IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 24 September 2023, accepted 8 November 2023, date of publication 13 November 2023, date of current version 27 November 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3332123*

RESEARCH ARTICLE

# A Novel Fuzzy Time Series Method Based on Dynamic Ridge Polynomial Neural Network With Penalty Term and Fuzzy Clustering Analysis

**BOYANG WANG[1,2], XISONG MIAO[1,2], HUANYU WEI[1], MD. GOLAM SAKLAIN[1], YINAN ZHI[1], HONGYAN JIN[1], AND JIAXIE LI[1]**

[1]College of Information Engineering, Kunming University, Kunming 650214, China
[2]Key Laboratory of Data Governance and Intelligent Decision, Universities of Yunnan, Kunming 650214, China

Corresponding author: Xisong Miao (miaoxisong_kmu@163.com)

This work was supported in part by the Local Universities Joint Special Program-Funded Project: Research on Spatio-Temporal Evolution Analysis Models and Algorithms of Online Public Opinion Based on Complex Networks under Grant 202301BA070001-017, in part by the Special Basic Cooperative Research Programs of Yunnan Provincial Undergraduate Universities Association under Grant 202101BA070001-149, in part by the National Natural Science Foundation of China under Grant 62066023, and in part by the Yunnan Students' Innovation and Entrepreneurship Training Program under Grant S202311393011 and Grant S202311393001.

**ABSTRACT** Due to the limitations of traditional time series models in handling semantic values and small-scale data, the concept of fuzzy time series forecasting has been introduced in academia. This model performs exceptionally well on fuzzy datasets, prompting many researchers to delve into this field. The general process of fuzzy time series analysis consists of the following stages: 1) domain partitioning; 2) formation of fuzzy sets for fuzzifying data; 3) extraction of fuzzy relationships; and 4) forecasting and defuzzification. Domain partitioning and the extraction of fuzzy relationships have always been crucial components of fuzzy time series forecasting. Until now, neural networks have been less commonly applied in the step of determining fuzzy relationships. Some researchers have attempted to utilize the Pi-Sigma neural network for the determination of fuzzy relationships. However, due to the fixed network structure that Pi-Sigma neural networks cannot adapt to changes over time, it has been indicated that it is not a universal approximator. Its performance in handling complex dynamic time series has not been satisfactory. In this paper, we utilize Fuzzy C-Means Clustering (FCM) to partition the domain into unequal-length intervals and employ a high-order dynamic neural network known as Dynamic Ridge Polynomial Neural Network (DRPNN). This network can start with a small basic structure and gradually increase its structural complexity as learning progresses until it achieves the required task accuracy, which demonstrates superior performance in handling complex time series data. During the training process, we employ a novel gradient descent training algorithm with penalty terms. We conducted tests on this algorithm using nine real-world datasets and performed Friedman and Bonferroni-Dunn tests to ensure that the proposed algorithm exhibits statistical performance superiority compared to other methods in the literature. The results indicate that our algorithm outperforms those from other studies.

**INDEX TERMS** Dynamic ridge polynomial neural network, fuzzy time series, fuzzy C-means clustering, penalty term.

## I. INTRODUCTION

Time series data refers to data collected over time, describing the process and patterns of change in things. It typically refers to precise data with general patterns, such as population, economy, food production, etc. By analyzing and discovering the inherent laws of the data, people try to make predictions about possible future occurrences as accurately as possible. In recent years, many researchers have applied time series

The associate editor coordinating the review of this manuscript and approving it for publication was Zhigao Zheng.

prediction in different fields, and have also proposed many methods.

It is believed that the origin of time series analysis can be traced back to the Auto Regressive (AR) model proposed by British statistician G. U. Yule in 1927. Subsequently in 1931, Walker established the MA (Moving Average) model and ARMA (Autoregressive Moving Average) model inspired by the AR model. Thanks to the research by many scientists, the 'Box-Jenkins' method [1] proposed by Box and Jenkins pioneered the traditional time series prediction; Hamilton [2], Durbin and Koopman [3], Aly [4] and others also proposed using state space equations and Kalman filters for time series prediction. Although these classical methods can be used for various time series predictions, they all require time series to be univariate, homoscedastic linear models, normally distributed, etc. It has been found that these assumptions do not hold in some cases, and these models may not be able to predict complex real-world time series. With the rise of machine learning and artificial neural networks (ANN), classical SVM, BN have also achieved good results in time series prediction. Al-Hadeethi et al. [5] applied SVM to signal time series for epilepsy seizure diagnosis. Zhao et al. [6] and others proposed using Bayesian ensemble algorithms to detect time series in satellites. Farahani and Hajiagha [7] and Ali et al. [8], among others, used ensemble ANN to predict stock prices and financial time series. Jin et al. [9] used Bayesian networks for the prediction of massive time series data. Although the above methods can handle most real problems, some problems still cannot be solved. These problems usually contain fuzzy and unclear uncertainties. In recent years, many time series prediction methods tend to be based on fuzzy methods based on fuzzy set theory.

In 1965, Negoita [10] proposed the concept of fuzzy theory and fuzzy logic, and initially established a model for problems with uncertain, fuzzy linguistic variables. In 1994, Song and Chissom built a prediction model for fuzzy time series based on this [11] and [12], pioneering the theoretical and applied research on fuzzy time series. Many researchers have made tremendous efforts in different steps of fuzzy time series prediction. For example, in fuzzy partition of domains, there were initially average partition methods represented by Song [11] and [12], Chen [13] and Lee et al. [14]; subsequently, scholars represented by Huarng and Yu [15], Chen et al. [16] and Yan et al. [17] proposed methods to partition domains based on the distribution of sample data; currently common methods are clustering algorithms [18], [19], [20], [21], [22], [23], which utilize algorithms to cluster sample data and determine partition based on clustering results. In recent years' research on fuzzy time series prediction, many scholars have also improved Song's method [24], [25], [26], [27], [28], [29] and achieved good prediction results.

When dealing with complex fuzzy time series, we can use different types of ANN models in the analysis process to extract and analyze the time series, such as Multilayer Perceptron (MLP) with additive neuron structure, Multiplicative Neuron Model Neural Network (MNM-ANN) with multiplicative neuron structure. However, MLP and some other neural network models have relatively complex architectures that require a lot of weights to be obtained through training, which leads to slower convergence when dealing with complex problems [30]. Therefore, in recent years, Bas et al. [31] and others have proposed using Pi-Sigma (PSNN), a high-order neural network with fewer weights, to determine fuzzy relationships and this network has fewer weights compared to other neural networks, allowing it to converge more quickly. However, PSNN cannot dynamically change its network structure over time, for some more complex nonlinear time series, the accuracy of prediction by a single neural network may be affected. DRPNN [32] neural network is a generalization of Pi-Sigma which retains the advantage of fewer weights of PSNN, and can better predict complex time series.

In this research, Firstly, we propose a novel approach that utilizes a high-order neural network known as DRPNN to define fuzzy relationships in high-order fuzzy time series. Next, we employ FCM clustering for domain partitioning. an improved gradient training algorithm based on penalty terms to train the network weights. Finally, we apply this approach to nine commonly used real-time series datasets, comparing the results with other methods and conducting Friedman and Bonferroni-Dunn tests to demonstrate the statistical performance superiority of the proposed method.

The structure of this paper is as follows: Section two introduces fuzzy time series. Section three presents the FCM clustering algorithm. Section four introduces the DRPNN neural network, while Section five presents an improved gradient descent algorithm based on penalty terms. Section six introduces the newly proposed method. Section seven carries out experimental testing, comparing it with other literature methods and conducting Friedman and Bonferroni-Dunn tests. The final section summarizes the conclusions of this study and discusses some potential future methods for fuzzy time series forecasting.

## II. FUZZY TIME SERIES MODEL

Fuzzy time series were first proposed by Song and Chinssom (1993), who regarded dynamic processes with semantic values as fuzzy time series and described the fuzzy time series by fuzzy relational equations.

*Definition 1:* Suppose $U$ is the universe of discourse, $U = \{u_1, u_2, u_3 \ldots, u_n\}$, The fuzzy set $A$ on $U$ is defined as follows:

$$A = \frac{f_A(u_1)}{u_1} + \frac{f_A(u_2)}{u_2} + \frac{f_A(u_3)}{u_3} + \ldots + \frac{f_A(u_n)}{u_n} \tag{1}$$

where $f_A(\cdot)$ is the affiliation function of the fuzzy set $A$, $f_A : U \to [0, 1]$, $f_A(u_i)$ denotes the affiliation of $u_i$ on the fuzzy set $A$.

*Definition 2:* Let $Y(t)$ be the universe of discourse on which fuzzy sets $f_i(t)$ $(i = 1, 2, 3, \ldots)$ are defined, if $F(t)$ is

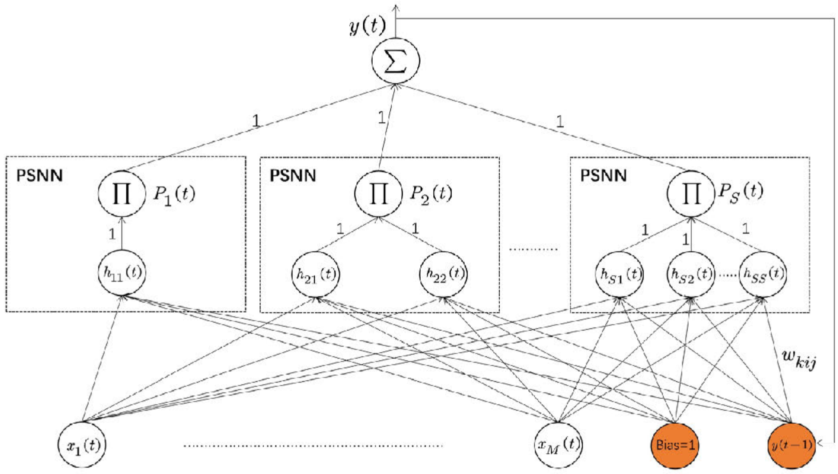



**FIGURE 1.** DRPNN architecture diagram.

**TABLE 1.** Definitions of variables.

| Variables | Definitions |
|---|---|
| $T_k(t)$ | Overall input function at time $t$ |
| $x_k(t)$ | *k-th* initial input at time $t$ |
| $y(t-1)$ | Output of the network at the previous moment |
| $P_i(t)$ | Output of the *i-th* PSNN unit at time $t$ |
| $f(\cdot)$ | Nonlinear activation function |
| $S$ | Total number of PSNN units |
| $h_{ij}(t)$ | Output value of the *j-th* node in the *i-th* PSNN block at time $t$ |
| $M$ | Number of initial inputs |
| $w_{kij}(t)$ | Weight between the *j-th* node in the *i-th* PSNN unit and the *k-th* input node at time $t$ |
| $Q$ | Length of the test set |
| $H(t)$ | Mean square error function at time $t$ |
| $E(t)$ | Mean square error function with penalty terms at time $t$ |
| $\lambda(t)$ | Dynamic penalty factor |
| $c$ | Scaling factor |
| $\eta$ | Learning rate |

the set of $f_i(t)$, then $F(t)$ is called a fuzzy time series defined on $Y(t)$.

*Definition 3:* If there is a fuzzy logical relation $F(t)$ that can be caused by $F(t-1)$, then this relation can be defined as: $F(t-1) \rightarrow F(t)$, and this model is called a first-order fuzzy time series model.

*Definition 4:* If $F(t)$ can be caused by $F(t-1), F(t-2), \ldots, F(t-n)$, the $n_{th}$ order fuzzy time series model can be defined as $F(t-n), F(t-n+1), \ldots, F(t-1)$.

**Fuzzy C-Means Clustering**

The Fuzzy C-Means Clustering (FCM) algorithm is a clustering algorithm that uses affiliation to determine the degree to which each data point belongs to a certain cluster. It was proposed by Bezdek et al. [33] and improves on the earlier hard clustering (HCM) method. It takes into account the ambiguity of attributes and considers that samples can belong to more than one class at the same time, but have different degrees of subordination to these classes, i.e., different degrees of affiliation.

FCM takes into account the fuzzy characteristics and distribution characteristics of the data points, so the results obtained by FCM can well reflect the aggregation characteristics of the data, and the objective function and constraints of FCM clustering are as follows:

$$J_\beta(u, V) = \sum_{i=1}^{N} \sum_{j=1}^{C} (u_{ij})^m \parallel x_i - v_j \parallel^2 \tag{2}$$

$$st.\begin{cases} 0 \le u_{ij} \le 1, \forall i, j \\ \sum_{j=1}^{C} u_{ij} = 1, \forall j \\ 0 \le \sum_{i=1}^{N} u_{ij} \le N, \forall i \end{cases} \tag{3}$$

where, $x_i$ is the data value, $N$ is the number of data points, $C$ is the number of clusters, $m$ is the fuzzy coefficient, m $\in (1, \infty)$ , $u_{ij}$ represents the degree of membership of the $i_{th}$ data point belonging to the $j_{th}$ cluster, $V$ represents the set of cluster centers, $v_j$ represents the cluster center of the $j_{th}$ cluster, $\|\cdot\|$ is the Euclidean norm.

The FCM clustering algorithm uses an iterative approach to obtain the optimal cluster centers. The iteration formula is as follows:

$$v_j^{(q)} = \frac{\sum_{i=1}^{N} \left(u_{ij}^{(q)}\right)^m x_i}{\sum_{i=1}^{N} \left(u_{ij}^{(q)}\right)^m} \tag{4}$$

$$u_{ij}^{(q+1)} = \frac{1}{\sum_{k=1}^{C} \left( \frac{\left\| x_i - v_j^{(q)} \right\|}{\left\| x_i - v_k^{(q)} \right\|} \right)^{\frac{2}{m-1}}} \tag{5}$$

where $q$ represents the number of iterations. The iteration stopping criteria is that the objective function is less than a given value or the number of iterations reaches a given value.

## III. DYNAMIC RIDGE POLYNOMIAL NEURAL NETWORK

Dynamic Ridge Polynomial Neural Network (DRPNN) is a recurrent neural network that not only has the expanded architecture and functionality of RPNN [34], but also incorporates feedback connections from the output layer to the input layer. Due to the addition of feedback connections, compared to RPNN and many other current higher order neural networks, DRPNN can better simulate the dynamic changes of fuzzy time series [35], [36].

DRPNN uses an asynchronous update rule based on Pi-Sigma units, and consists of multiple PSNNs acting as hidden layer nodes. DRPNN starts from a small basic structure, and as the number of iterations increases, PSNNs with increasing order are added to the hidden layer of DRPNN to achieve growth in network complexity, until the requirements of the mapping task are met.

The architecture of DRPNN is shown in Fig.1. The feedback connections between the input layer and output layer feedback the network's output to the PSNNs in the hidden layer, thereby learning the output result of the previous sample. Regarding the connection weights, only the weights between the input layer and hidden layer are learnable, the rest of the weights are 1, which reduces the complexity of the network and can mitigate the risk of overfitting to some extent.

The input function is as follows:

$$T_k(t) = \begin{cases} x_k(t), & if\ 1 \le k \le M \\ 1, & if\ k = M+1 \\ y(t-1), & if\ k = M+2 \end{cases} \tag{6}$$

The output function is as follows:

$$h_{ij} = \sum_{k=1}^{M} w_{kij}(t) T_k(t) + w_{(M+1)ij} + y(t-1)\, w_{(M+2)ij} \tag{7}$$

$$P_i(t) = \prod_{j=1}^{i} h_{ij}(t) \tag{8}$$

$$y(t) = f\left(\sum_{i=1}^{S} P_i(t)\right) \tag{9}$$

### A. GRADIENT DESCENT TRAINING ALGORITHM WITH REGULARIZATION TERM

In the training process of DRPNN, we used a gradient descent algorithm with a penalty term, which accelerated the network's iteration speed and reduced the risk of overfitting.

Assuming $D$ is the desired output of the network, the root mean square error (RMSE) function with a penalty term is as follows:

$$e(t) = D - y(t) \tag{10}$$

$$E(t) = \frac{1}{2Q} \sum_{q=1}^{Q} \left[ \left(e^{(q)}(t)\right)^2 + \lambda(t) \|w(t)\|^2 \right] \tag{11}$$

Taking into account the issue of underfitting caused by an excessively large penalty term, we used a dynamic penalty factor, and the dynamic penalty factor function is as follows:

$$\lambda(t) = c\lambda(t-1) \tag{12}$$

From equation (11), the gradient algorithm with a penalty term is as follows:

$$\begin{aligned} \Delta w_{kij} &= -\eta \frac{\partial E(t)}{\partial w_{kij}} \\ &= -\eta \frac{1}{Q} \sum_{q=1}^{Q} \left[ -e^{(q)}(t) f'\left(\sum_{i=1}^{S} P_i(t)\right) \frac{\partial P_i(t)}{\partial w_{kij}} + \lambda(t) w_{kij}(t) \right] \end{aligned} \tag{13}$$

where

$$\frac{\partial P_i(t)}{\partial w_{kij}} = \frac{\partial P_i(t)}{\partial h_{ij}(t)} \frac{\partial h_{ij}(t)}{\partial w_{kij}} \tag{14}$$

From (6) and (14), we can deduce the following:

$$\frac{\partial h_{ij}(t)}{\partial w_{kij}} = \begin{cases} x_k(t) + w_{(M+2)ij}(t)\,\dfrac{\partial y(t-1)}{\partial w_{kij}}, & if\, 1 \le k \le M \\ 1 + w_{(M+2)ij}(t)\,\dfrac{\partial y(t-1)}{\partial w_{kij}}, & if k = M+1 \\ y(t-1) + w_{(M+2)ij}(t)\,\dfrac{\partial y(t-1)}{\partial w_{kij}}, & if k = M+2 \end{cases} \tag{15}$$

From (14) and (15), we can deduce the following:

$$\frac{\partial P_i(t)}{\partial w_{kij}} = \prod_{u=1,u\neq j}^{i} h_{iu}(t)\left[T_k(t) + w_{(M+2)ij}(t)\,\frac{\partial y(t-1)}{\partial w_{kij}}\right] \tag{16}$$

Substituting (15) and (16) into equation (14), we finally obtain:

$$\begin{aligned}\Delta w_{kij} &= -\eta\frac{1}{Q}\sum_{q=1}^{Q}\Bigg\{-\left(e^{(q)}(t)\right)\left(f'\left(\sum_{i=1}^{S}P_i(t)\right)\left(\prod_{u=1,u\neq j}^{i}h_{iu}(t)\right)\right)\\ &\quad\times\left[T_k(t)+\left(w_{(M+2)ij}(t)\right)\left(\frac{\partial y(t-1)}{\partial w_{kij}}\right)\right]+\lambda(t)\,w_{kij}(t)\Bigg\}\end{aligned} \tag{17}$$

## IV. THE PROPOSED METHOD

This study proposes a novel approach. First, it utilizes the FCM clustering method for domain partitioning based on data fuzziness to obtain optimal cluster centers. This ensures that dispersed data points are gathered around the cluster centers to the greatest extent possible. Compared to previous methods with uniform domain partitioning, this non-uniform domain partitioning can more effectively extract data distribution characteristics, resulting in the formation of optimal fuzzy sets.

Next, it employs DRPNN to define fuzzy relationships in high-order fuzzy time series and utilizes an improved gradient descent algorithm with a penalty term to train network weights. We compared this approach with existing methods and found that it offers higher prediction accuracy and better statistical performance.

Below, we present the specific steps of this algorithm and a numerical example to illustrate its implementation.

**Step 1. Utilize the Fuzzy C-Means (FCM) clustering algorithm to partition the domain.**

Algorithm 1 demonstrates clustering the dataset using the Fuzzy C-Means (FCM) algorithm and determining the domain partition based on the clustering results.

**Step 2. Fuzzify the dataset.**

Based on the domain $A$ obtained in Step 1, obtain the fuzzy sets $< H_1, H_2 \ldots H_C >$. Here, $C$ represents the number of clusters, and triangular functions are used to define the fuzzy sets. The expression is as follows:

$$H_1 = \frac{1}{a_1} + \frac{0}{a_2} + \frac{0}{a_3} + \frac{0}{a_4} + \ldots + \frac{0}{a_C}$$

$$H_2 = \frac{0}{a_1} + \frac{1}{a_2} + \frac{0}{a_3} + \frac{0}{a_4} + \ldots + \frac{0}{a_C}$$

$$\ldots$$

$$H_C = \frac{0}{a_1} + \frac{0}{a_2} + \frac{0}{a_3} + \ldots + \frac{0}{a_{C-1}} + \frac{0}{a_C} \tag{18}$$

**Algorithm 1** FCM Cluster

**Input:** Number of data points ($N$), Number of clusters ($C$), Number of iterations ($Q$)
**Output:** Domain set $A$

//Initialize membership degree $\gamma_{ij}^{(0)}$ for each data point $x_i$ across different classes.
**for** $i = 1$ *to* $N$ **do**
  **for** $j = 1$ *to* $C$ **do**
  Initialize the membership degree vector: $\gamma_{ij} = -1.0 + (1.0 - (-1.0)) * rand\,()\,/RAND\_MAX$
  **end for**
**end for**
**for** $i = 1$ *to* $Q$ **do**
  **for** $j = 1$ *to* $C$ **do**
    **for** $z = 1$ *to* $N$ **do**
      Update the cluster centers $v_j$
    **end for**
  **end for**
  **for** $j = 1$ *to* $N$ **do**
    **for** $z = 1 to C$ **do**
      Update the membership degree vector $u_{iz}$
    **end for**
  **end for**
**end for**
//Sort $v_j$ in ascending order
**for** $i = 1$ *to* $C$ **do**
  Determine the domain boundaries $o_i \leftarrow (v_i + v_{i+1})\,/2.0$, partition the domain into different subintervals using the domain boundaries.
**end for**
return $A$

When fuzzifying data points in the dataset, you assign each data point to the fuzzy set for which it has the highest membership degree. For example, if a data point has the highest membership degree in interval 1, then the corresponding fuzzy set for that data point is $H_1$. Through the steps described above, you have transformed the time series into a fuzzy time series with fuzzy attributes.

**Step 3. Extract logical fuzzy relationships.**

The extraction of logical fuzzy relationships depends on confirming different orders for various categories within the time series dataset. For instance, a second-order logical fuzzy relationship can be denoted as $H_{i1}, H_{i2} \rightarrow H_j$

**Step 4. Train the DRPNN network.**

Use the subscripts representing the antecedents of logical fuzzy relationships as input values for training data, and use the subscripts representing the fuzzy set in the consequent as output values for training data. This process will result in a trained DRPNN neural network.

**Step 5. Prediction.**

Divide the time series into a training set and a prediction set. Similarly, use the antecedents of the prediction set's data as input values for prediction, and use the output values as the indices of fuzzy sets to complete the prediction.

**Step 6. Defuzzification.**

If the obtained output is 1, it signifies that the prediction corresponds to fuzzy set $H_1$. In such a case, you can choose the midpoint of this subinterval as the defuzzied prediction value. The defuzzification process for other output results follows a similar approach.

**Step 7. Evaluate the effectiveness of the model.**

This study employed three evaluation metrics to assess the effectiveness of the model, namely the root mean square error (RMSE), symmetric mean absolute percentage error (SMAPE), and mean absolute scaled error (MASE). The formulas for these metrics are represented as (19), (20), and (21).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{19}$$

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{y}_i\right|}{\left(\left|y_i\right| + \left|\hat{y}_i\right|\right)/2} \tag{20}$$

$$MASE = \frac{MAE}{MAE_{naive}} \tag{21}$$

$n$ represents the dataset size, $y_i$ represents the actual output, and $\hat{y}_i$ represents the standard output.

### A. NUMERICAL EXAMPLE

#### 1) STEP 1. UTILIZE THE FUZZY C-MEANS (FCM) CLUSTERING ALGORITHM TO PARTITION THE DOMAIN

First, we need to use the FCM clustering algorithm to partition the domain of the dataset in the SunSpot dataset, $N = 288, C = 7, Q = 250$, First, obtain 7 cluster centers and arrange them in ascending order: $\{7.89, 24.16, 42.16, 64.30, 85.51, 110.91, 149.96\}$. According to Algorithm 1, the subintervals of the domain partition are as follows:

$$a_1 = [0, 16.021]$$
$$a_2 = [16.021, 33.16]$$
$$a_3 = [33.16, 53.23]$$
$$a_4 = [53.23, 74.90]$$
$$a_5 = [74.90, 98.21]$$
$$a_6 = [98.21, 130.43]$$
$$a_7 = [130.43, 190.2]$$

#### 2) STEP 2. FUZZIFY THE DATASET

According to Equation (18), each data point is assigned the fuzzy set for which it has the highest membership degree. For example, if the value of data point 1 is 5 and it has the highest membership degree in subinterval $a_1$, then the corresponding fuzzy set for that data point should be $H_1$. Through this step, we perform fuzzification on all data points in the dataset.

#### 3) STEP 3. EXTRACT LOGICAL FUZZY RELATIONSHIPS

In this data experiment, a fifth-order fuzzy logic relationship was used for prediction, which means there are 5 antecedents and 1 consequent. Table 2 is obtained based on Definition 4.

#### 4) STEP 4. TRAIN THE DRPNN NETWORK

In this experiment, the dataset is divided into a training set with 245 data points and a test set with 43 data points.

Input nodes $M = 7$, the number of PSNN units $S = 1$, learning rate $\eta = 0.01$, the initial value of the dynamic penalty factor $\lambda(t) = 100$, the reduction coefficient $c = 0.9$. Obtain the trained DRPNN network.

#### 5) STEP 5. PREDICTION

Input the antecedents of the test set as input data into the network, and the output values should correspond to the indices of the predicted fuzzy sets.

#### 6) STEP 6. DEFUZZIFICATION

If the obtained output is 1, it signifies that the prediction corresponds to fuzzy set $H_1$. In such a case, you can choose the midpoint of this subinterval as the defuzzied prediction value. The defuzzification process for other output results follows a similar approach.

#### 7) STEP 7. EVALUATE THE EFFECTIVENESS OF THE MODEL

Three evaluation metrics were employed to assess the effectiveness of the model. Table 3 presents the test data from the SunSpot dataset and the predicted data from different methods. Fig.2. provides a visual representation of the prediction performance of various methods through a line chart.

## V. EMPIRICAL ANALYSIS AND DISCUSSION

To conduct experimental validation, this study considered 9 time series datasets with different characteristics, as shown in Table 4. These datasets were sourced from the time series data library [37] and included the Taiwan Capitalization Weighted Stock Index (TAIEX) [38] closing data from 2006 to 2008. Table 5 provides the statistical features of each dataset. To assess the effectiveness of the proposed method, it was compared with five existing methods, and statistical analysis demonstrated the superior performance of the proposed method.

We divided the original dataset into two parts: a training set and a test set. Due to the varying characteristics of each dataset, the training input order differs, as indicated in Table 6. Regarding the prediction results, we first analyzed the results using three different performance metrics.

### A. THE EFFICIENCY METRIC RMSE

This section analyzes the prediction accuracy of different methods in different datasets using the RMSE metric. RMSE is sensitive to outlier data and is also dependent on the dataset's scale. Table 7 presents the average RMSE results

**TABLE 2.** Notations for fifth order fuzzy time series.

| $F(t-5)$ | $F(t-4)$ | $F(t-3)$ | $F(t-2)$ | $F(t-1)$ | $F(t)$ | Input1 | Input2 | Input3 | Input4 | Input5 | Target |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $H_1$ | $H_1$ | $H_1$ | $H_2$ | $H_3$ | $H_4$ | 1 | 1 | 1 | 2 | 3 | 4 |
| $H_1$ | $H_1$ | $H_2$ | $H_3$ | $H_4$ | $H_2$ | 1 | 1 | 2 | 3 | 4 | 2 |
| $H_1$ | $H_2$ | $H_3$ | $H_4$ | $H_2$ | $H_2$ | 1 | 2 | 3 | 4 | 2 | 2 |
| $H_2$ | $H_3$ | $H_4$ | $H_2$ | $H_2$ | $H_1$ | 2 | 3 | 4 | 2 | 2 | 1 |
| | | | | | ... | | | | | | |

**TABLE 3.** Forecasts of different methods on test set of sunspot time series.

| Time series Date | Aladag et al. [39] | Aladag [40] | Bisht and Kumar [41] | Bas et al. [31] | Gupta and Kumar [42] | proposed method |
|---|---|---|---|---|---|---|
| 33.2 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 92.6 | 75.0 | 95.0 | 94.2 | 95.0 | 83.2 | 86.6 |
| 151.6 | 135.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 136.3 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 160.3 |
| 134.7 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 160.3 |
| 83.9 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 69.4 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 86.6 |
| 31.5 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 13.9 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 24.6 |
| 4.4 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 24.6 |
| 38.0 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 141.7 | 75.0 | 95.0 | 94.2 | 95.0 | 83.2 | 86.6 |
| 190.2 | 165.0 | 95.0 | 94.2 | 95.0 | 83.2 | 160.3 |
| 184.8 | 75.0 | 95.0 | 90.6 | 95.0 | 83.2 | 160.3 |
| 159.0 | 75.0 | 95.0 | 90.6 | 95.0 | 83.2 | 160.3 |
| 112.3 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 53.9 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 86.6 |
| 37.5 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 27.9 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 10.2 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 24.6 |
| 15.1 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 24.6 |
| 47.0 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 93.8 | 165.0 | 95.0 | 94.2 | 95.0 | 83.2 | 86.6 |
| 105.9 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 105.5 | 75.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 104.5 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 66.6 | 75.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 68.9 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 64.1 |
| 38.0 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 64.1 |
| 34.5 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 15.5 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 12.6 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 27.5 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 92.5 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 64.1 |
| 155.4 | 165.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 154.7 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 160.3 |
| 140.5 | 75.0 | 95.0 | 94.2 | 95.0 | 83.2 | 160.3 |
| 115.9 | 105.0 | 95.0 | 94.2 | 95.0 | 83.2 | 114.3 |
| 66.6 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 86.6 |
| 45.9 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 17.9 | 45.0 | 95.0 | 94.2 | 95.0 | 83.2 | 43.2 |
| 13.4 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 24.6 |
| 29.2 | 15.0 | 95.0 | 94.2 | 95.0 | 83.2 | 24.6 |
| **RMSE** | 90.176 | 56.74 | 56.73 | 70.6 | 53.99 | **21.7** |
| **SMAPE** | 180.65 | 69.28 | 69.33 | 73.4 | 68.71 | **32.6** |
| **MASE** | 2.741 | 1.10 | 1.86 | 0.91 | 1.78 | **0.65** |

obtained after 50 simulations for each dataset group. From Table 7, it can be observed that the proposed method outperforms other methods in 8 out of the 9 time series, with Aladag [40] achieving the best performance in one time series.

### B. THE EFFICIENCY METRIC SMAPE

In this section, the prediction accuracy of different methods in different datasets is analyzed using the SMAPE metric. Table 7 provides the SMAPE results obtained after 50 simulations for each dataset group. From Table 8, it can be observed

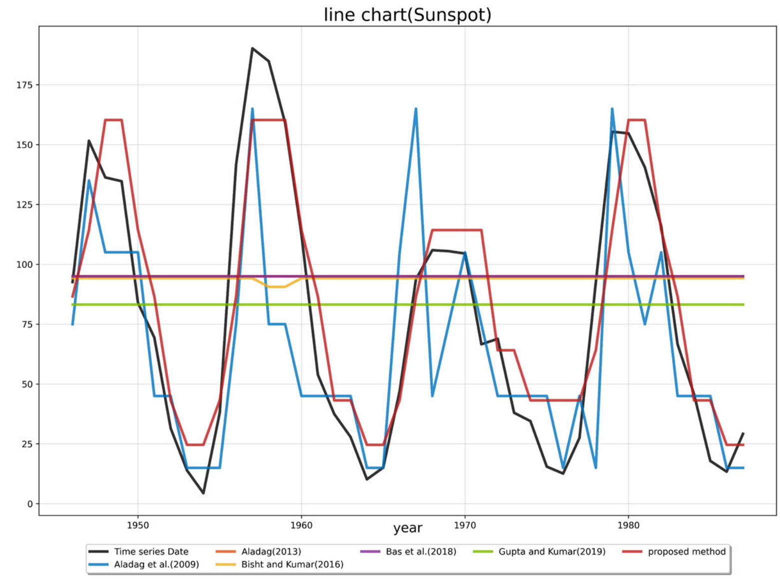



**FIGURE 2.** Predicted and actual values of different methods in the sunspot time series data.

**TABLE 4.** Dataset description.

| Data set name | Description |
|---|---|
| Gasoline | Monthly gasoline demand Ontario gallon millions Jan 1960 – Dec 1975 |
| Lynx | Number of lynx trapped annually in Mackenzie river from 1821 to 1934 |
| Passenger | International airline passengers total in thousand (Jan-49 to Dec-60) |
| Rainfall | Total annual rainfall (in inches), London, England from 1813 to 1912 |
| Sunspot | Annual wolf Sunspot number from 1700 to 1987 |
| Traffic | Monthly traffic fatalities in Ontario from 1960 to 1974 |
| TAIEX2006 | Daily closing Taiwan Capitalization Weighted Stock Index (TAIEX) data in year 2006 |
| TAIEX2007 | Daily closing Taiwan Capitalization Weighted Stock Index (TAIEX) data in year 2007 |
| TAIEX2008 | Daily closing Taiwan Capitalization Weighted Stock Index (TAIEX) data in year 2008 |

**TABLE 5.** Statistical information of time series.

| Time series | Minimum | Maximum | Mean | Standard Deviation | Kurtosis | Skewness |
|---|---|---|---|---|---|---|
| Gasoline | 86890 | 255920 | 162060 | 41662 | 2.2569 | 0.32 |
| Lynx | 39 | 6991 | 1538 | 1585.8 | 4.46 | 1.34 |
| Passenger | 104 | 622 | 280.3 | 119.97 | 2.60 | 0.57 |
| Rainfall | 16.93 | 38.1 | 24.82 | 4.21 | 3.30 | 0.67 |
| Sunspot | 0 | 190.2 | 48.43 | 39.42 | 3.65 | 1.03 |
| Traffic | 55 | 256 | 132.23 | 39.01 | 2.69 | 0.28 |
| TAIEX2006 | 6257.8 | 7823.7 | 6841.1 | 374.39 | 2.64 | 0.76 |
| TAIEX2007 | 7344.6 | 9809.9 | 8509.9 | 648.69 | 1.86 | 0.35 |
| TAIEX2008 | 4089.9 | 9295.2 | 7020.6 | 1609.1 | 1.81 | -0.46 |

that the proposed method outperforms other methods in all 9 time series.

### C. THE EFFICIENCY METRIC MASE

In this section, the prediction accuracy of different methods in different datasets is analyzed using the MASE metric. Table 8 provides the MASE results obtained after 50 simulations for each dataset group. From Table 9, it can be observed that the proposed method outperforms other methods in all 9 time series.

### D. ALGORITHM PERFORMANCE EVALUATION

In the experiments mentioned above, we studied the performance of the six algorithms separately on nine datasets. However, it's also important to assess the performance of the proposed algorithm overall, considering the datasets as

**TABLE 6. Data split and order of the time series dataset.**

| Dataset information | training set | test set | Order of the time series data |
|---|---|---|---|
| Gasoline | 163 | 29 | 10 |
| Lynx | 97 | 17 | 6 |
| Passenger | 123 | 21 | 13 |
| Rainfall | 85 | 15 | 3 |
| Sunspot | 245 | 43 | 5 |
| Traffic | 153 | 27 | 7 |
| TAIEX2006 | 210 | 37 | 7 |
| TAIEX2007 | 206 | 37 | 4 |
| TAIEX2008 | 211 | 37 | 4 |

**TABLE 7. Mean RMSE on test set (Bold face denotes best values).**

| Time series Date | Aladag et al. [39] | Aladag [40] | Bisht and Kumar [41] | Bas et al. [31] | Gupta and Kumar [42] | proposed method |
|---|---|---|---|---|---|---|
| Gasoline | 224344.96 | 57299.00 | 49789.83 | 37202.00 | 57031.52 | **14280.86** |
| Lynx | 1784.22 | 3750.80 | 2296.16 | 3750.80 | 1977.69 | **508.31** |
| Passenger | 463.14 | 173.16 | 85.20 | 117.49 | 192.98 | **39.36** |
| Rainfall | 20.91 | 5.04 | 5.57 | 5.06 | 5.27 | **4.36** |
| Sunspot | 90.18 | 56.74 | 56.73 | 70.6 | 53.99 | **21.7** |
| Traffic | 158.00 | 41.85 | 41.83 | 76.47 | 41.97 | **19.79** |
| TAIEX2006 | 7488.45 | 408.09 | 344.88 | 410.27 | 344.88 | **112.89** |
| TAIEX2007 | 8432.23 | 381.25 | 385.7 | 382.43 | **212.85** | 252.83 |
| TAIEX2008 | 4437.63 | 2376.80 | 373.45 | 2376.80 | 813.41 | **254.02** |

**TABLE 8. Mean SMAPE on test set (Bold face denotes best values).**

| Time series Date | Aladag et al. [39] | Aladag [40] | Bisht and Kumar [41] | Bas et al. [31] | Gupta and Kumar [42] | proposed method |
|---|---|---|---|---|---|---|
| Gasoline | 199.96 | 26.51 | 17.24 | 16.08 | 20.71 | **5.36** |
| Lynx | 198.35 | 124.50 | 102.97 | 124.50 | 98.08 | **50.58** |
| Passenger | 194.60 | 38.75 | 13.34 | 25.30 | 34.58 | **6.14** |
| Rainfall | 144.94 | 16.24 | 18.79 | 16.31 | 17.42 | **12.70** |
| Sunspot | 180.65 | 69.28 | 69.33 | 73.35 | 68.71 | **32.6** |
| Traffic | 190.52 | 22.30 | 22.42 | 37.52 | 22.32 | **11.12** |
| TAIEX2006 | 199.44 | 5.20 | 2.31 | 5.24 | 4.16 | **1.154** |
| TAIEX2007 | 199.26 | 3.53 | 1.94 | 3.62 | 1.94 | **1.43** |
| TAIEX2008 | 198.93 | 42.16 | 7.52 | 42.16 | 16.57 | **4.71** |

**TABLE 9. Mean MASE on test set (Bold face denotes best values).**

| Time series Date | Aladag et al. [39] | Aladag [40] | Bisht and Kumar [41] | Bas et al. [31] | Gupta and Kumar [42] | proposed method |
|---|---|---|---|---|---|---|
| Gasoline | 13.70 | 2.02 | 1.51 | 1.79 | 2.31 | **0.72** |
| Lynx | 2.26 | 0.40 | 3.39 | 0.41 | 2.90 | **0.37** |
| Passenger | 9.68 | 2.27 | 1.32 | 2.24 | 2.55 | **0.49** |
| Rainfall | 5.06 | 1.35 | 1.20 | 1.36 | 1.10 | **0.78** |
| Sunspot | 2.74 | 1.10 | 1.86 | 0.91 | 1.78 | **0.65** |
| Traffic | 5.55 | 1.45 | 1.24 | 1.11 | 1.23 | **0.57** |
| TAIEX2006 | 175.72 | 1.56 | 4.16 | 1.57 | 7.22 | **1.43** |
| TAIEX2007 | 73.56 | 1.47 | 1.28 | 1.48 | 1.42 | **0.58** |
| TAIEX2008 | 60.07 | 0.75 | 4.66 | 0.75 | 10.84 | **0.70** |

a whole. In this section, we utilized the Friedman and Bonferroni-Dunn tests for this analysis.

We first conducted the Friedman test on three efficiency metrics: RMSE, SMAPE, and MASE, to ensure the

**TABLE 10. Average rank table.**

| efficiency metric rankings | Aladag et al. [39] | Aladag [40] | Bisht and Kumar [41] | Bas et al. [31] | Gupta and Kumar [42] | proposed method |
|---|---|---|---|---|---|---|
| Average Rank of RMSE | 5.55 | 3.89 | 3.17 | 4.11 | 3.17 | 1.11 |
| Average Rank of SMAPE | 6 | 3.78 | 3.06 | 4.11 | 3.06 | 1 |
| Average Rank of MASE | 5.78 | 3.39 | 3.56 | 3.17 | 4.11 | 1 |

**TABLE 11. Friedman test table (RMSE).**

| Algorithm | Sample size | Median | Standard deviation | Statistic | P | the value of Cohen's f |
|---|---|---|---|---|---|---|
| Aladag et al. [39] | 9 | 1784.22 | 73900.373 | 28.227 | 0.000*** | 0.27 |
| Aladag [40] | 9 | 381.25 | 18845.073 | | | |
| Bisht and Kumar [41] | 9 | 171.48 | 16480.844 | | | |
| Gupta and Kumar [42] | 9 | 212.85 | 18869.051 | | | |
| Bas et al. [31] | 9 | 382.43 | 12171.105 | | | |
| proposed method | 9 | 112.89 | 4712.626 | | | |

Note: ***, **, * represent significance levels of 1%, 5%, and 10% respectively

**TABLE 12. Friedman test table (SMAPE).**

| Algorithm | Sample size | Median | Standard deviation | Statistic | P | the value of Cohen's f |
|---|---|---|---|---|---|---|
| Aladag et al. [39] | 9 | 198.35 | 17.894 | 33.147 | 0.000*** | 2.052 |
| Aladag [40] | 9 | 26.51 | 38.062 | | | |
| Bisht and Kumar [41] | 9 | 13.87 | 34.71 | | | |
| Gupta and Kumar [42] | 9 | 25.3 | 38.917 | | | |
| Bas et al. [31] | 9 | 20.71 | 31.765 | | | |
| proposed method | 9 | 6.14 | 16.634 | | | |

Note: ***, **, * represent significance levels of 1%, 5%, and 10% respectively

**TABLE 13. Friedman test table (MASE).**

| Algorithm | Sample size | Median | Standard deviation | Statistic | P | the value of Cohen's f |
|---|---|---|---|---|---|---|
| Aladag et al. [39] | 9 | 9.68 | 57.841 | 25.516 | 0.000*** | 0.617 |
| Aladag [40] | 9 | 1.45 | 0.579 | | | |
| Bisht and Kumar [41] | 9 | 1.48 | 1.378 | | | |
| Gupta and Kumar [42] | 9 | 1.35 | 0.561 | | | |
| Bas et al. [31] | 9 | 2.31 | 3.328 | | | |
| proposed method | 9 | 9.68 | 57.841 | | | |

Note: ***, **, * represent significance levels of 1%, 5%, and 10% respectively

significance of algorithm performance. To do this, we ranked the performance of different algorithms on different datasets based on the goodness of fit for the three efficiency metrics. This ranking is shown in Table 10.

Then, we employed the Friedman test to determine whether there is a significant difference in the performance of these six algorithms. Tables 11, 12, and 13 respectively present the results of the Friedman test for the three efficiency metrics: RMSE, SMAPE, and MASE.

As shown in Tables 11, 12, and 13, the p-values are $0.000{*}{*}{*}$ for all cases. Therefore, the statistical results are

**TABLE 14. Critical values for the two-tailed Bonferroni-Dunn test; the number of classifiers include the control classifier.**

| classifiers | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| $q_{0.05}$ | 1.960 | 2.241 | 2.394 | 2.498 | 2.576 | 2.638 | 2.690 | 2.724 | 2.773 |
| $q_{0.10}$ | 1.645 | 1.960 | 2.128 | 2.241 | 2.326 | 2.394 | 2.450 | 2.498 | 2.539 |

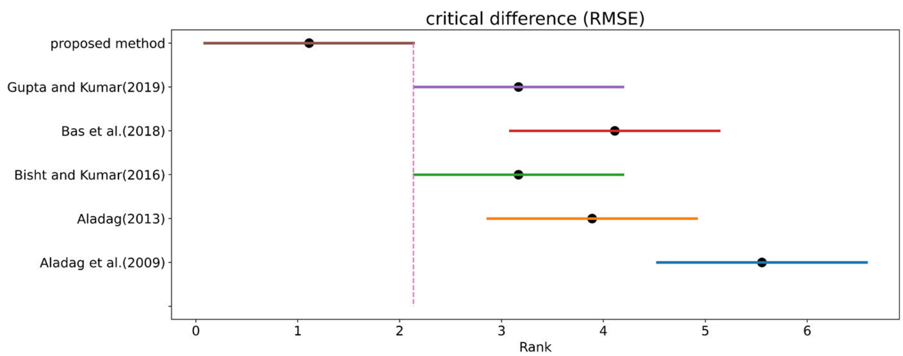



**FIGURE 3. CD diagram for the RMSE efficiency metric.**

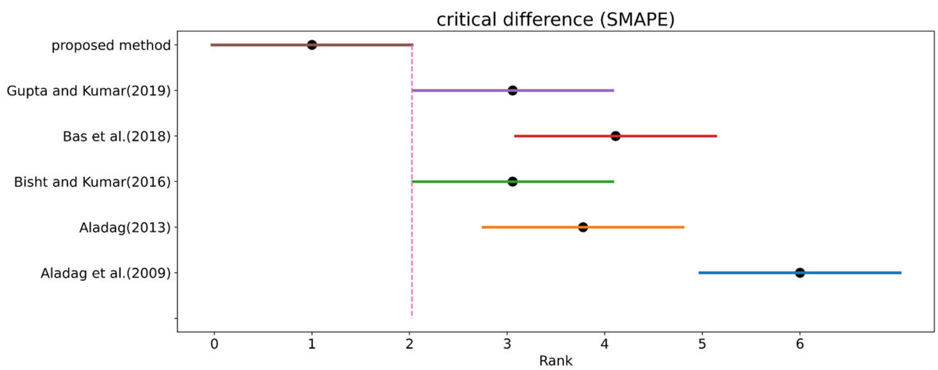



**FIGURE 4. CD diagram for the SMAPE efficiency metric.**

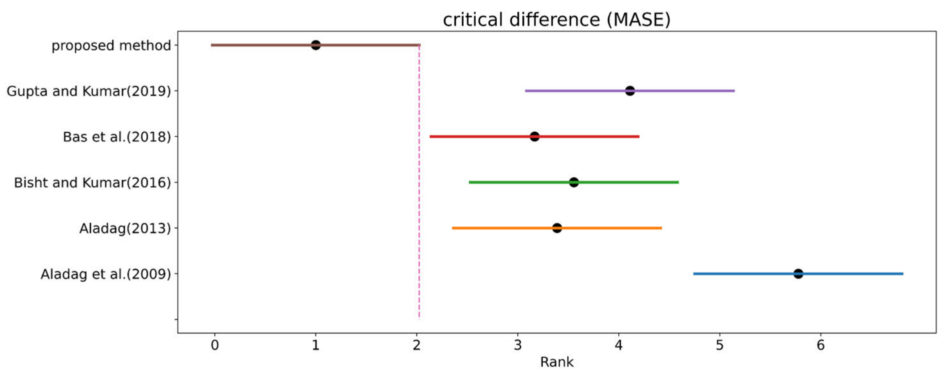



**FIGURE 5. CD diagram for the MASE efficiency metric.**

significant. There are significant differences in the performance of the 6 algorithms.

Next, we will use the Bonferroni-Dunn test with a 90% confidence level for post hoc testing. The proposed algorithm will be compared separately with the remaining 5 algorithms. The difference between the average ranks of each algorithm will be compared with the critical difference (CD) threshold. If the difference is greater than CD, it indicates that the algorithm with the higher average rank is statistically superior to the one with the lower average rank. Otherwise, there is no statistical difference between the two algorithms.

The formula for the critical difference (CD) is as follows:

$$CD = q_{\alpha}\sqrt{\frac{k(k+1)}{6N}} \tag{22}$$

$k$ is the number of algorithms, $N$ is the number of datasets.

From Table 14 and equation (22), we can obtain the CD diagrams for different efficiency metrics, as shown in Fig.3, Fig.4, and Fig.5. We find that the proposed method is significantly better than the other algorithms in terms of RMSE, SMAPE, and MASE, which ensures the statistical superiority of the proposed method over the other algorithms.

## VI. CONCLUSION

Determining fuzzy relationships has always been a key research focus in fuzzy time series forecasting. This study proposes a fuzzy time series forecasting method based on FCM clustering and a high-order DRPNN neural network (incorporated with a penalized gradient descent algorithm). FCM clustering is used to divide the domains by clustering the dataset, allowing more precise partitioning that is beneficial for improving prediction performance. The high-order DRPNN network is then used to determine fuzzy relationships. A penalized gradient descent algorithm is utilized during network training to accelerate convergence and reduce overfitting risks to some extent. Compared to other methods in the literature, this method provides superior prediction performance. Friedman and Bonferroni-Dunn tests are conducted to ensure the statistical significance of the algorithm's superiority based on different efficiency metrics.

The main contribution of this study is the usage of the DRPNN network to determine fuzzy relationships. Compared to Bas et al. [31] who used PSNN to determine fuzzy relationships, DRPNN can better simulate the intricate dynamics of complex fuzzy time series while retaining the advantages of fewer weights and lower complexity.

In future studies, deep learning methods could be used to determine fuzzy relationships, and different ANNs can be tried in various stages of the fuzzy time series framework.

In our future work, we will continuously track the latest methods and compare our approach to the new methods. And we will gradually expand the scope of application of this method, further find the limitations of this method and improve and perfect it.

## REFERENCES


[1] G. E. P. Box and G. Jenkins, *Time Series Analysis, forecasting and Control*, 1990.

[2] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2020.

[3] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*, vol. 38. Oxford, U.K.: OUP Oxford, 2012.

[4] H. H. H. Aly, "An intelligent hybrid model of neuro wavelet, time series and recurrent Kalman filter for wind speed forecasting," *Sustain. Energy Technol. Assessments*, vol. 41, Oct. 2020, Art. no. 100802.

[5] H. Al-Hadeethi, S. Abdulla, M. Diykh, R. C. Deo, and J. H. Green, "Adaptive boost LS-SVM classification approach for time-series signal classification in epileptic seizure diagnosis applications," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113676.

[6] K. Zhao, M. A. Wulder, T. Hu, R. Bright, Q. Wu, H. Qin, Y. Li, E. Toman, B. Mallick, X. Zhang, and M. Brown, "Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm," *Remote Sens. Environ.*, vol. 232, Oct. 2019, Art. no. 111181.

[7] M. S. Farahani and S. H. R. Hajiagha, "Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models," *Soft Comput.*, vol. 25, no. 13, pp. 8483–8513, Jul. 2021.

[8] M. Ali, D. M. Khan, M. Aamir, A. Ali, and Z. Ahmad, "Predicting the direction movement of financial time series using artificial neural network and support vector machine," *Complexity*, vol. 2021, pp. 1–13, Dec. 2021.

[9] X.-B. Jin, W.-T. Gong, J.-L. Kong, Y.-T. Bai, and T.-L. Su, "A variational Bayesian deep network with data self-screening layer for massive time-series data forecasting," *Entropy*, vol. 24, no. 3, p. 335, Feb. 2022.

[10] C. Virgil Negoita, "Fuzzy sets," *Fuzzy Sets Syst.*, vol. 133, no. 2, p. 275, Jan. 2003.

[11] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series—Part I," *Fuzzy Sets Syst.*, vol. 54, no. 1, pp. 1–9, Feb. 1993.

[12] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series—Part II," *Fuzzy Sets Syst.*, vol. 62, no. 1, pp. 1–8, Feb. 1994.

[13] S.-M. Chen, "Forecasting enrollments based on fuzzy time series," *Fuzzy Sets Syst.*, vol. 81, no. 3, pp. 311–319, Aug. 1996.

[14] M. H. Lee, R. Efendi, and Z. Ismail, "Modified weighted for enrollment forecasting based on fuzzy time series," *Matematika, Malaysian J. Ind. Appl. Math.*, vol. 25, pp. 67–78, Jun. 2009.

[15] K. Huarng and T. H.-K. Yu, "Ratio-based lengths of intervals to improve fuzzy time series forecasting," *IEEE Trans. Syst., Man Cybern., B, Cybern.*, vol. 36, no. 2, pp. 328–340, Apr. 2006.

[16] S.-M. Chen, X.-Y. Zou, and G. C. Gunawan, "Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques," *Inf. Sci.*, vol. 500, pp. 127–139, Oct. 2019.

[17] L. Yan, Y. Liu, and Y. Liu, "Interval feature transformation for time series classification using perceptually important points," *Appl. Sci.*, vol. 10, no. 16, p. 5428, Aug. 2020.

[18] H. Zhang, H. Li, N. Chen, S. Chen, and J. Liu, "Novel fuzzy clustering algorithm with variable multi-pixel fitting spatial information for image segmentation," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108201.

[19] T. Ren, H. Wang, H. Feng, C. Xu, G. Liu, and P. Ding, "Study on the improved fuzzy clustering algorithm and its application in brain image segmentation," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105503.

[20] Y. Jiang, K. Zhao, K. Xia, J. Xue, L. Zhou, Y. Ding, and P. Qian, "A novel distributed multitask fuzzy clustering algorithm for automatic MR brain image segmentation," *J. Med. Syst.*, vol. 43, no. 5, pp. 1–9, May 2019.

[21] E. Bas and E. Egrioglu, "A fuzzy regression functions approach based on Gustafson–Kessel clustering algorithm," *Inf. Sci.*, vol. 592, pp. 206–214, May 2022.

[22] M. S. Alam, M. M. Rahman, M. A. Hossain, M. K. Islam, K. M. Ahmed, K. T. Ahmed, B. C. Singh, and M. S. Miah, "Automatic human brain tumor detection in MRI image using template-based K means and improved fuzzy C means clustering algorithm," *Big Data Cognit. Comput.*, vol. 3, no. 2, p. 27, May 2019.

[23] Q. Feng, L. Chen, C. L. P. Chen, and L. Guo, "Deep fuzzy clustering—A representation learning approach," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1420–1433, Jul. 2020.

[24] N. Baklouti, A. Abraham, and A. M. Alimi, "A beta basis function interval type-2 fuzzy neural network for time series applications," *Eng. Appl. Artif. Intell.*, vol. 71, pp. 259–274, May 2018.

[25] O. Cagcag Yolcu and H.-K. Lam, "A combined robust fuzzy time series method for prediction of time series," *Neurocomputing*, vol. 247, pp. 87–101, Jul. 2017.

[26] S. Panigrahi and H. S. Behera, "A study on leading machine learning techniques for high order fuzzy time series forecasting," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103245.

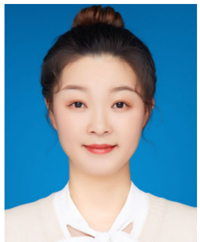
[27] R. M. Pattanayak, S. Panigrahi, and H. S. Behera, "High-order fuzzy time series forecasting by using membership values along with data and support vector machine," *Arabian J. Sci. Eng.*, vol. 45, no. 12, pp. 10311–10325, Dec. 2020.

[28] R. M. Pattanayak, H. S. Behera, and S. Panigrahi, "A novel probabilistic intuitionistic fuzzy set based model for high order fuzzy time series forecasting," *Eng. Appl. Artif. Intell.*, vol. 99, Mar. 2021, Art. no. 104136.

[29] R. M. Pattanayak, H. S. Behera, and S. Panigrahi, "A novel high order hesitant fuzzy time series forecasting by using mean aggregated membership value with support vector machine," *Inf. Sci.*, vol. 626, pp. 494–523, May 2023.
[30] Y. Shin and J. Ghosh, "The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation," in *Proc. Seattle Int. Joint Conf. Neural Netw.*, Jul. 1991, pp. 13–18.
[31] E. Bas, C. Grosan, E. Egrioglu, and U. Yolcu, "High order fuzzy time series method based on pi-sigma neural network," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 350–356, Jun. 2018.
[32] R. Ghazali, A. J. Hussain, and P. Liatsis, "Dynamic ridge polynomial neural network: Forecasting the univariate non-stationary and stationary trading signals," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3765–3776, Apr. 2011.
[33] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy C-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, Jan. 1984.
[34] Y. Shin and J. Ghosh, "Ridge polynomial networks," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 610–622, May 1995.
[35] D. Al-Jumeily, R. Ghazali, and A. Hussain, "Predicting physical time series using dynamic ridge polynomial neural networks," *PLoS ONE*, vol. 9, no. 8, Aug. 2014, Art. no. e105766.
[36] R. Ghazali, A. J. Hussain, D. Al-Jumeily, and P. Lisboa, "Time series prediction using dynamic ridge polynomial neural networks," in *Proc. 2nd Int. Conf. Develop. eSyst. Eng.*, Dec. 2009, pp. 354–363.
[37] R. Hyndman and Y. Yang. (2018). *TSDL: Time Series Data Library*. [Online]. Available: https://pkg.yangzhuoranyang.com/tsdl/articles/tsdl.html
[38] *TAIEX Total Index Historical Data*. Accessed: 2023. [Online]. Available: https://www.twse.com.tw/en/page/trading/indices/MI_5MINS_HIST.html
[39] C. H. Aladag, M. A. Basaran, E. Egrioglu, U. Yolcu, and V. R. Uslu, "Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4228–4231, Apr. 2009.
[40] C. H. Aladag, "Using multiplicative neuron model to establish fuzzy logic relationships," *Expert Syst. Appl.*, vol. 40, no. 3, pp. 850–853, Feb. 2013.
[41] K. Bisht and S. Kumar, "Fuzzy time series forecasting method based on hesitant fuzzy sets," *Expert Syst. Appl.*, vol. 64, pp. 557–568, Dec. 2016.
[42] K. K. Gupta and S. Kumar, "Hesitant probabilistic fuzzy set based time series forecasting method," *Granular Comput.*, vol. 4, no. 4, pp. 739–758, Oct. 2019.

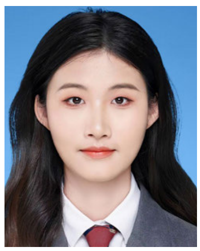
**BOYANG WANG** is currently pursuing the B.Sc. degree with Kunming University. His research interests include neural networks, machine learning, and data analysis.

**XISONG MIAO** received the master's degree from Third Military Medical University. He is currently an Associate Professor with Kunming University. His research interests include public health and digital healthcare.

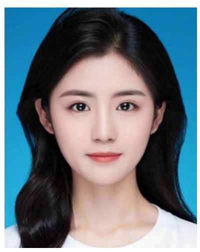


**HUANYU WEI** received the Ph.D. degree in plant pathology from Yunnan Agricultural University. She was a recipient of the Yunnan Provincial Science and Technology Progress First Prize, in 2022.

**MD. GOLAM SAKLAIN** received the B.S. degree from Kunming University, where he is currently pursuing the M.Sc. degree. His research interests include neural networks, machine learning, and data analysis.



**YINAN ZHI** is currently pursuing the B.Sc. degree with Kunming University. Her research interests include neural networks, machine learning, and data analysis.



**HONGYAN JIN** is currently pursuing the B.Sc. degree with Kunming University. Her research interests include neural networks, machine learning, and data analysis.

**JIAXIE LI** is currently pursuing the B.Sc. degree with Kunming University. Her research interests include neural networks, machine learning, and data analysis.