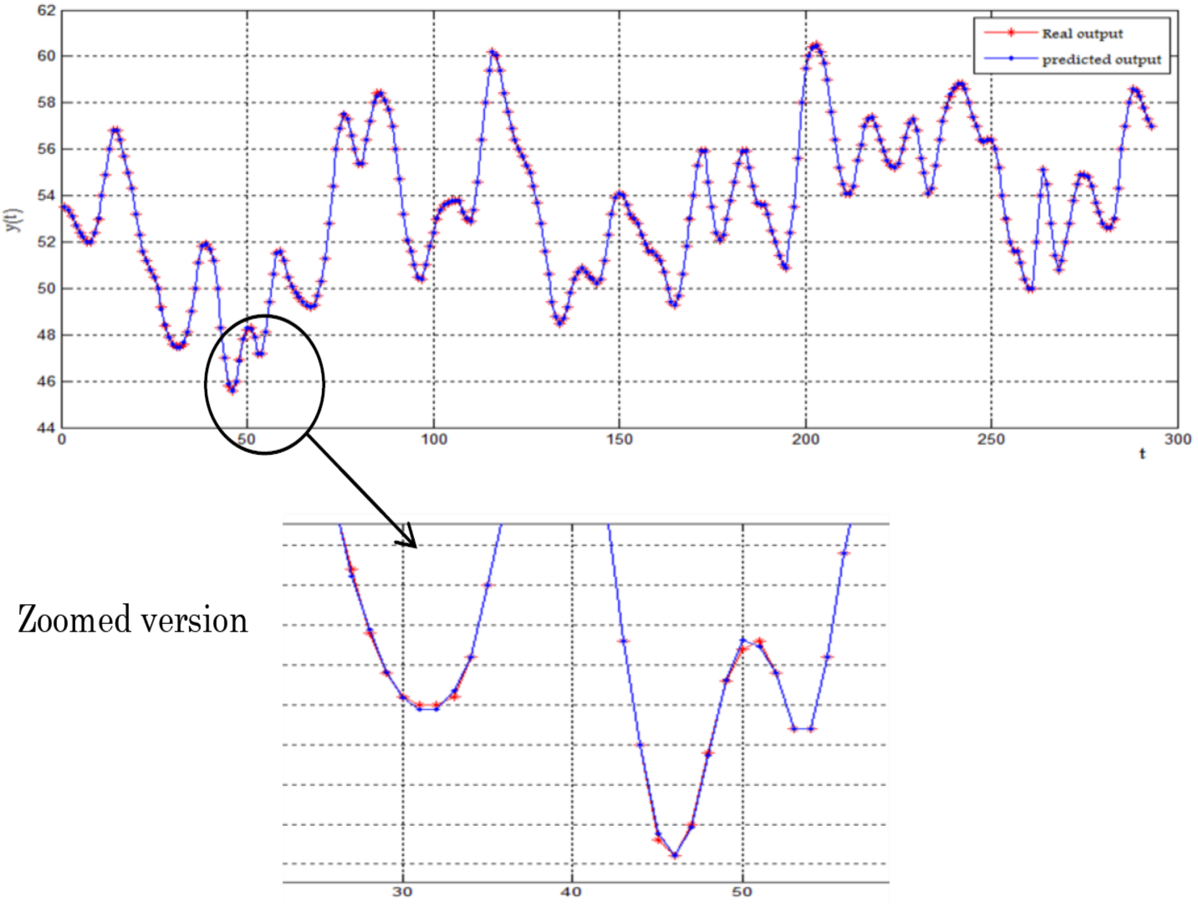

Figure 6. Box-Jenkins dataset prediction

Table 4. Comparison of PFP with other methods

Model	No of training epochs	MSE
[49]	100	0.055
[50]	Not available	0.055
[51]	Not available	0.066
[52]	Not available	0.068
[53]	Not available	0.202
[45]	100	0.0512
PFP	1	0.0021

6. **Conclusion.** In this paper a fuzzy model for predicting non-stationary chaotic time series is proposed. The basic FTS prediction algorithm contains four basic steps. PFP selects the best available methods in each step. Further improvement on the accuracy is obtained by linear search of optimum shape of membership functions and selecting T-norm operators with the lowest MSE on the antecedent of TS fuzzy rules. In cases when the output is outside U, PFP replaces the invalid output with a reasonable value. The performance is evaluated using standard data. It is compared with perfect predictor using MG time series data. PFP is compared with some methods in the literature using Mackey-Glass and Box-Jenkins gas furnace data. PFP has the lowest MSE. In future PFP can be improved in terms of accuracy and in terms of reducing number of rules. The number of rules can be reduced by eliminating rules that have little contribution to the antecedent evaluation. The PFP accuracy can be improved by using type 2 fuzzy sets. Therefore, rule reduction and applying type 2 fuzzy sets are worth of future investigating.

REFERENCES

[1] I. Kim and S.-R. Lee, A fuzzy time series prediction method based on consecutive values, *Proc. of IEEE International Conf. on Fuzzy Systems*, pp.22-25, 1999.

[2] Q. Song and B. S. Chissom, Fuzzy time series and its models, *Fuzzy Sets and Systems*, vol.54, no.3, pp.269-277, 1993.

[3] Q. Song and B. S. Chissom, Forecasting enrollments with fuzzy time series – Part I, *Fuzzy Sets and Systems*, vol.54, 1993.

[4] Q. Song and B. S. Chissom, Forecasting enrollments with fuzzy time series – Part II, *Fuzzy Sets and Systems*, vol.62, 1994.

[5] Y.-H. Yu and L.-X. Song, On fuzzy time series method, *IEEE the 3rd International Conf. on Knowledge Acquisition and Modeling*, pp.297-300, 2010.

[6] C.-C. Tsai and S.-J. Wu, A study for second order modeling of fuzzy time series, *Proc. of IEEE International Conf. on Fuzzy Systems*, pp.719-725, 1999.

[7] W.-K. Wong et al., Adaptive time variant models for fuzzy time series forecasting, *IEEE Trans. Systems, Man and Cybernetics*, vol.40, pp.1531-1542, 2010.

[8] Y.-K. Bang and C.-H. Lee, Fuzzy time series prediction using hierarchical clustering algorithms, *Journal of Expert Systems with Applications*, pp.4312-4325, 2011.

[9] B. Samanta, Prediction of chaotic series using computational intelligence, *Journal of Expert Systems with Applications*, pp.11406-11411, 2011.

[10] G. Zhang et al., Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting*, vol.14, pp.35-62, 1998.

[11] J. V. Tu, Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, *Journal of ANN*, vol.49, pp.1225-1231, 1996.

[12] R. Chandra and S. Chand, Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance, *Applied Soft Computing*, vol.49, pp.462-473, 2016.

[13] E. Momeni, R. Nazir, D. J. Armaghani and H. Maizir, Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN, *Journal of Measurement*, vol.57, pp.122-131, 2014.

[14] N. R. Dhamge, S. L. Atmapoojya and M. S. Kadu, Genetic algorithm driven ANN model for runoff estimation, *Procedia Technology*, vol.6, pp.501-508, 2012.

[15] H. Yang and X. Hu, Wavelet neural network with improved genetic algorithm for traffic flow time series prediction, *International Journal for Light and Electron Optics*, vol.127, pp.8103-8110, 2016.

[16] D. Pradeepkumar and V. Ravi, Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network, *Applied Soft Computing*, vol.58, pp.35-52, 2017.

[17] V. Ravi, D. Pradeepkumar and K. Deb, Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms, *Swarm and Evolutionary Computation*, vol.36, pp.136-149, 2017.

[18] V. Gandhi, G. Prasad, D. Coyle, L. Behera and T. M. McGinnity, Quantum neural network-based EEG filtering for a brain-computer interface, *IEEE Trans. Neural Networks and Learning Systems*, vol.25, pp.278-288, 2014.

[19] R. Zhou, H. Zheng, N. Jiang and Q. Ding, Self-organizing quantum neural network, *Proc. of the 2006 IEEE International Joint Conf. on Neural Network*, 2006.

[20] Y. Cui, J. Shi and Z. Wang, Complex rotation quantum dynamic neural network (CRQDNN) using complex quantum neuron (CQN): Applications to time series prediction, *Journal of Neural Networks*, vol.71, pp.11-26, 2015.

[21] S.-M. Chen, Forecasting enrollments based on fuzzy time series, *Fuzzy Sets and Systems*, vol.81, pp.311-319, 1996.

[22] J.-R. Hwang, S.-M. Chen and C.-H. Lee, Handling forecasting problems using fuzzy time series, *Fuzzy Sets and Systems*, vol.100, pp.217-228, 1998.

[23] K. Huarng, Effective lengths of intervals to improve forecasting in fuzzy time series, *Fuzzy Sets and Systems*, vol.123, pp.387-394, 2001.

[24] S.-M. Chen and N.-Y. Chung, Forecasting enrollments using high order fuzzy time series and genetic algorithms, *International Journal of Intelligent Systems*, vol.21, pp.485-501, 2006.

[25] C.-D. Chen and S.-M. Chen, A new method to forecast the TAIEX based on fuzzy time series, *IEEE International Conf. on Systems, Man, and Cybernetics*, 2009.

[26] S.-M. Chen and N.-Y. Chung, Forecasting enrollments of students by using fuzzy time series and genetic algorithms, *Information and Management Sciences*, vol.17, pp.1-17, 2006.

[27] C.-H. Cheng, J.-R. Chang and C.-A. Yeh, Entropy based and trapezoid fuzzification based fuzzy time series approaches for forecasting IT project cost, *Technological Forecasting & Social Change*, vol.73, pp.524-542, 2006.

[28] R. Chang et al., Cardinality based fuzzy time series for forecasting enrollments, *Fuzzy Sets and Systems*, 2007.

[29] S.-T. Li, Y.-C. Cheng and S.-Y. Lin, A FCM based deterministic forecasting model for fuzzy time series, *Computers and Mathematics with Applications*, vol.56, pp.3052-3063, 2008.

[30] F. Radmehr and N. S. Gharneh, Using a simulated annealing-based approach for forecasting first order fuzzy time series, *Australian Journal of Basic and Applied Sciences*, vol.5, no.6, pp.201-209, 2011.

[31] M. H. Fazel et al., Fuzzy time series based on defining interval length with imperialist competitive algorithm, *IEEE Annual Meeting of North American Fuzzy Information Processing Society*, 2010.

[32] Y.-K. Bang and C.-H. Lee, Fuzzy time series prediction with data preprocessing and error compensation based on correlation analysis, *IEEE the 3rd International Conf. on Convergence and Hybrid Information Technology*, vol.2, pp.714-721, 2008.

[33] S.-T. Li et al., Deterministic vector long term forecasting for fuzzy time series, *Fuzzy Sets and Systems*, vol.161, pp.1852-1870, 2010.

[34] J. C. Bezdek et al., FCM: The fuzzy C-means clustering algorithm, *Computers and Geosciences*, vol.10, pp.191-203, 1984.

[35] L. T. Koczy et al., Fuzzy rule extraction from input-output data, *Advances in Fuzzy System Application and Theory*, vol.21, pp.199-216, 2004.

[36] R. A. Rutenbar, Simulated annealing algorithms: An overview, *IEEE Circuits and Devices Magazine*, vol.5, pp.19-26, 1989.

[37] P. D'Ursoa and A. Maharajb, Autocorrelation-based fuzzy clustering of time series, *Fuzzy Sets and Systems*, vol.160, pp.3565-3589, 2009.

[38] J.-R. Chang and Y.-J. Huang, A weighted fuzzy time series model based on adoptive OWA operators, *IEEE International Conf. on Uncertainty Reasoning and Knowledge Engineering*, 2011.

[39] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Addison Wesley Longman Inc., 1998.

[40] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models*, The MIT Press, 2001.

[41] G. J. Klir, *Fuzzy Logic: A Specialized Tutorial*, Burlington Academic Press, 2004.

[42] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems, Man and Cybernetics*, vol.15, 1985.

[43] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*, Pearson Education Limited, 2005.

[44] R. Zemouri, R. Gouriveau and N. Zerhouni, Defining and applying prediction performance metrics on a recurrent NARX time series model, *Neurocomputing*, vol.73, pp.2506-2521, 2010.

[45] B. Rezaee and F. Zarandi, Data driven fuzzy modeling for Takagi-Sugeno-Kang fuzzy system, *Information Sciences*, vol.180, pp.241-255, 2010.

[46] B. Samanta, Prediction of chaotic time series using computational intelligence, *Expert Systems with Applications*, vol.38, pp.11406-11411, 2011.

[47] X. Yin et al., Prediction of chaotic time series of neural network and an improved algorithm, *IEEE the 5th International Conference on Bio-Inspired Computing*, pp.1282-1286, 2010.

[48] R. N. Yadvav, P. K. Kalra and J. John, Time series prediction with single multiplicative neuron model, *Applied Soft Computing*, vol.7, pp.1157-1163, 2007.

[49] A. E. Gaweda and J. M. Zurada, Data-driven linguistic modeling using relational fuzzy rules, *IEEE Trans. Fuzzy Systems*, vol.11, pp.121-134, 2003.

[50] E. Kim, M. Park, S. Ji and M. Park, A new approach to fuzzy modeling, *IEEE Trans. Fuzzy Systems*, vol.5, pp.318-329, 1997.

[51] H. Wang et al., Agent-based evolutionary approach for interpretable rule-based knowledge extraction, *IEEE Trans. System, Man, and Cybernetics, Part C (Applications and Reviews)*, vol.35, no.2, pp.143-155, 2005.

[52] M. Sugeno and K. Tanaka, Successive identification of a fuzzy model and its application to prediction of complex system, *Fuzzy Set Systems*, vol.42, pp.315-334, 1991.

[53] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, Holden Day, 1970.