



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A novel hybrid deep fuzzy model based on gradient descent algorithm with application to time series forecasting

Hui Zhang^a, Bo Sun^a, Wei Peng^{b,c,*}

^a School of Control Science and Engineering, Shandong University, Jinan, 250014, China

^b The Key Laboratory of Intelligent Buildings Technology of Shandong Province, Shandong Jianzhu University, Jinan, 250101, China

^c School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan, 250101, China

ARTICLE INFO

Keywords:

Deep fuzzy system
Time series prediction
Gradient descent
Interpretability
Regularization

ABSTRACT

Deep fuzzy systems are widely used in time series forecasting tasks due to their excellent nonlinear transformation capabilities and interpretability. However, traditional optimization methods and fixed network structures cannot make the deep fuzzy system obtain the expected prediction accuracy and generalization ability. Therefore, a novel hybrid deep fuzzy model (HDFM) is proposed in this paper. Firstly, two types fuzzy modules, namely the type-1 TSK fuzzy module (T1TSKFM) and the interval type-2 TSK fuzzy module (IT2TSKFM), are respectively presented and designed in detail. And, the gradient expressions of the fuzzy parameters, including the antecedent and consequent parameters, are also derived in detail. Secondly, in order to optimize the fuzzy parameters and to further accelerate the module convergence, a novel parameter optimization strategy is presented, combining the gradient descent with the Regularization, the DropRule and the AdaBound algorithms. Thirdly, a new stacked hybrid deep fuzzy architecture is proposed, which can be automatically trained and constructed using the designed T1TSKFM and the IT2TSKFM. Then, the detailed data-driven learning and updating strategy are given in step by step way. In addition, both the layered structure interpretability and the fuzzy rule interpretability are respectively analyzed. This can guarantee that the proposed model not only has the better forecasting accuracy, but also has the higher interpretability. Finally, in order to verify the effectiveness of the proposed method, several comparative experiments are given. Experimental results show that the forecasting performance of the proposed model outperforms the other comparisons, such as the DIRM-DFM, the IT2DIRM-DFM, the DCFS, and the ANFIS method. At the same time, the proposed model has better interpretability and more flexible construction property.

1. Introduction

Time series forecasting problem is widely used in daily life, such as the stock prediction (Singh & Srivastava, 2017), the power load prediction (Li, Su, & Shu, 2014), the photovoltaic power generation prediction (Cervone, Clemente-Harding, Alessandrini, & Delle Monache, 2017; Chu et al., 2015; Hossain, Mekhilef, Danesh, Olatomiwa, & Shamshirband, 2017; Li, Wen, Tseng, & Wang, 2019; Zhou, Zhou, Gong, & Jiang, 2020) and so on. Accurate prediction will directly help decision makers and managers to timely adjust policies or make proper planning.

Time series forecasting task is a typical nonlinear regression problem. As a simple and effective way, machine learning methods have been widely and successfully applied in solving time series prediction problems (Cheng, Wan, Choo, et al., 2018; Karevan & Suykens, 2020; Liu, Gong, Yang and Chen, 2020; Liu, Liu and Wu, 2020),

benefitting from the better generalization performance and prediction performance. In Liu, Liu et al. (2020), a new convolutional neural network FCB-CNN was proposed and verified on 12 benchmark datasets, which effectively improved the stability and prediction accuracy of deep learning in time series prediction. Liu, Gong et al. (2020) introduced the attention mechanism into the recurrent neural network(RNN), and successfully applied in four prediction scenarios, i.e., energy, finance, environment and medical fields. Karevan and Suykens (2020) proposed a T-LSTM model based on transductive learning and data-driven strategy and applied it to the accurate prediction of weather conditions. Cheng et al. (2018) proposed a method of unsupervised pre-training through deep confidence networks, and then supervised curve fitting by RNN to successfully predict the change trend of time series in IoT applications. Although the above machine learning methods show well prediction ability and generalization ability,

* Corresponding author at: The Key Laboratory of Intelligent Buildings Technology of Shandong Province, Shandong Jianzhu University, Jinan, 250101, China.
E-mail addresses: 202234949@mail.sdu.edu.cn (H. Zhang), sunbo@sdu.edu.cn (B. Sun), pengwei19@sdjzu.edu.cn (W. Peng).

<https://doi.org/10.1016/j.eswa.2023.121988>

Received 14 June 2023; Received in revised form 11 September 2023; Accepted 2 October 2023

Available online 12 October 2023

0957-4174/© 2023 Elsevier Ltd. All rights reserved.

they still face with some problems, such as difficulty in tuning hyper parameters, rely on the large training set and weak interpretability.

Benefitting from the higher interpretability and stronger approximation ability, TSK fuzzy logic system (Coupland & John, 2008; Nguyen et al., 2019) has gradually become an efficient method to solve the modeling and uncertainty characterization of complex systems. Therefore, the TSK fuzzy logic system has been successfully applied to time series prediction scenarios (Bilgili, Ilhan, & Ünal, 2022; Tak, Evren, Tez, & Egrioglu, 2018; Zhang & Peng, 2021), and exhibited outstanding predictive performance. In Tak et al. (2018), an improved cyclic type-1 fuzzy method was proposed by considering the disturbance term on the basis of the method of moving average (MA) model. Zhang and Peng (2021) proposed an interval type-2 fuzzy logic system (IT2FLS) based on particle swarm optimization (PSO) algorithm and used it to accurately predict time series data in application fields with strong randomness, such as building energy consumption. A time series prediction method based on ANFIS and LSTM was proposed and successfully applied it to the hourly prediction of air pressure time series in Bilgili et al. (2022). In addition, some recent forecasting and optimization works on fuzzy time series have also gained widespread attention (Singh, 2020, 2021; Singh & Huang, 2019). Singh (2021) proposed a fuzzy quantum time series forecasting model (FQTSFM) and focused on the impact of the selection of the universe of discourse and the determination of the fuzzy degree of the memberships on time series forecasting performance. In Singh (2020), the fuzzy membership degree was used to describe the neutrosophic set (NS), and a solution based on the PSO algorithm was given for the optimal selection of the domain of time series datasets, which obtained better results than the benchmark model. In Singh and Huang (2019), a neutrosophic entropy decision rule based on the IF-THEN structure is proposed, and the ANN-based architecture takes NEDR as input to evolve the prediction results, improve the performance of the artificial neural network and obtain the best prediction results. However, these models based on fuzzy systems usually have the shallow hierarchy structure, and still have a risk of the rule explosion, especially when facing the increase of input dimension or handling the big data.

Inspired by the idea of neural network, some modular fuzzy models have been proposed recently, especially the fuzzy models based on single input rule module (SIRM-FM) (Yi, Yubazaki, & Hirota, 2001) and its improved versions, FSIRM-FM (Seki, Ishii, & Mizumoto, 2008), T2SIRM-FM (Li & Yi, 2010), FWSIRM-FM (Li, Gao, Yi, & Zhang, 2016), DIRM-DFM (Li, Zhou, Peng, Lv, & Luo, 2020), IT2DIRM-DFM (Peng, Zhou, Li, Deng, & Zhang, 2021). These modular fuzzy models behave good interpretability and nonlinear approximation ability, and can effectively describe the uncertainties. However, in terms of structure, the above methods generally are composed of fuzzy modules on a single type, and lacks of flexibility. On the other hand, the above deep fuzzy models generally use the least squares method or its improved algorithm for the parameter optimization. Because the parameter accuracy optimized largely effects the prediction performance and generalization ability of the models, it is critical to find the more efficiently method to optimize the parameters.

In order to solve the parameters optimization problem, improve the generalization ability and guarantee the interpretability, a novel hybrid deep fuzzy system (HDFM) is proposed in this study. Different from the deep fuzzy models in Li et al. (2016), Li and Yi (2010), Li et al. (2020), Peng et al. (2021), Seki et al. (2008) and Yi et al. (2001), the proposed HDFM fuses two types of fuzzy modules to improve the construction flexibility and obtain the tradeoff of computation time and prediction accuracy. On the other hand, the proposed HDFM no longer restricts the input dimension of the fuzzy module. Therefore, the width of the HDFM will be reduced under the same input dimension of the model, and computation time will also be further shortened. And then, a parameter optimization strategy combining the gradient descent with the Regularization, the DropRule and the AdaBound algorithms is presented and used to optimized each fuzzy modules of HDFM. Compared with the

optimization method based on the least squares algorithm used in Li et al. (2016), Li and Yi (2010), Li et al. (2020), Peng et al. (2021), Seki et al. (2008) and Yi et al. (2001), the proposed method(parameter optimization strategy) effectively suppresses the overfitting problem and achieves better generalization performance.

Here, the main contributions and innovations of this study are summarized as follows:

- Two types fuzzy modules, i.e., type-1 TSK fuzzy module (T1TSKFM) and the interval type-2 TSK fuzzy module (IT2TSKFM), are presented and designed in respectively. Then, the gradients of parameters, including the antecedents and the consequents, of the IT2TSKFM are given in detail. In addition, the gradient-based optimization algorithm is presented, combining with the Regularization, the DropRule and the AdaBound methods, and then is applied to the IT2TSKFM to optimize the fuzzy parameters and to further accelerate the module convergence.
- A new stacked hybrid deep fuzzy architecture is proposed, which can be automatically trained and constructed using the designed T1TSKFM and the IT2TSKFM. Then, the detailed data-driven learning strategy is given in step by step way. In addition, both the layered structure interpretability and the fuzzy rule interpretability are respectively analyzed.
- In order to verify the effectiveness of the proposed method, several comparative experiments are given. Experimental results show that the forecasting performance of the proposed model outperforms the other comparisons, such as the DIRM-DFM, the IT2DIRM-DFM, the DCFs, and the ANFIS method, by keeping better interpretability and more flexible construction property.

The remaining content is organized as follows. In Section 2, the theoretical background is reviewed, mainly including fuzzy sets and fuzzy systems. In Section 3, two types of TSK fuzzy modules are designed. Then, the training strategy and the gradients of the parameters for IT2TSKFM are given in respectively. In Section 4, the novel HDFM is proposed, which can be automatically trained and constructed using the designed T1TSKFM and the IT2TSKFM. Furthermore, the automatically structure updating strategy is given. Then, the detailed data-driven learning strategy of the HDFM is given in step by step way. Both the layered interpretability and rule interpretability are also analyzed. In Section 5, comparative experiments are designed on three public datasets, and the advantages and disadvantages of each comparison model are analyzed by combining various performance indicators. Finally, the conclusion is given in Section 6.

2. Background review

In this section, two types of fuzzy sets and the corresponding fuzzy logic systems will be briefly reviewed.

2.1. Fuzzy sets

Generally, a fuzzy set \tilde{A} (Zadeh, 1965) is defined as a set function, and the domain X can be mapped between $[0, 1]$. For the fuzzy set \tilde{A} , a membership function (MF) $\mu_{\tilde{A}}(x)$ is used to describe mapping relationship on the $X \rightarrow [0, 1]$.

For the continuous domain X , then the fuzzy set \tilde{A} can be expressed as

$$\tilde{A} = \int_{x \in X} \frac{\mu_{\tilde{A}}(x)}{x}, \quad (2.1)$$

where, $0 \leq \mu_{\tilde{A}}(x) \leq 1$.

If the domain X is discrete and expressed as X_d , then the fuzzy set \tilde{A} can be expressed as

$$\tilde{A} = \sum_{x \in X_d} \frac{\mu_{\tilde{A}}(x)}{x}. \quad (2.2)$$

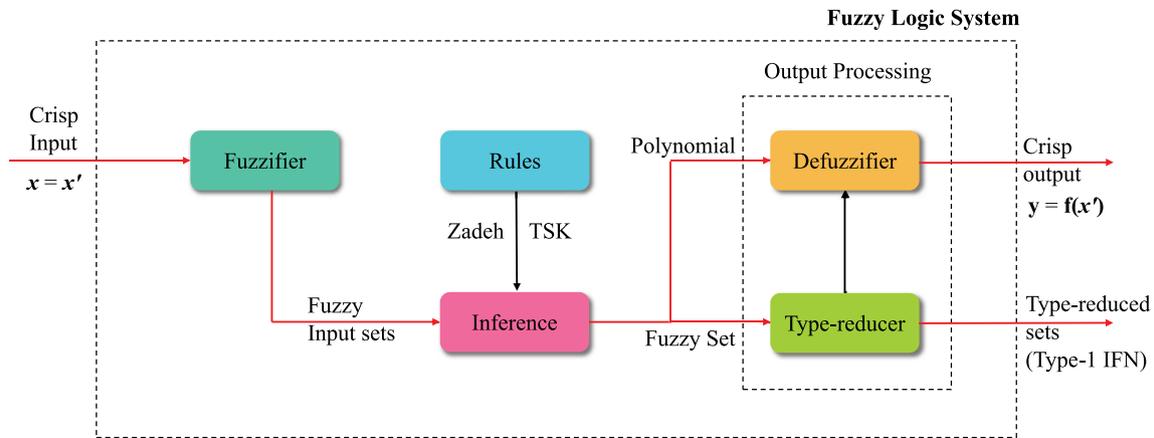


Fig. 1. Schematic diagram of the T2FLS.

A type-2 fuzzy set(T2FS) \tilde{A} (Zadeh, 1975), often called a general type-2 fuzzy set(GT2FS), is a set of three dimensional spaces, which is usually described by

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid x \in X, u \in U \equiv [0, 1]\}, \quad (2.3)$$

where, $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. X is of the principal variable x of the fuzzy set \tilde{A} . U is the domain of the secondary variable u of the fuzzy set \tilde{A} , and is often assumed to be $[0, 1]$. So the fuzzy set \tilde{A} for GT2FS can be denoted as

$$\tilde{A} = \int_{x \in X} \int_{u \in [0,1]} \frac{\mu_{\tilde{A}}(x, u)}{x, u}. \quad (2.4)$$

However, for $\forall x \in X$ and $u \in U \in [0, 1]$, if $\mu_{\tilde{A}}(x, u) = 1$, then GT2FS will be simplified as an interval type-2 fuzzy set (IT2FS) (Mendel, 2017), as

$$\tilde{A} = \int_{x \in X} \int_{u \in [0,1]} 1/(x, u). \quad (2.5)$$

2.2. Fuzzy logic systems

Fuzzy logic systems are developed on the basis of fuzzy sets, and have been widely used in modeling and control. Simply speaking, by converting the type-1 fuzzy set (T1FS) into T2FS, the type-1 fuzzy logic system (T1FLS) can be transformed into type-2 fuzzy logic system (T2FLS). In fact, there is still some great difference between the T1FLS and the T2FLS. Fig. 1 shows the block diagram of the T2FLS.

(1) Fuzzifier: it realizes that the accurate input vector $\mathbf{x} = (x_1, x_2, \dots, x_M)^T \in X_1 \times X_2 \times \dots \times X_M \equiv X$ maps to the fuzzy set \tilde{A} on the domain U . The single-valued fuzzifier is an effective way.

(2) Rule Base: the TSK fuzzy rule is a common expressions of the consequents.

$$\text{IF } x_1 \text{ is } \tilde{X}_{r,1} \text{ and...and } x_M \text{ is } \tilde{X}_{r,M} \text{ THEN } y_r \text{ is } g(x_1, x_2, \dots, x_M), \quad (2.6)$$

where, $r = 1, 2, \dots, R$, denotes the r th rule. $\tilde{X}_{r,1}, \dots, \tilde{X}_{r,M}$ are the antecedents of the rule. For the consequent, $g(x_1, x_2, \dots, x_M)$ is the linear polynomial for the TSK rule.

(3) Fuzzy Inference: it is mainly used to calculate the fired degree from multiple antecedents for each of fuzzy rules. The product t -norm is a common method.

(4) Output Processing: for the TSK fuzzy system, some effective direct defuzzification methods, such as Biglarbegian–Melek–Mendel (BMM) (Begian, Melek, & Mendel, 2008), can be used to directly obtain the crisp output.

3. The design of two types of TSK fuzzy modules and optimization

In this section, two types of TSK fuzzy modules, i.e., the T1TSKFM and the IT2TSKFM are given in detailed. Then, a novel strategy for optimizing the TSK fuzzy modules is proposed, in order to optimize the parameters and to accelerate the modules convergence.

3.1. The design of two types of TSK fuzzy modules

A. Type-1 TSK Fuzzy Module

Assume that the input vector $\mathbf{x} = (x_1, \dots, x_N)^T \in \mathbb{R}^{N \times M}$, $\mathbf{x}_N = (x_{s1}, \dots, x_{sM})$, $s = 1, \dots, N$. M denotes the dimension of input samples, and N denotes the number of samples. Here, the Gaussian MF (G-MF) is chosen to describe the fuzzy set, and its membership degree is calculated as follows

$$\mu_{\tilde{X}_{r,j}} = \exp \left[-\frac{(x_{sj} - m_{r,j})^2}{2\sigma_{r,j}^2} \right]. \quad (3.1)$$

For the T1TSKFM, the r th rule can be expressed as follows

$$\text{IF } x_{s1} \text{ is } \tilde{X}_{r,1} \text{ and...and } x_{sM} \text{ is } \tilde{X}_{r,M} \text{ THEN } y_r = \omega_{r,0} + \sum_{j=1}^M \omega_{r,j} x_{sj}, \quad (3.2)$$

where $r = 1, \dots, R$, $\omega_{r,0}$ and $\omega_{r,j}$ are the consequent parameters of the rule.

The fired degree of the r th fuzzy rule can be obtained by the product t -norm

$$F_r(\mathbf{x}) = \prod_{j=1}^M \mu_{\tilde{X}_{r,j}}(x_{sj}), \quad (3.3)$$

where $\mu_{\tilde{X}_{r,j}}$ is the membership degree of input x_{sj} in the fuzzy set $\tilde{X}_{r,j}$. The final fuzzy output can be expressed as

$$y = \frac{\sum_{r=1}^R F_r(\mathbf{x}) \cdot y_r}{\sum_{r=1}^R F_r(\mathbf{x})} = \frac{\sum_{r=1}^R \prod_{j=1}^M \mu_{\tilde{X}_{r,j}}(x_{sj}) (\omega_{r,0} + \sum_{j=1}^M \omega_{r,j} x_{sj})}{\sum_{r=1}^R \prod_{j=1}^M \mu_{\tilde{X}_{r,j}}(x_{sj})}. \quad (3.4)$$

B. Interval Type-2 TSK Fuzzy Module

The G-MFs are chosen to characterize the IT2FSs and can be expressed as

$$\bar{\mu}_{\tilde{X}_{r,j}} = \exp \left[-\frac{(x_{sj} - m_{r,j})^2}{2\bar{\sigma}_{r,j}^2} \right], \quad (3.5)$$

$$\underline{\mu}_{\tilde{X}_{r,j}} = \exp \left[-\frac{(x_{sj} - m_{r,j})^2}{2\underline{\sigma}_{r,j}^2} \right]. \quad (3.6)$$

For the IT2TSKFM, the r th rule is given as follows

$$\text{IF } \mathbf{x}_{s1} \text{ is } \widetilde{X}_{r,1} \text{ and...and } \mathbf{x}_{sM} \text{ is } \widetilde{X}_{r,M} \text{ THEN } y_r = \left[\underline{y}_r(\mathbf{x}), \overline{y}_r(\mathbf{x}) \right], \quad (3.7)$$

where, $r = 1, \dots, R$, $\underline{\omega}_{r,0}$, $\overline{\omega}_{r,j}$, $\underline{\omega}_{r,0}$, and $\overline{\omega}_{r,j}$ are the consequent parameters of rule, and $\underline{y}_r(\mathbf{x}) = \underline{\omega}_{r,0} + \sum_{j=1}^M \underline{\omega}_{r,j} \mathbf{x}_{sj}$, $\overline{y}_r(\mathbf{x}) = \overline{\omega}_{r,0} + \sum_{j=1}^M \overline{\omega}_{r,j} \mathbf{x}_{sj}$.

The fired degree interval of the r th fuzzy rule can be obtained as

$$F_r(\mathbf{x}) = \left[\underline{f}_r(\mathbf{x}), \overline{f}_r(\mathbf{x}) \right], \quad (3.8)$$

where $\overline{f}_r(\mathbf{x}) = \prod_{j=1}^M \overline{\mu}_{\widetilde{X}_{r,j}}(\mathbf{x}_{sj})$, and $\underline{f}_r(\mathbf{x}) = \prod_{j=1}^M \underline{\mu}_{\widetilde{X}_{r,j}}(\mathbf{x}_{sj})$.

The final fuzzy output calculated by BMM can be expressed as

$$\hat{y}(\mathbf{x}) = \alpha \frac{\sum_{r=1}^R \underline{f}_r(\mathbf{x}) \underline{y}_r(\mathbf{x})}{\sum_{r=1}^R \underline{f}_r(\mathbf{x})} + \beta \frac{\sum_{r=1}^R \overline{f}_r(\mathbf{x}) \overline{y}_r(\mathbf{x})}{\sum_{r=1}^R \overline{f}_r(\mathbf{x})}, \quad (3.9)$$

where, α and β are coefficients of BMM, and $\alpha + \beta = 1$, $\alpha, \beta \geq 0$.

3.2. Optimization algorithm for IT2TSKFM

Inspired by the ideas of literature (Wu, Yuan, Huang, & Tan, 2019), a new regularization item and the detailed gradient formulations for the IT2TSKFM are presented. Then, a novel parameter optimization strategy combining the gradient descent with the Regularization, the DropRule and the AdaBound algorithms is given and applied to optimize the parameters of IT2TSKFM and to further accelerate the module convergence.

A. Regularization and Gradient Formulations

In this work, a new loss function for the IT2TSKFM is proposed as follows by introducing the l_2 regularization, in order to improve the parameter tuning

$$L = \frac{1}{2} \sum_{n=1}^N [y_n - \hat{y}(\mathbf{x}_n)]^2 + \frac{\lambda}{2} \sum_{r=1}^R (\underline{\omega}_r^2 + \overline{\omega}_r^2). \quad (3.10)$$

Substituting Eq. (3.9) into (3.10), the loss function can be further expressed as

$$L = \frac{1}{2} \sum_{n=1}^N \left[y_n - \left(\alpha \frac{\sum_{r=1}^R \underline{f}_r(\mathbf{x}_n) \underline{y}_r(\mathbf{x}_n)}{\sum_{r=1}^R \underline{f}_r(\mathbf{x}_n)} + \beta \frac{\sum_{r=1}^R \overline{f}_r(\mathbf{x}_n) \overline{y}_r(\mathbf{x}_n)}{\sum_{r=1}^R \overline{f}_r(\mathbf{x}_n)} \right) \right]^2 + \frac{\lambda}{2} \sum_{r=1}^R (\underline{\omega}_r^2 + \overline{\omega}_r^2), \quad (3.11)$$

where λ is the coefficient of the l_2 regularization, and $\lambda \geq 0$.

According to the loss function (3.11), the gradient derivation of the parameters for the IT2TSKFM can be obtained as follows

$$\begin{aligned} \frac{\partial L}{\partial c_{r,m}} &= \frac{1}{2} \sum_{n=1}^{N_{bs}} \sum_{k=1}^R \frac{\partial L}{\partial \hat{y}(\mathbf{x}_n)} \left[\frac{\partial \hat{y}(\mathbf{x}_n)}{\partial \underline{f}_k(\mathbf{x}_n)} \frac{\partial \underline{f}_k(\mathbf{x}_n)}{\partial \underline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})} \frac{\partial \underline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})}{\partial c_{r,m}} \right. \\ &\quad \left. + \frac{\partial \hat{y}(\mathbf{x}_n)}{\partial \overline{f}_k(\mathbf{x}_n)} \frac{\partial \overline{f}_k(\mathbf{x}_n)}{\partial \overline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})} \frac{\partial \overline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})}{\partial c_{r,m}} \right] \\ &= \sum_{n=1}^{N_{bs}} \sum_{k \in \Phi(r,m)} \left[\alpha \frac{\underline{y}_k(\mathbf{x}_n) \sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) - \sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) \underline{y}_i(\mathbf{x}_n)}{\left[\sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) \right]^2} (\hat{y}(\mathbf{x}_n) - y_n) \underline{f}_k(\mathbf{x}_n) \frac{(x_{n,m} - c_{r,m})}{\sigma_{r,m}^2} \right. \\ &\quad \left. + \beta \frac{\overline{y}_k(\mathbf{x}_n) \sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) - \sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) \overline{y}_i(\mathbf{x}_n)}{\left[\sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) \right]^2} (\hat{y}(\mathbf{x}_n) - y_n) \overline{f}_k(\mathbf{x}_n) \frac{(x_{n,m} - c_{r,m})}{\overline{\sigma}_{r,m}^2} \right], \end{aligned} \quad (3.12)$$

$$\frac{\partial L}{\partial \sigma_{r,m}} = \frac{1}{2} \sum_{n=1}^{N_{bs}} \sum_{k=1}^R \frac{\partial L}{\partial \hat{y}(\mathbf{x}_n)} \frac{\partial \hat{y}(\mathbf{x}_n)}{\partial \underline{f}_k(\mathbf{x}_n)} \frac{\partial \underline{f}_k(\mathbf{x}_n)}{\partial \underline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})} \frac{\partial \underline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})}{\partial \sigma_{r,m}}$$

$$= \sum_{n=1}^{N_{bs}} \sum_{k \in \Phi(r,m)} \left[\alpha \frac{\left[\underline{y}_k(\mathbf{x}_n) \sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) - \sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) \underline{y}_i(\mathbf{x}_n) \right]}{\left[\sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) \right]^2} (\hat{y}(\mathbf{x}_n) - y_n) \underline{f}_k(\mathbf{x}_n) \frac{(x_{n,m} - c_{r,m})^2}{\sigma_{r,m}^3} \right], \quad (3.13)$$

$$\begin{aligned} \frac{\partial L}{\partial \overline{\sigma}_{r,m}} &= \frac{1}{2} \sum_{n=1}^{N_{bs}} \sum_{k=1}^R \frac{\partial L}{\partial \hat{y}(\mathbf{x}_n)} \frac{\partial \hat{y}(\mathbf{x}_n)}{\partial \overline{f}_k(\mathbf{x}_n)} \frac{\partial \overline{f}_k(\mathbf{x}_n)}{\partial \overline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})} \frac{\partial \overline{\mu}_{\widetilde{X}_{k,m}}(x_{n,m})}{\partial \overline{\sigma}_{r,m}} \\ &= \sum_{n=1}^{N_{bs}} \sum_{k \in \Phi(r,m)} \left[\beta \frac{\left[\overline{y}_k(\mathbf{x}_n) \sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) - \sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) \overline{y}_i(\mathbf{x}_n) \right]}{\left[\sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) \right]^2} (\hat{y}(\mathbf{x}_n) - y_n) \overline{f}_k(\mathbf{x}_n) \frac{(x_{n,m} - c_{r,m})^2}{\overline{\sigma}_{r,m}^3} \right], \end{aligned} \quad (3.14)$$

$$\begin{aligned} \frac{\partial L}{\partial \underline{\omega}_r} &= \frac{1}{2} \sum_{n=1}^{N_{bs}} \frac{\partial L}{\partial \hat{y}(\mathbf{x}_n)} \frac{\partial \hat{y}(\mathbf{x}_n)}{\partial \underline{y}_r(\mathbf{x}_n)} \frac{\partial \underline{y}_r(\mathbf{x}_n)}{\partial \underline{\omega}_r} + \frac{\lambda}{2} \frac{\partial L}{\partial \underline{\omega}_r} \\ &= \sum_{n=1}^{N_{bs}} \left[(\hat{y}(\mathbf{x}_n) - y_n) \left[\alpha \frac{\underline{f}_k(\mathbf{x}_n)}{\left[\sum_{i=1}^R \underline{f}_i(\mathbf{x}_n) \right]^2} \right] \underline{\omega}_r \right] + \lambda \underline{\omega}_r \end{aligned} \quad (3.15)$$

$$\begin{aligned} \frac{\partial L}{\partial \overline{\omega}_r} &= \frac{1}{2} \sum_{n=1}^{N_{bs}} \frac{\partial L}{\partial \hat{y}(\mathbf{x}_n)} \frac{\partial \hat{y}(\mathbf{x}_n)}{\partial \overline{y}_r(\mathbf{x}_n)} \frac{\partial \overline{y}_r(\mathbf{x}_n)}{\partial \overline{\omega}_r} + \frac{\lambda}{2} \frac{\partial L}{\partial \overline{\omega}_r} \\ &= \sum_{n=1}^{N_{bs}} \left[\hat{y}(\mathbf{x}_n) - y_n \left[\beta \frac{\overline{f}_k(\mathbf{x}_n)}{\left[\sum_{i=1}^R \overline{f}_i(\mathbf{x}_n) \right]^2} \right] \overline{\omega}_r \right] + \lambda \overline{\omega}_r, \end{aligned} \quad (3.16)$$

where $\Phi(r, m)$ is the rules which contains the $\widetilde{X}_{r,m}$, $x_{n,0} \equiv 1$, and N_{bs} is the batch size in the gradient descent. A mini-batch gradient descent algorithm is presented to optimize the antecedent and consequent parameters of the IT2TSKFM in this work. In each epoch, this algorithm randomly selects N_{bs} training samples and calculates the gradient. Then, the parameters of the IT2TSKFM will be updated. Make θ_k is the parameter vector of the k th iterates, and $\partial L / \partial \theta_k$ is the first-order gradient. Then, the update rule is

$$\theta_k = \theta_{k-1} - \alpha \frac{\partial L}{\partial \theta_k}, \quad (3.17)$$

where, α is learning rate.

B. IT2TSKFM's Training Strategy

Referring to the ideas of Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014) and Wan, Zeiler, Zhang, Le Cun, andergus (2013), this paper adopts the DropRule algorithm to accelerate the module training and to improve the generalization ability of the IT2TSKFM. On the other hand, the AdaBound (Luo, Xiong, Liu, & Sun, 2019) algorithm is used to effectively deal with the sparse gradient and non-stationary conditions, then applied to further accelerate the convergence of the IT2TSKFM.

Thanks to the proposed loss function and the corresponded gradient formulations of the IT2TSKFM, the integrated module training strategy which combines the gradient descent and AdaBound algorithm is firstly introduced to the IT2TSKFM and used to optimize the parameters. The proposed integrated module training strategy have more effective convergence and more excellent generalization ability. The adopted bound functions are respectively given as follows

$$l(k) = 0.01 - \frac{0.01}{(1 - \beta_2) k + 1}, \quad (3.18)$$

$$u(k) = 0.01 - \frac{0.01}{(1 - \beta_2) k}, \quad (3.19)$$

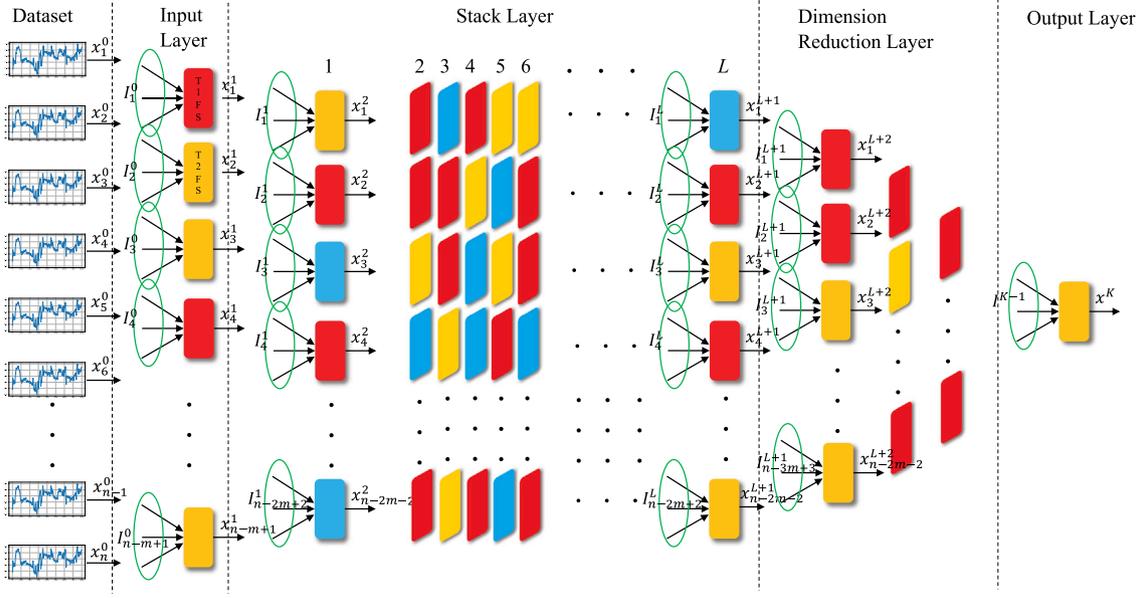


Fig. 2. Hierarchical structure of the proposed hybrid deep fuzzy model.

when $k \rightarrow 0$, the learning rate α is limited to the interval $[0, +\infty)$, and when $k \rightarrow +\infty$, the learning rate α is limited to a constant 0.01.

4. The proposed hybrid deep Fuzzy model

Due to the powerful ability for handling uncertainties and better interpretability, modular fuzzy models, especially modular deep fuzzy model, are regarded as one of the promising methods for dealing with the big data applications. Inspired by the ideas of the SAE (Vincent et al., 2010) and the DCFS (Wang, 2019), a novel stacked modular hybrid deep fuzzy model (HDFM) is presented in this section. In detail, the new structure of the proposed HDFM is described and the auto-construction strategy of each layer is given based on the optimized T1TSKFM and the IT2TSKFM. Then, the hierarchical data-driven learning strategy of the proposed HDFM is detailed given in layered by layered way. In addition, both the layered interpretability and the rule interpretability is respectively analyzed.

In this section, the hierarchical structure of the proposed HDFM will be given in detailed. Then, the auto-construction strategy will be presented.

4.1. The hierarchical structure and the performance evaluation metrics

A. The Hierarchical Structure of the HDFM

The hierarchical structure of the proposed HDFM consists of the input layer, the stack layer, the dimension reduction layer and the output layer, and is automatically constructed in stacking manner by using the optimized T1TSKFM and the IT2TSKFM. As shown in Fig. 2, each box is a component of the HDFM and represents a TSK fuzzy module (red boxes represent the trained T1TSKFMs, yellow boxes represent the trained IT2TSKFMs, blue boxed represent the training and updating T1TSKFMs or the IT2TSKFMs). The dimension of the HDFM and the input dimension of each module are adaptive according to the dimension of the data sample and the number of former modules. Without loss of generality, assuming that the dimensions of the l th layer in the HDFM is M , and can be denoted as $I^l = (x_1^l, \dots, x_M^l)$. Similarly, assuming that input dimension of the n th TSK fuzzy module (denoted as TSK-FM $_n^l$ is C , and can be expressed as $I_n^l = (x_{n,1}^l, \dots, x_{n,C}^l)$. Then, the r th fuzzy rule in TSK-FM $_n^l$ can be expressed as follows.

IF $x_{n,1}^l$ is $(\tilde{X}_1^r)_n^l$ and, ..., and $x_{n,C}^l$ is $(\tilde{X}_C^r)_n^l$

THEN $(y_r)_n^l$ is $g_r(x_{n,1}^l, \dots, x_{n,C}^l)_n^l$ (4.1)

It is important to note that for different types of TSK-FM $_n^l$ (T1TSKFM or IT2TSKFM), the expression of $g_r(x_{n,1}^l, \dots, x_{n,C}^l)_n^l$ is different.

T1TSKFM: $y_n^l(x_{n,1}^l, \dots, x_{n,C}^l)$

$$= \frac{\sum_{l=1}^R \prod_{c=1}^C \mu(\tilde{X}_c^r)_n^l(x_{n,c}^l) \cdot (\omega_{r,0} + \sum_{c=1}^C \omega_{r,c} x_{n,c}^l)_n^l}{\sum_{l=1}^R \prod_{c=1}^C \mu(\tilde{X}_c^r)_n^l(x_{n,c}^l)} \quad (4.2)$$

IT2TSKFM: $y_n^l(x_{n,1}^l, \dots, x_{n,C}^l)$

$$= \alpha \frac{\sum_{l=1}^R \prod_{c=1}^C \underline{\mu}(\tilde{X}_c^r)_n^l(x_{n,c}^l) \cdot (\omega_{r,0} + \sum_{c=1}^C \omega_{r,c} x_{n,c}^l)_n^l}{\sum_{l=1}^R \prod_{c=1}^C \underline{\mu}(\tilde{X}_c^r)_n^l(x_{n,c}^l)} + \beta \frac{\sum_{r=1}^R \prod_{c=1}^C \bar{\mu}(\tilde{X}_c^r)_n^l(x_{n,c}^l) \cdot (\bar{\omega}_{r,0} + \sum_{c=1}^C \bar{\omega}_{r,c} x_{n,c}^l)_n^l}{\sum_{l=1}^R \prod_{c=1}^C \bar{\mu}(\tilde{X}_c^r)_n^l(x_{n,c}^l)} \quad (4.3)$$

$y_n^l(x_{n,1}^l, \dots, x_{n,C}^l)$ is the output of TSK-FM $_n^l$, and then is regarded as the input for the next layer.

As shown in Fig. 2, the stacked HDFM is built in stacking manner by the optimized T1TSKFMs or IT2TSKFMs. The dimension of each layer in the HDFM is adaptive.

Input layer: The input layer implements the mapping from the source dataset to the model. Here, assuming that the dimension of the input dataset is M and the sliding window size of the fuzzy module in this layer is C , then the initialed $M - C + 1$ T1TSKFMs will be automatically generated.

For the $M - C + 1$ T1TSKFMs in the input layer, the input data can be expressed as $I_n^0 = (x_{n,1}^0, \dots, x_{n,C}^0)$, $n = 1, \dots, M - C + 1$. Then, the dimension of output is $M - C + 1$, and can be denoted as $y_n^0(x_{n,1}^0, \dots, x_{n,C}^0)$.

Stack layer: It is clearly seen that the dimension of the stack layer is $M - C + 1$ since the dimension of the input layer is M . And assuming that the sliding window size of the fuzzy module in the stack layer also is C , thus the width of the stack layer is $M - 2C + 2$. Similarly, assuming

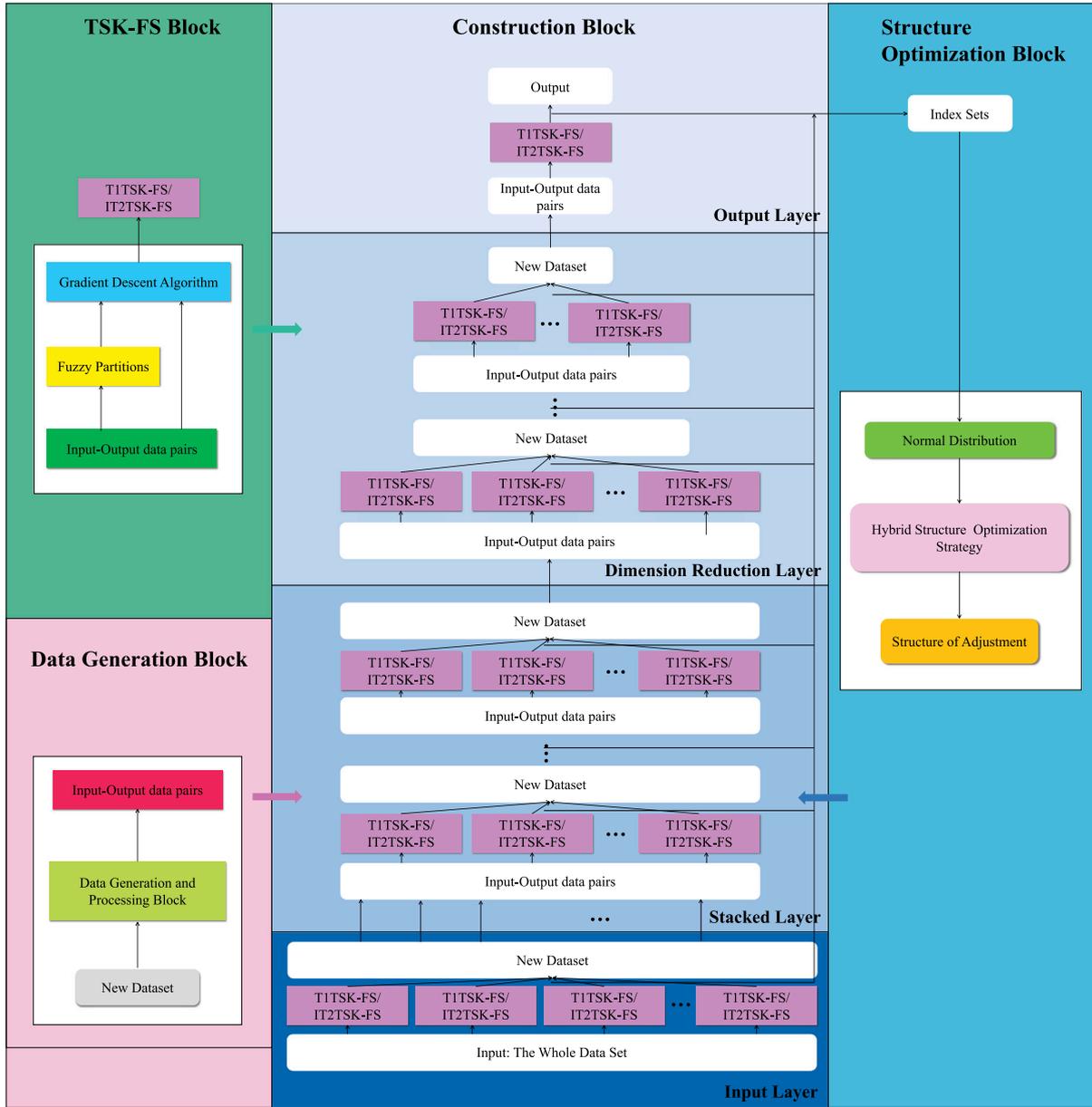


Fig. 3. Flow chart of the proposed HDFM.

that the depth of the stack layer is L , thus the total number of the fuzzy modules is $(M - 2C + 2)L$. For anyone fuzzy module in this layer, the input can be denoted as $I_n^l = (x_{n,1}^l, \dots, x_{n,M-C+1}^l)$ and the output is $y_n^l(x_{n,1}^l, \dots, x_{n,M-C+1}^l)$ with $l = 1, 2, \dots, L$ and $n = 1, \dots, M - C + 1$. It is easily know that the width of this layer will naturally decrease as the depth increase, due to there will be lack of $C(C - 1)$ input for the $C - 1$ th module in each layer. In order to solve the dimension decrease of width, the $C(C - 1)$ outputs of the former layer will be randomly selected as the input for the $(C - 1)$ th module in any layer.

Dimension reduction layer: In this layer, the width will naturally decrease until to be C fuzzy modules as the depth increase. The output of the C fuzzy modules, i.e., the last layer of the dimension reduction layer, will consist of the input of the next layer (output layer). Assume that the depth of the HDFM is K , then the input of the C fuzzy modules is $x_n^{K-L-2} = (x_{n,1}^{K-L-3}, \dots, x_{n,C}^{K-L-3})$ and the corresponding output is $y_n^{K-L-2}(x_{n,1}^{K-L-3}, \dots, x_{n,C}^{K-L-3})$.

Output layer: taking the outputs of the C fuzzy modules $y_n^{K-L-2}(x_{n,1}^{K-L-3}, \dots, x_{n,C}^{K-L-3})$, $n = 0, 1, \dots, C - 1$, as the inputs, i.e., $I_n^{K-L-1} =$

$y_n^{K-L-2}(x_{n,1}^{K-L-3}, \dots, x_{n,C}^{K-L-3})$, the output layer will provide the final forecasting result $y_n^{K-L-1}(x_{n,1}^{K-L-2}, \dots, x_{n,C}^{K-L-2})$.

B. Performance Evaluation Metrics:

In this work, the mean square error ($RMSE$) and the training time (T) are chosen as the key evaluation indicators for each of fuzzy modules. The proposed HDFM updates the hybrid structure according to the performance evaluation metrics ($RMSE$ and T). $RMSE$ measures the prediction accuracy of different fuzzy modules. At the same time, the training time is also considered. That is to say, we expect to minimize the training time overhead while obtaining superior prediction accuracy. For anyone fuzzy module k , the $RMSE$ and T are respectively defined as follows

$$RMSE_k = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y}^{(n)} - y^{(n)})^2}, \quad (4.4)$$

$$T_k = \sum_{n=1}^N (t_{\text{end}} - t_{\text{start}})_n. \quad (4.5)$$

For the k th module, the proportion of the $RMSE_k$ and the T_k in all fuzzy modules can be obtained as follows

$$WRMSE_k = \frac{RMSE_k}{\sum_{k=1}^K RMSE_k}, \quad (4.6)$$

$$WT_k = \frac{T_k}{\sum_{k=1}^K T_k}. \quad (4.7)$$

Then, in order to balance the training time and forecasting precision, a comprehensive evaluation metrics is introduced and defined as

$$Index_k = \frac{1}{p \times WRMSE_k + q \times WT_k}, \quad (4.8)$$

where p and q are balancing factors, and $p + q = 1$.

For the proposed HDFM, all the components (the T1TSKFM or the IT2TSKFM) are automatically constructed and updated, according to the evaluation metrics. In the begin, all the modules of the HDFM are initiated by the T1TSKFM. Then, the $Index_k$ is calculated for everyone T1TSKFM. The corresponding T1TSKFM will be replaced by the IT2TSKFM, if the $Index_k$ exceeds the threshold. The process will repeat in layer-by-layer way, until all the modules are successfully updated.

4.2. Data-driven learning strategy of the HDFM

The data-driven learning flowchart of the proposed HDFM is shown in Fig. 3. The flowchart mainly consists of the data generation block, the fuzzy module design block, the stacked HDFM construction block, and the hybrid structure optimization block.

Assume that, the input variables $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is derived from time series data ($S(t-n), S(t-n+1), \dots, S(t-1)$) the output y is $S(t)$, then the input-output data pair for the proposed HDFM can be expressed as

$$[x_1(k), x_2(k), \dots, x_n(k); y(k)] = [S(k), S(k+1), \dots, S(k+n-1); S(k+n)], \quad (4.9)$$

where $k = 1, 2, \dots, p$, p is the total number of training data pairs generated from the data generation block.

The input layer can be denoted as the zero-layer, and the input-output data pair of the i th fuzzy module is denoted as

$$DS_i^0 = \{x_i^0(k), x_{i+1}^0(k), \dots, x_{i+C-1}^0(k); y_i^0(k)\}_{k=1}^p, \quad (4.10)$$

where $i = 1, 2, \dots, M - C + 1$.

In the stack layer, the output obtained from the input layer is used as input for the current layer. The same process is repeated for the subsequent layers, until to the last layer. And the corresponding $M - 2C + 2$ fuzzy modules is constructed. The training data set for the i th fuzzy module in the l th layer is expressed as

$$DS_i^l = \{x_i^l(k), x_{i+1}^l(k), \dots, x_{i+C-1}^l(k); y_i^l(k)\}_{k=1}^p, \quad (4.11)$$

where $i = 1, 2, \dots, M - 2C + 2, l = 1, 2, \dots, L$ and L is the depth of the stack layer.

In the dimension reduction layer, the width decreases with the increase of the depth. Similarly, the training data set for the i th fuzzy module in the h th layer is

$$DS_i^h = \{x_i^h(k), x_{i+1}^h(k), \dots, x_{i+C-1}^h(k); y_i^h(k)\}_{k=1}^p, \quad (4.12)$$

where $i = 1, 2, \dots, M - 3C + 3, h = 1, 2, \dots, H$, H denotes the depth of the dimension reduction layer, and $H = (M - 4C + 3)/C$. There are C fuzzy modules in the last layer of dimension reduction layer, and the output set is

$$\{y_1^H(k), y_2^H(k), \dots, y_C^H(k)\}_{k=1}^p. \quad (4.13)$$

This will be accepted as the input data sequence for the output layer.

In the output layer, the final training dataset is represented as

$$DS_{\text{Output}} = \{x_1(k), x_2(k), \dots, x_C(k); y(k)\}_{k=1}^p \quad (4.14)$$

From Fig. 3, it can be clearly seen that the expected output data y is considered in every layer for the training data. Hence, the proposed stacked HDFM has stronger robustness. On the other hand, everyone fuzzy module is optimized by the gradient descent based module optimization strategy, and is updated by the auto-construction strategy. This will powerfully guarantee the overall outstanding performance of the HDFM.

Even if one of the fuzzy modules fails, the output of the other TSKFM will compensate for it to achieve high performance. The learning algorithm of the proposed HDFM is given in algorithm 1.

Algorithm 1: The proposed HDFM Learning Algorithm

Require:
 $W_{\text{InputLayer}}$: The width of Input Layer;
 $D_{\text{StackLayer}}$: The depth of Stack Layer;
 M : The input dimension of each T1TSKFM or IT2TSKFM;
 $S(t)$: The input time series;
 T : Model learning times;

Initialize:
 $W_{\text{StackLayer}} \leftarrow W_{\text{InputLayer}} - 2M - 2$
 $W_{\text{DRLayer}} \leftarrow W_{\text{InputLayer}} - 3M - 3$; // This only represents the width of the first layer
 $D_{\text{DRLayer}} \leftarrow \frac{W_{\text{InputLayer}} - 3M - 3}{M - 1}$
 $n \leftarrow W_{\text{InputLayer}} + W_{\text{StackLayer}} * D_{\text{StackLayer}} + (\frac{W_{\text{DRLayer}} - M}{M} + 1) + 1$; // The number of All TSK-FSS
Random $ST[n]$; // The structural parameters of HDFM is randomly initialized, which value is chosen form $\{-1, 1\}$, and indicates type of TSK-FSS respectively.
Random $JN[n]$; // The structural indicators of HDFM is randomly initialized.
 $t \leftarrow 0$;

Compute:
Generate dataset of Input Layer, DS_i^0 , with Eq.(4.4) and Eq.(4.5)
 $i = 1, 2, \dots, W_{\text{InputLayer}}$.

while $t < T$ **do**
 $t \leftarrow t + 1$;
for $n = 1, \dots, W_{\text{InputLayer}}$ **do**
Train the n -th module in the input layer and compute its output;
update the partial indicators $JN[1 \sim W_{\text{InputLayer}}]$;
end
Generate input dataset of Stack Layer from the output of all models of Input Layer;
for $m = 1, \dots, D_{\text{StackLayer}}$ **do**
for $n = 1, \dots, W_{\text{StackLayer}}$ **do**
Train the n -th module in the m -th Stack Layer and compute its output;
end
update the partial indicators
 $JN[W_{\text{InputLayer}} + 1 \sim W_{\text{InputLayer}} + 1 + W_{\text{StackLayer}} * D_{\text{StackLayer}}]$;
end
Generate input dataset of Dimension Reduction Layer from the Stack Layer;
for $m = 1, \dots, D_{\text{DRLayer}}$ **do**
 $W_{\text{DRLayer}} \leftarrow W_{\text{InputLayer}} - (3 + m) * M - (3 + m)$;
for $n = 1, \dots, W_{\text{DRLayer}}$ **do**
Train the n -th module in the m -th Stack Layer and compute its output;
end
update the partial indicators of JN ;
end
Generate input dataset of Output Layer from the Dimension Reduction Layer;
Train the last module in Output Layer and compute its output;
update the last indicators of JN ;
 $N(\mu, \sigma) \sim JN$; //Form a set of indicators into a normal distribution
for $i = 1, \dots, n$ **do**
if $JN[i] < \mu - 3\sigma$ **then**
| $JN[i] \leftarrow \sim JN[i]$;
end
end
end

4.3. Interpretability analysis for the proposed HDFM

The designed fuzzy modules (i.e., the T1TSKFM and the IT2TSKFM) perfectly inherit the interpretability advantages (including the structural interpretability and the rule interpretability) of the fuzzy logic system. Moreover, benefit from the stacked hierarchical structure, the working process of the proposed HDFM can easily understand.

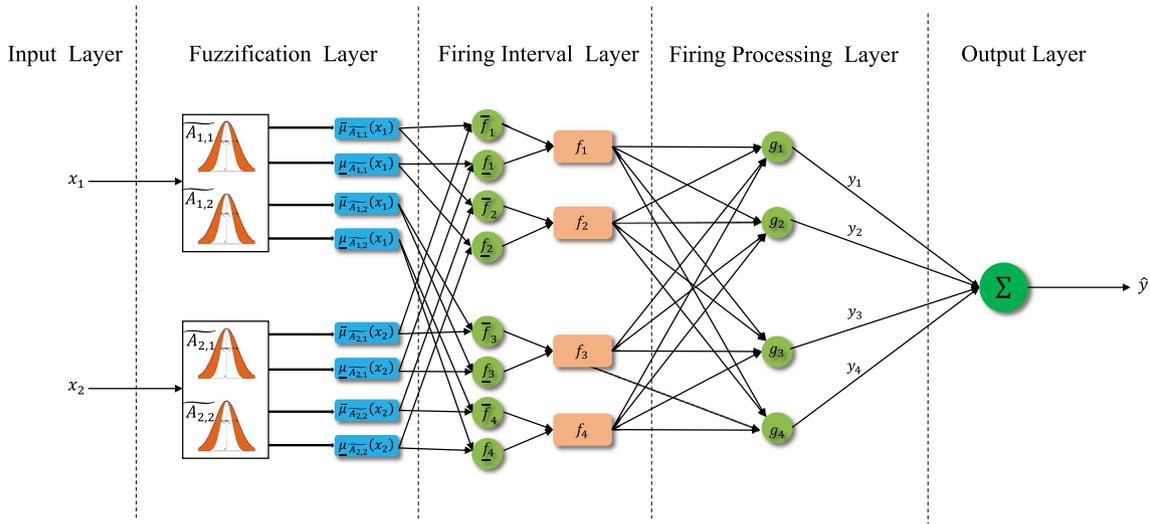


Fig. 4. Structural interpretability of the designed IT2TSKFM.

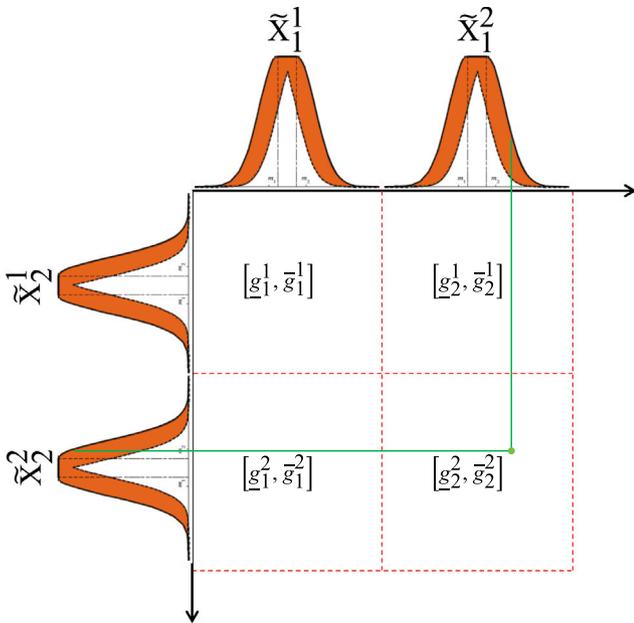


Fig. 5. Schematic diagram of rule interpretability of the IT2TSKFM.

A. Interpretability of Structure

Taking the designed IT2TSKFM as an example, the hierarchical structure consists of the input layer, the fuzzification layer, the interval firing layer, the firing process layer, and the output layer, as shown in Fig. 4. The input layer extracts the training/testing samples from the source dataset and transforms those into the standard input style. The fuzzification layer converts the extracted samples into the IT2FSs using the single-valued fuzzifier. The interval-firing layer computes the fired degree for each fuzzy rule and obtains the corresponding interval fire degree. The firing process layer computes the normalized firing degree of each fuzzy rule. The output layer gives the final crisp output by BMM algorithm. Therefore, we can clearly interpret how the IT2TSKFM works and understand how the data flows in the IT2TSKFM in layer-by-layer way.

B. Interpretability of Rules

The characteristics of the TSK fuzzy rules naturally guarantee that which rule is fired can be known clearly, and its firing degree can be compute easily. Without loss of the generations, suppose that there are

two IT2FSs, then the fuzzy rules can be denoted as

$$Rule_i^j : \text{ IF } x_1 \text{ is } \widetilde{X}_{1,i} \text{ and } x_2 \text{ is } \widetilde{X}_{2,j} \text{ THEN } y_r \text{ is } [g(x_1, x_2), \bar{g}(x_1, x_2)]_i^j, \tag{4.15}$$

The fired interval weight is $[g_1^1, \bar{g}_1^1], [g_2^1, \bar{g}_2^1], [g_1^2, \bar{g}_1^2]$ and $[g_2^2, \bar{g}_2^2]$, as shown in Fig. 5. In addition the fired degree also can be easily calculated for the corresponding input variables, according to the designed fuzzy rules. Consequently, the interpretability of the fuzzy modules is clear.

Therefore, the proposed HDFM has better interpretability benefiting from the module stacked architecture and the corresponding interpretable components, i.e., the structural interpretability and the rule interpretability. Thus, we can clearly observe how the data flows and accurately locate which fuzzy rule is fired.

5. Experiments

In this section, the comparison models will be briefly introduced, and then the performance evaluation indicators will be given. In addition, the proposed HDFM is applied in three different application scenarios and the corresponding results are respectively analyzed.

5.1. Comparison models and evaluation indicators

A. Comparison Models

At present, many kinds of deep fuzzy systems have been proposed, i.e., the DIRM-DFM (Li et al., 2020), the IT2DIRM-DFM (Peng et al., 2021), and the prediction performance prior the FWSIRM-FM, the SIRM-FM, and the SAE models. Hence, the DIRM-DFM and the IT2DIRM-DFM models are taken as the comparison models. And, DCFS and ANFIS are also considered as two common methods. In addition, in order to further fully evaluate the performance of the proposed HDFM, the pure interval type-2 deep fuzzy model (pIT2DFM) and the pure type-1 deep fuzzy model (pT1DFM) are also considered as the comparisons. The detailed configurations for the HDFM and its comparisons are listed in Table 1.

B. Evaluation Indicators

In order to fully evaluated the performance of the proposed HDFM, the root mean square error (RMSE), the mean absolute error (MAE), the correlation coefficient (R^2), and the training time are chosen as the evaluation indicators. And, the corresponding definitions are as follows

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (\hat{y}^{(j)} - y^{(j)})^2}, \tag{5.1}$$

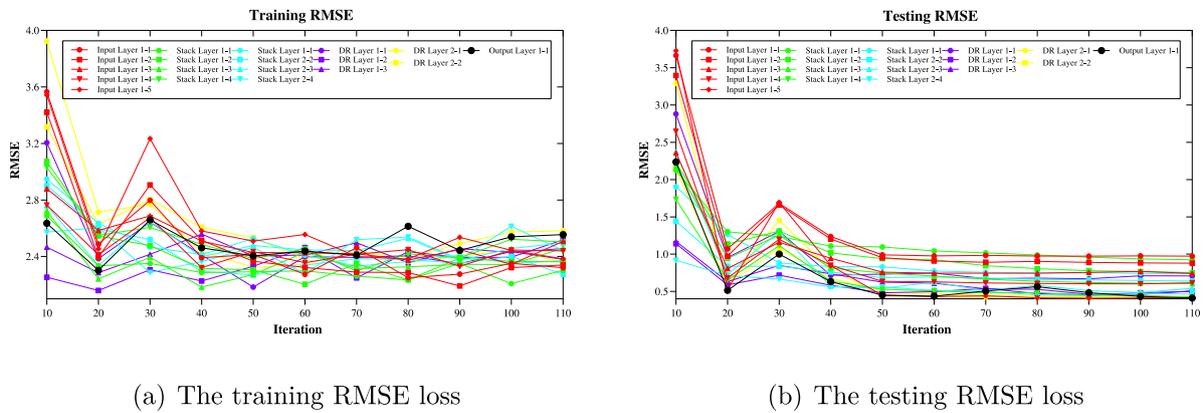


Fig. 6. The training and testing RMSEs of each modules of the proposed HDFM.

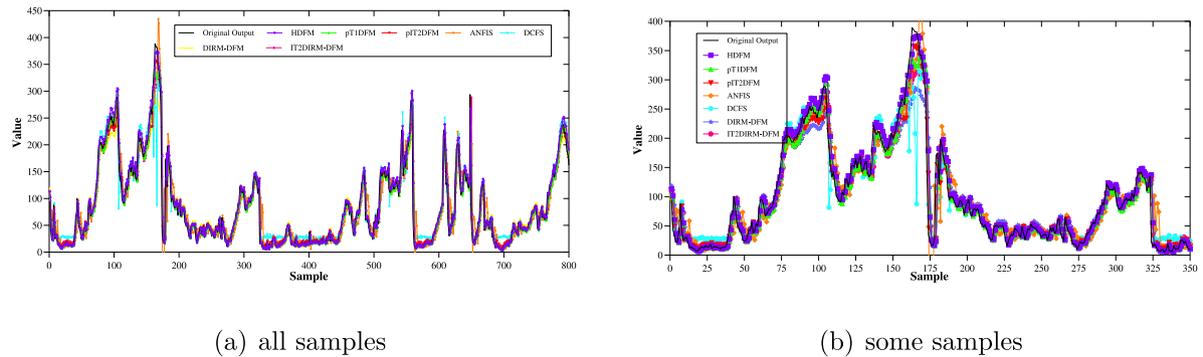


Fig. 7. The results of all models in experiment 1.

Table 1
The configuration of each model in Experiment 1.

Models	nMFs	No. of modules/nodes	MF	α, β	epoch	λ	P	γ	N_{bs}
HDFM	2	19	Gaussian	0.5	110	0.05	0.5	0.01	256
pIT2DFM	2	19	Gaussian	0.5	110	0.05	0.5	0.01	256
pT1DFM	2	19	Gaussian	-	110	0.05	0.5	0.01	256
IT2DIRM-DFM	2	15	Triangular	0.5	-	-	-	-	-
DIRM-DFM	2	15	Triangular	-	-	-	-	-	-
ANFIS	-	161	Triangular	-	10	-	-	-	-
DCFS	2	15	Triangular	-	-	-	-	-	-

$$MAE = \frac{1}{n} \sum_{j=1}^n |\hat{y}^{(j)} - y^{(j)}|, \tag{5.2}$$

$$R^2 = 1 - \frac{\sum_{j=1}^n |\hat{y}^{(j)} - y^{(j)}|^2}{\sum_{j=1}^n |\hat{y}^{(j)} - \bar{y}^{(j)}|^2}. \tag{5.3}$$

According to the definitions, we can clearly observe that the smaller value of the RMSE and MAE means the higher prediction accuracy, and the larger value of R^2 indices the stronger fitting performance of the model. Moreover, the smaller training time means the faster convergence of the model.

5.2. PM2.5 prediction

A. dataset

The dataset consists of the weather data and the air pollution data collected by the American embassy in Beijing from 2010 to 2014. In addition, the resource dataset can be download from <http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data> For the HDFM and its

comparisons, 2400 samples are chosen as the train data, and 600 samples are used as the test data.

B. Model Configuration

In the experiment, the input layer of HDFM consists of 5 modules, the stack layer consists of 8 modules in 2 layers, the dimension reduction layer consists of 5 modules in 2 layers, and the output layer consists of 1 module in 1 layer. In other words, the HDFM model used in the experiment consists of 19 modules.

As the Table 1 listed, nMFs represents the number of fuzzy sets corresponding to each input, and MF represents the types of the membership functions. α and β are the parameter of BMM algorithm, epoch represents the number of iterations of gradient descent algorithm, λ represents the regularization coefficient, P represents the DropRate, and γ represents the selected learning rate.

In fact, the DIRM-DFM proposed in Li et al. (2020) does not have the stack layer. However, in order to ensure the rationality of the comparison experiment as much as possible, the total number of layers set in the experiment is consistent with that of other depth fuzzy system models.

C. Result

In order to effectively help us understand the working details of the designed fuzzy module at different levels, the optimized antecedent and consequent parameters of the fuzzy rules based on the proposed gradient descent algorithm are given in detail, as shown in Tables A.1 and A.2 respectively.

In addition, in order to demonstrate the effectiveness of the gradient descent algorithm on the proposed fuzzy module, the RMSE iterative process of the training and testing of the proposed HDFM is shown in Fig. 6.

Fig. 6 respectively reflects the RMSE losses and the iterative process of different modules at different layers. In both the training process Fig. 6(a) and the testing process Fig. 6(b), it can be clearly observed that the RMSE losses decrease rapidly with iteration, and the losses also

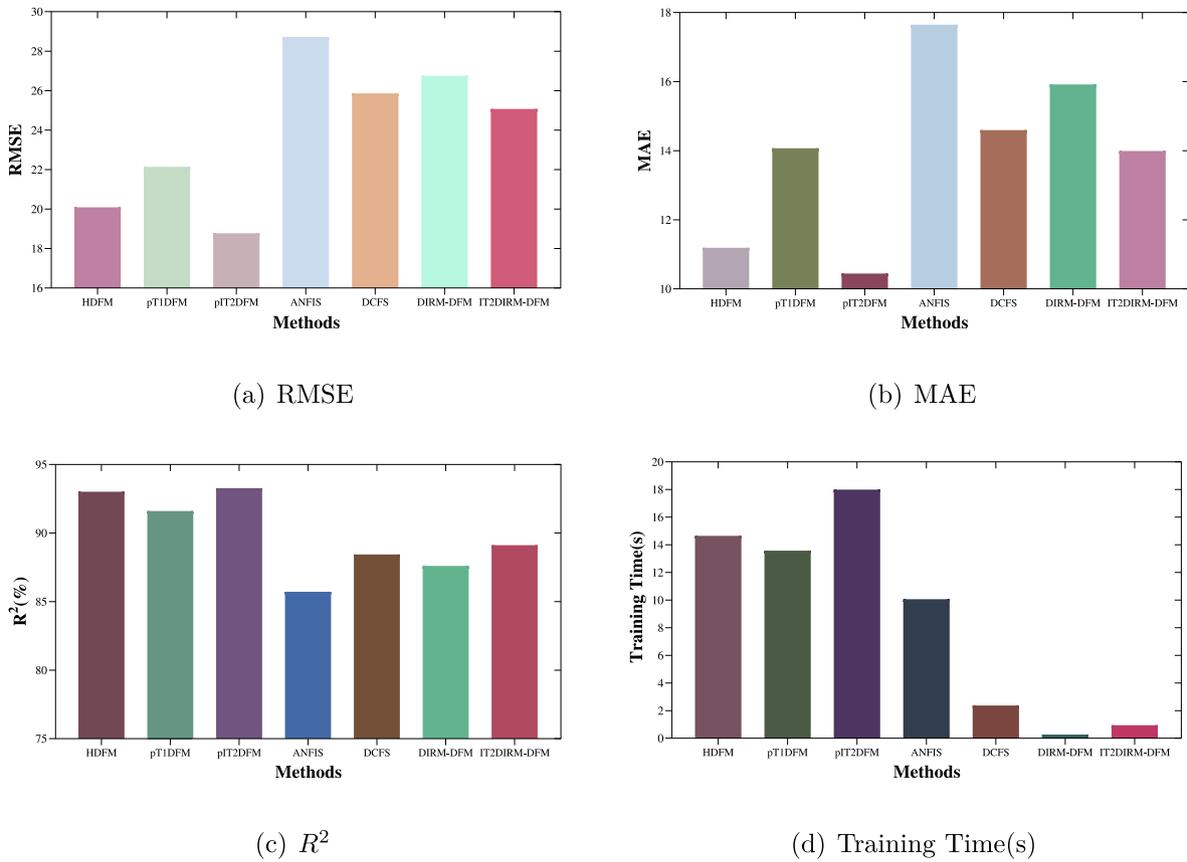


Fig. 8. Statistical results of all model performance indicators in experiment 1.

Table 2 Performance Indictors for each model in Experiment 1.

Model	HDFM	pT1DFM	pIT2DFM	ANFIS	DCFS	DIRM-DFM	IT2DIRM-DFM
RMSE	20.088221	22.142541	18.777954	28.720478	25.857866	26.760593	25.071705
MAE	11.188253	14.072682	10.445991	17.649337	14.594523	15.919494	13.991901
R ²	93.0138%	91.6091%	93.2796%	85.7186%	88.4354%	87.6048%	89.1174%
Training time(s)	14.652756	13.574561	17.995795	10.065363	2.366962	0.261919	0.938072

converge rapidly with the increase of HDFM depth. At the same time, according to the observation, the final RMSE loss in the testing process is smaller than that in the training process, which fully proves that the developed optimization algorithm has an important improvement in the generalization ability of the model.

In order to verify the validity of the proposed HDFM prediction, the prediction results of HDFM and its comparison models are plotted in Fig. 7(a), and the first 350 samples are replotted in Fig. 7(b) for a clearer observation of the prediction effect. It can be clearly seen that the proposed HDFM can effectively track the changing trend of PM2.5 index and achieve accurate prediction. The predicted effect was better than that of DIRM-DFM, IT2DIRM-DFM, pT1DFM, ANFIS and DCFS, and slightly worse than that of pIT2DFM.

In order to further accurately evaluate the predictive performance and efficiency of the model, the numerical results of HDFM performance indicators and their comparisons are shown in Table 2. To further see the performance of the different models clearly, the visualization results are shown in Fig. 8. It is easy to see that the prediction accuracy of the proposed HDFM is better than that of DIRM-DFM, IT2DIRM-DFM, pT1DFM, ANFIS and DCFS. While pIT2DFM had better predictions and better fit, it had the worst training time of all the models. As the proposed HDFM requires training modules' parameters

iteratively, replacing module and updating architecture, the training time for the proposed HDFM is not optimal. Nevertheless, the proposed HDFM strikes a balance between predictive accuracy and training time.

Finally, the three-dimensional waterfall diagram of the absolute errors between the predicted results of all models and the real values in the testing process is provided, as shown in Fig. 9. In addition, the absolute errors of 20 consecutive samples randomly selected are also provided, as shown in Table 3.

5.3. PV power prediction

A. dataset

For the second experiment, we selected Belgian PV data, which can be downloaded from <https://www.elia.be/nl>. The PV data was collected from March 1, 2019 to August 31, 2019 (a total of 184 days), and each sample was sampled at a 15-minute interval. In the experiment, a total of 8000 samples are used for the experiment, and the training set and the testing set are divided by a ratio of 3:1.

B. Model Configuration

The detailed configuration of each model is similar to the configuration listed in Table 1, except that the number of iterations is modified (epoch = 120), and the two parameters of the BMM algorithm are modified to $\alpha = 0.3$ and $\beta = 0.7$.

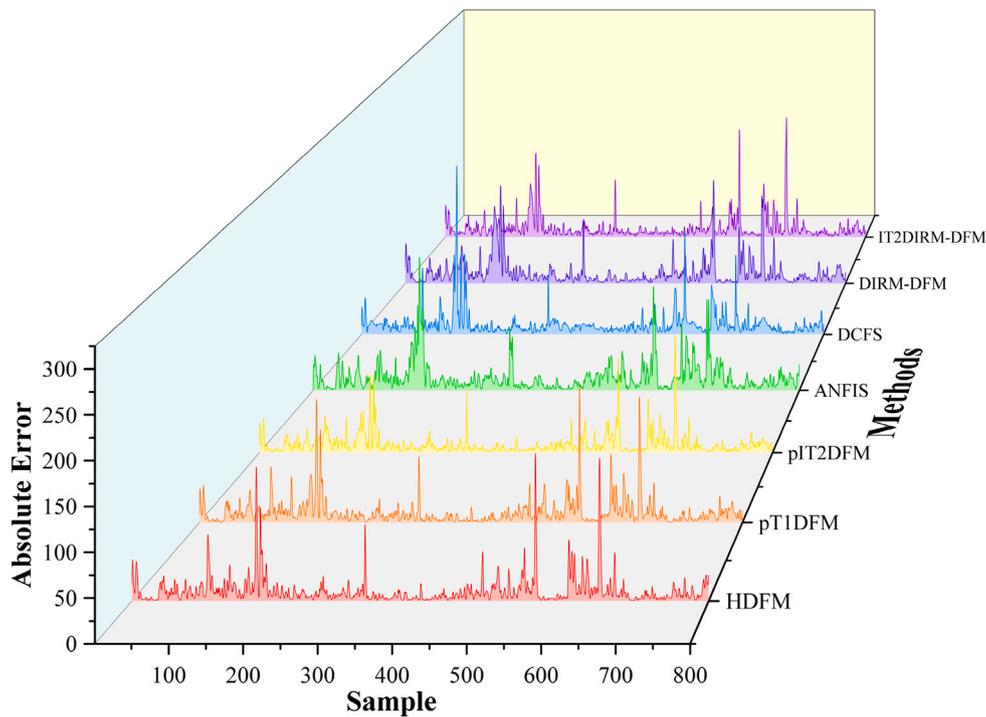


Fig. 9. The absolute errors of all models in experiment 1.

Table 3
The absolute errors of 20 consecutive samples randomly selected in Experiment 1.

Model	HDFM	pT1DFM	pIT2DFM	ANFIS	DCFS	DIRM-DFM	IT2DIRM-DFM
1	20.6799079	28.3918854	17.6032628	25.3888031	12.4821605	24.73036237	27.83616571
2	1.87541927	8.25448587	0.03766287	21.0504107	0.91415521	1.395277177	0.85573294
3	14.7543207	6.11171505	9.49403661	20.7322101	9.92392296	18.51520281	16.65909469
4	23.5332439	16.9353231	17.5709822	3.04916347	16.4207883	32.3303837	24.66975833
5	4.61919008	0.2218127	1.13007908	13.2171888	3.17943699	17.72808738	6.743419204
6	4.0837492	9.1257503	5.87143063	1.55556171	3.94619655	3.150228548	7.79009248
7	1.6225419	8.26293745	3.37254002	7.73371442	2.81262159	2.083779755	4.826048317
8	0.42974761	6.55758629	2.26046911	5.59864562	2.98233254	4.660177943	0.614133551
9	0.16242619	7.15163065	3.26633911	1.46479523	4.36983206	4.618690926	1.176068084
10	10.2849446	17.5244639	13.4766035	20.9151352	14.7733329	5.892351936	12.10232859
11	8.48007818	17.2980479	10.8443991	20.6135649	10.3690632	8.204090038	13.73056436
12	12.7913589	22.7631215	16.9111373	27.730521	14.0572433	13.22473015	17.14432464
13	7.6964895	19.0754327	12.0431798	27.382938	8.12163157	11.21785153	12.43950416
14	8.41891765	20.6969797	14.9489711	28.8609917	9.72560736	12.37527372	14.25064272
15	25.6114132	38.8455627	32.7101529	50.8151398	23.3434534	31.96796617	35.75520826
16	24.9215206	41.6753637	31.3424252	45.4223545	27.2078485	41.37121469	46.89138428
17	0.80719846	19.0689147	9.16936186	19.268556	13.5228501	23.02203003	23.66729343
18	1.74365771	17.2440896	13.4432061	23.9995667	5.15560448	19.44135634	8.780341114
19	11.5556248	6.75818593	3.36831729	12.9652619	21.9556542	12.11065209	5.045854965
20	10.5883425	6.74082669	6.32718729	9.95287552	19.1805778	11.0816062	0.50246105

Table 4
Performance Indicators for each model in Experiment 2.

Model	HDFM	pT1DFM	pIT2DFM	ANFIS	DCFS	DIRM-DFM	IT2DIRM-DFM
RMSE	33.136102	43.187805	31.397777	79.724196	66.474193	64.140267	50.320121
MAE	19.299784	23.535175	19.273752	48.962827	51.215905	40.720731	33.343199
R ²	99.2101%	98.6595%	99.2906%	95.4264%	96.8223%	97.0397%	98.1780%
Training time(s)	18.804246	17.740819	23.889196	1.460595	0.396264	0.66143	3.560936

C. Result

Specifically, for HDFM, Tables A.3 and A.4 show the learning optimal fuzzy rules of modules located in different layers.

In addition, the learning process of the rule parameters is shown in Fig. 10, and the RMSE loss decreases continuously with the iteration of the training process, which indicates that the developed gradient descent algorithm has the property of fast convergence, and reflects the effectiveness of the fuzzy system. The randomly initialized parameters

are constantly becoming accurate, and the predicted output of HDFM is constantly close to the real output.

In order to verify the fitting effect and prediction performance of the proposed HDFM, the prediction results of HDFM and its comparison model are plotted in Fig. 11(a), and 140 samples are replotted in Fig. 11(b) for a clearer observation of the prediction effect. It can be clearly seen that all models in the experiment can effectively track the change trend of photovoltaic energy generation, but there are differences in the prediction accuracy.

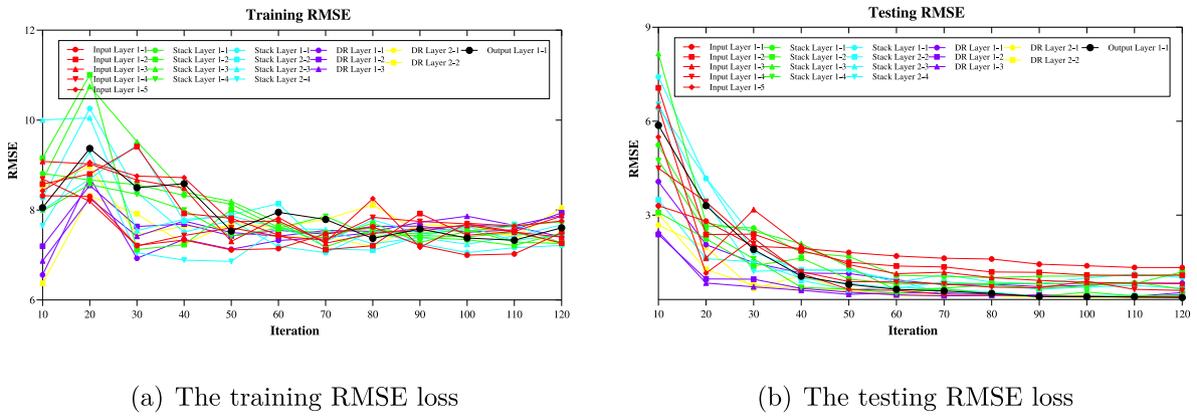


Fig. 10. The training and testing RMSEs of each modules of the proposed HDFM.

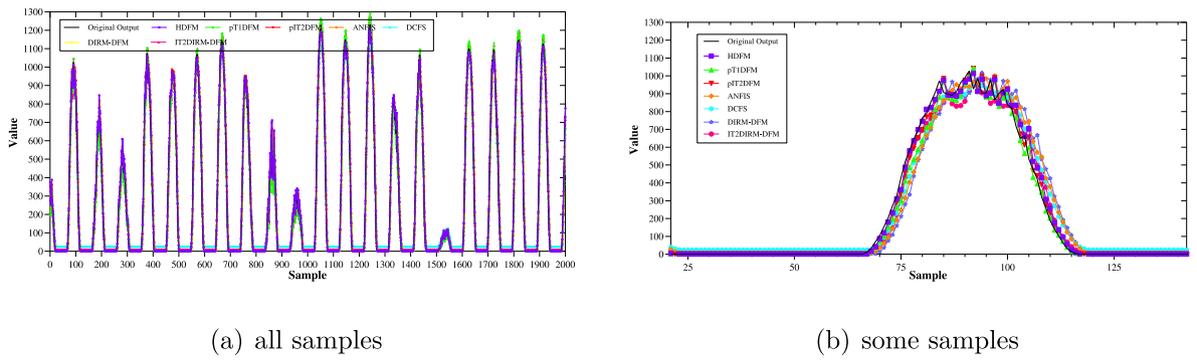


Fig. 11. The results of all models in experiment 2.

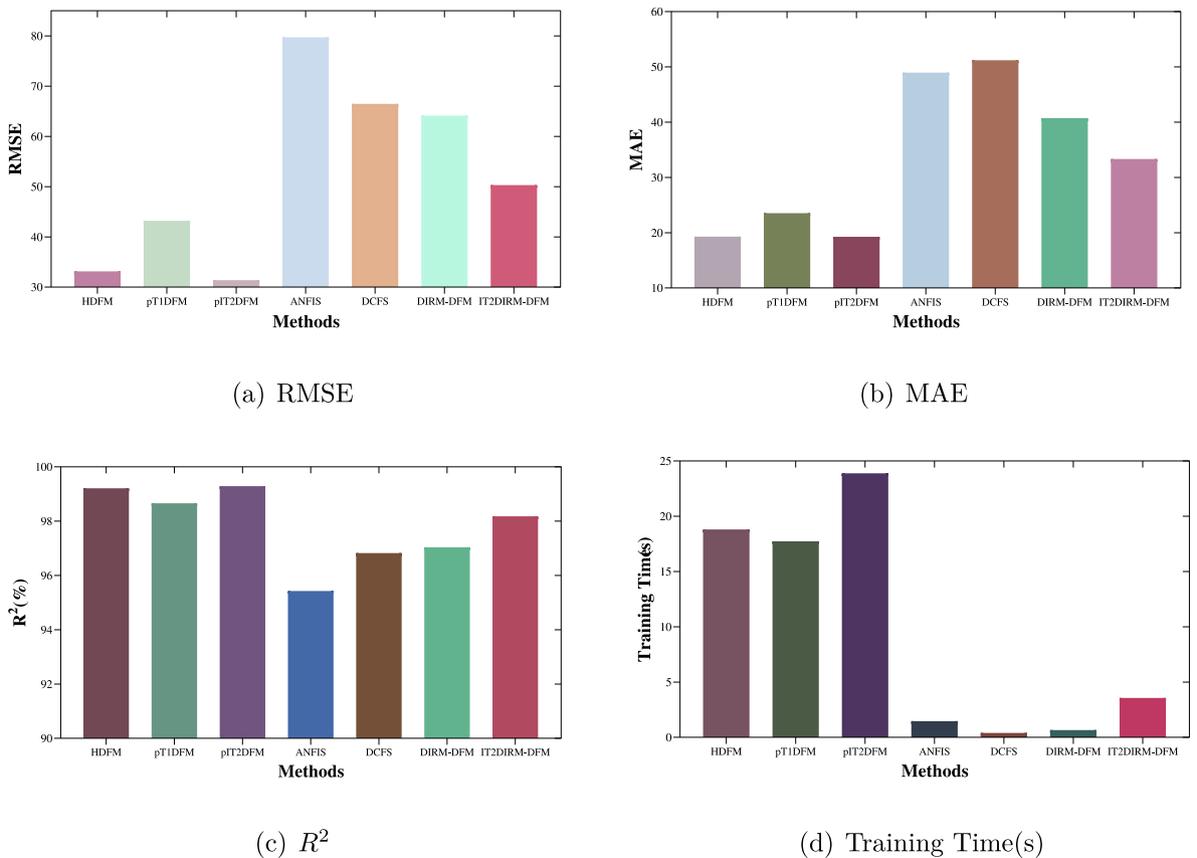


Fig. 12. Statistical results of all model performance indicators in experiment 2.

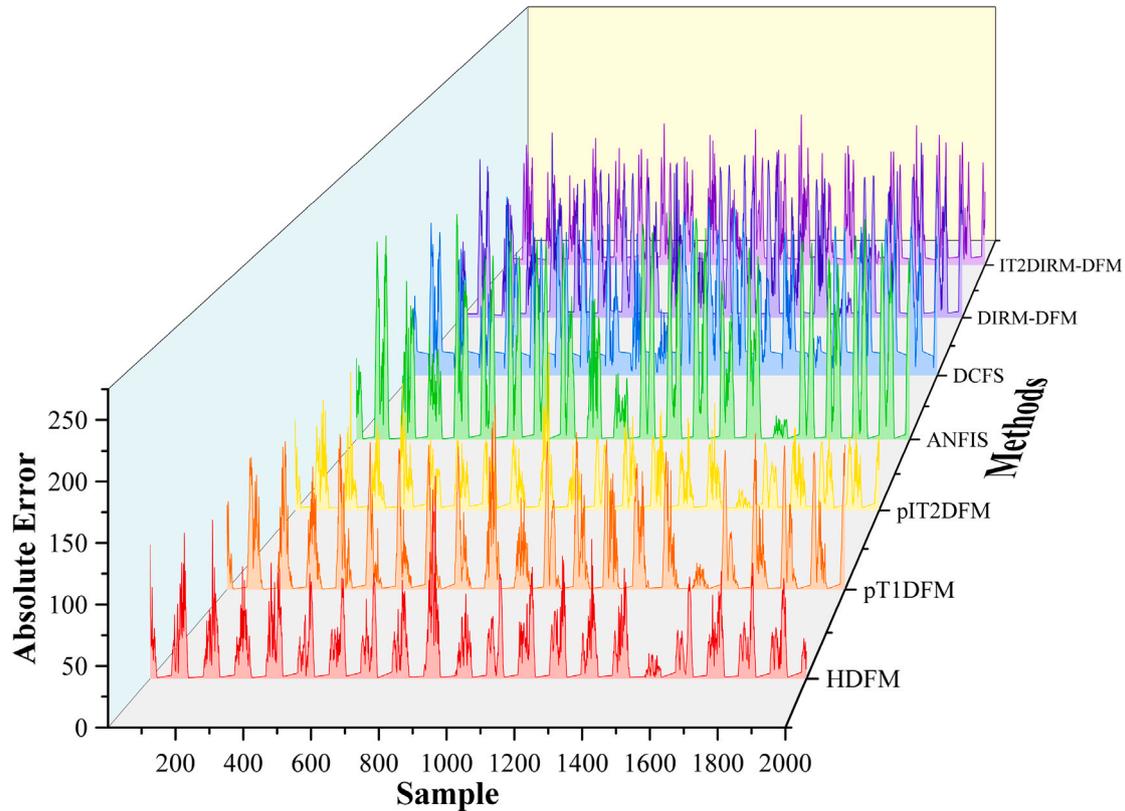


Fig. 13. The absolute errors of all models in experiment 2.

Specifically, statistical indicators RMSE, MAE, R^2 , and training time are shown in Table 4, and the corresponding visualization histogram is shown in Fig. 12. Table 4 and Fig. 12 both reflect that the RMSE, MAE and R^2 indicators of the proposed HDFM model are better than those of DIRM-DFM, IT2DIRM-DFM, pT1DFM, ANFIS and DCFS, but still slightly worse than pIT2DFM. However, in terms of training time, as all modules of pIT2DFM are interval type-2 TSK fuzzy modules (IT2TSKFs), the complexity of the system determines that it needs longer training time. The proposed HDFM is superior to pIT2DFM in this indicator.

Finally, in order to further analyze the differences in the predictive performance of each model in this experiment, the absolute errors of 20 continuous samples were randomly selected and listed in Table 5. And three-dimensional waterfall diagram of all test samples is drawn to reflect the global error analysis effect, as shown in Fig. 13.

5.4. Energy consumption prediction

A. dataset

This dataset is related to power consumption of three different distribution networks in the city of De Tuan in northern Morocco. The open dataset used in the experiment was sampled at a frequency of 10 min. And, 3000 samples were used for training, 1000 samples were used for testing. Dataset can be found from <http://archive.ics.uci.edu/ml/datasets/Power+consumption+of+Tetouan+city>.

B. Model Configuration

The detailed configuration of all models is almost the same as that in experiment 2, only the number of iterations is modified (epoch = 200), and the parameter nMFs is modified to 3.

C. Result

In order to effectively help us understand the working details of the designed fuzzy module at different levels, the antecedent and

consequent parameters of fuzzy rules are given in detail, as shown in Tables A.5 and A.6 respectively.

In addition, in order to demonstrate the effectiveness of the gradient descent algorithm with the Regularization, the DropRule and the AdaBound on the proposed fuzzy module, the training and testing RMSEs of the proposed HDFM is shown in Fig. 14.

Fig. 14 respectively reflects the decreasing process of RMSE losses of each HDFM module in the training process and the testing process. It can be clearly seen that the developed gradient descent algorithm with the Regularization, the DropRule and the AdaBound can achieve fast convergence of the fuzzy system and has strong generalization ability. Secondly, it can be observed that the performance of the module of the output layer (shown by Output Layer1-1 in Fig. 14) is not the best during the training process, but its RMSE loss can be minimized during the testing process.

In order to verify the effectiveness of the proposed prediction capability of HDFM, the prediction results of HDFM and its comparison models are drawn in Fig. 15(a). According to the observation, almost every model in the experiment can track the change trend of energy consumption data. However, in order to observe the prediction effect more clearly, we redraw the first 350 samples in Fig. 15(b). And, it can be clearly seen that HDFM, pT1DFM and pIT2DFM optimized by gradient descent algorithm have better prediction effect.

It is difficult to accurately measure the antecedent and consequent of each model simply by observing the prediction curve. Here, we further accurately evaluate the prediction performance of all models. The numerical results of the performance indicators of HDFM and its comparison methods are shown in Table 6. In order to visually observe the differences in performance indicators of different models, the visual results are shown in Fig. 16. It is easy to see that the prediction accuracy of the proposed HDFM is better than that of DIRM-DFM, IT2DIRM-DFM,

Table 5
The absolute errors of 20 consecutive samples randomly selected in Experiment 2.

Model	HDFM	pT1DFM	pIT2DFM	ANFIS	DCFS	DIRM-DFM	IT2DIRM-DFM
1	67.4747053	127.039044	92.6687117	181.844336	143.149248	149.6154077	111.123812
2	32.8254054	109.062774	64.4421123	204.724239	167.224388	186.9787661	87.66325918
3	3.8026523	96.9356469	42.2936802	180.618748	146.665255	144.2506845	23.34361181
4	9.64662286	112.450121	51.8895249	184.310428	131.012018	116.5173268	27.03398441
5	2.11039372	106.805087	42.5481045	161.507094	120.262404	107.5131957	44.73899462
6	19.3425163	128.19582	65.9594049	159.108202	135.688133	115.570651	69.59639627
7	23.9386835	77.560871	19.5131107	122.757449	107.736038	90.17373151	56.99781419
8	30.5670533	119.871129	72.2532555	142.413026	121.839821	85.76181936	90.44821853
9	38.7074387	93.6685612	46.7534627	134.982069	132.137042	118.0647807	119.3661812
10	59.8426916	93.1034242	59.7573249	149.657536	148.05456	121.8121107	157.7610436
11	69.4384572	63.1609841	80.1106389	43.9752524	24.0291304	4.851695323	55.10343682
12	3.97143733	12.7481078	4.68090531	21.1775916	15.5779799	69.27927357	18.71585069
13	2.49633177	20.6623577	4.10022245	27.3198002	10.2483354	3.221974439	47.71982092
14	24.08586024	9.9951501	23.4152845	39.7133886	30.0238825	21.4613319	81.48723918
15	49.6727299	74.1391567	48.6604789	34.3679817	84.7194465	78.85060156	129.9718946
16	25.3693062	33.2750572	16.366493	48.8678735	93.1104143	84.86558088	132.7605796
17	44.0937264	30.7160878	24.4756975	87.2669019	96.3387661	70.22364752	113.0132833
18	87.1191207	117.380847	115.546983	40.804853	35.2846302	53.95601224	34.14809436
19	73.1151984	69.8202144	58.1183306	30.1626407	20.970156	30.72544595	38.73721238
20	80.1519867	83.783714	95.2851705	96.2603148	42.803167	8.631554251	2.085682246

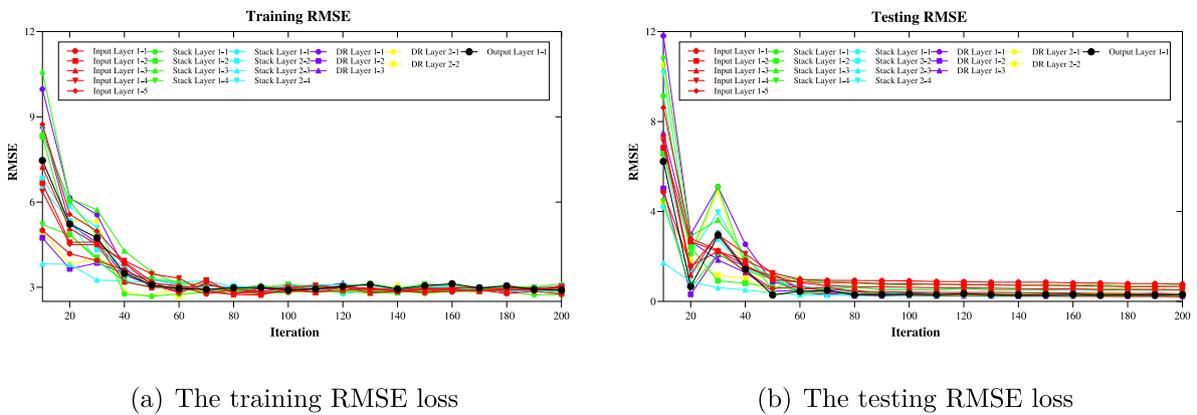


Fig. 14. The training and testing RMSEs of each modules of the proposed HDFM.

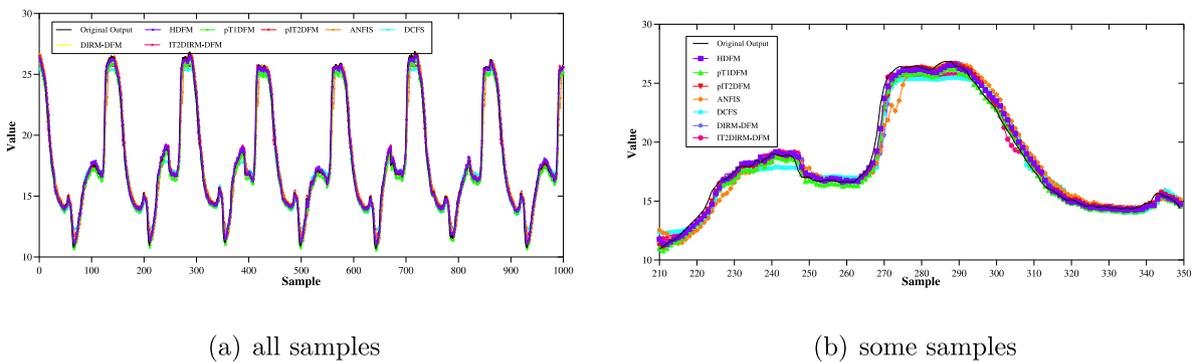


Fig. 15. The results of all models in experiment 3.

Table 6
Performance Indicators for each model in Experiment 3.

Model	HDFM	pT1DFM	pIT2DFM	ANFIS	DCFS	DIRM-DFM	IT2DIRM-DFM
RMSE	0.481945	0.536011	0.4471	1.012312	0.768686	0.807147	0.585615
MAE	0.343559	0.392338	0.288902	0.709316	0.578956	0.511931	0.40001
R ²	98.8038%	98.5281%	98.9705%	94.7224%	96.9575%	96.6447%	98.2338%
Training time(s)	34.804313	31.002932	37.285584	12.752908	0.244665	0.278579	1.435145

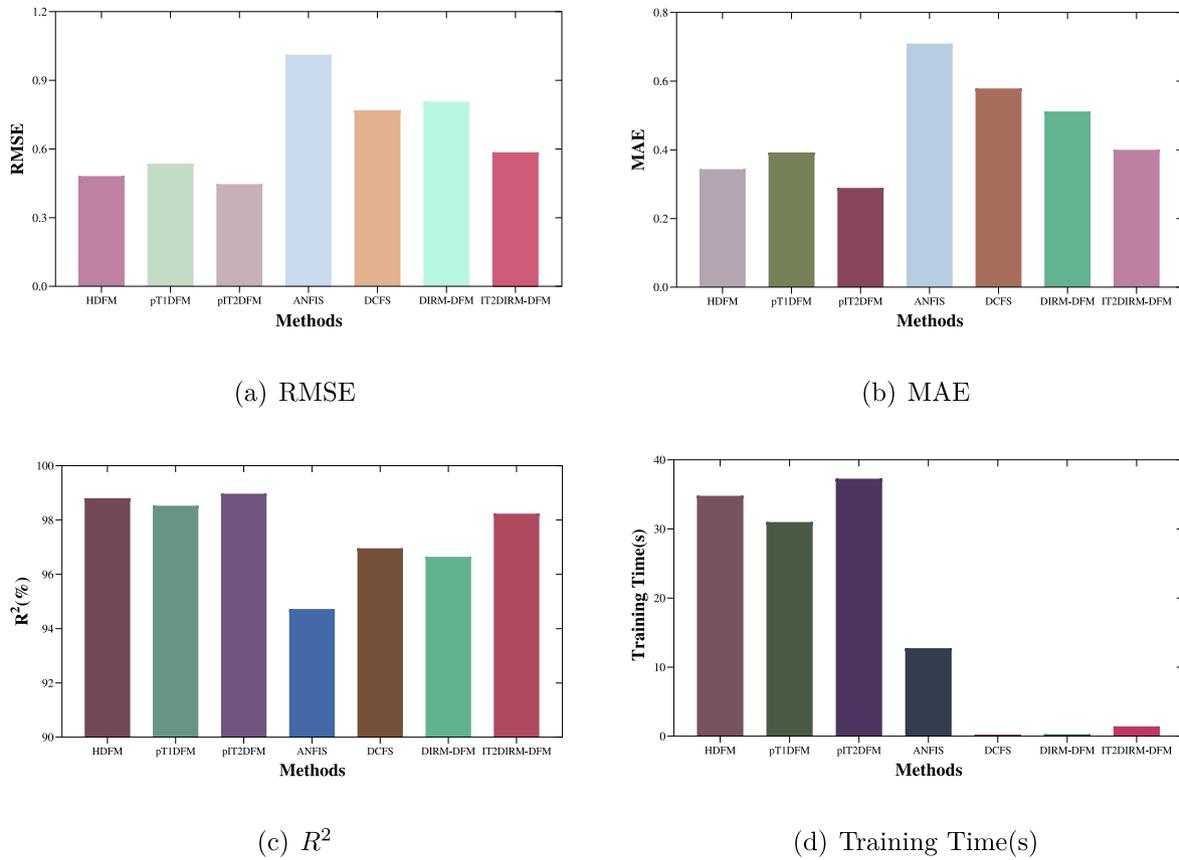


Fig. 16. Statistical results of all model performance indicators in experiment 3.

pT1DFM, ANFIS and DCFS. While pT2DFM had better predictions and better fit, it had the worst training time of all the models.

For a single sample, the absolute errors between the predicted value and the real value is shown in Table 7. The absolute errors of all samples will be shown by three-dimensional waterfall diagram, as shown in Fig. 17.

In this experiments, some key hyperparameters are chosen to test the robustness of HDFM. In addition, some less important parameters, such as α and β in the BMM method, are not discussed. As shown in Fig. 18, within a certain range of N_{bs} , P , γ and L , the proposed HDFM can maintain good predictive performance and generalization ability.

5.5. Summary and discussion

In the previous section, the proposed HDFM and comparison models were applied to three independent experiments in different scenarios, and the statistical results of each model were demonstrated in the form of graphs. Here, the above experimental results are summarized and discussed in more detail.

- From the process of RMSE loss iteration (Figs. 6, 10, 14), it can be seen that the gradient descent optimization algorithm developed in this study for deep fuzzy system can achieve fast convergence and improve the generalization performance of the proposed HDFM. According to the loss iteration diagram of different modules, it can be found that the input-hidden-output network structure of HDFM realizes the gradual approximation to the expected output. The output generated by each layer is a weak approximation to the expected output. And this weak approximation is passed as input to the next layer. In this stacking process, the approximation to the desired output is gradually

strengthened. In the end, the best output is obtained in the output layer. These information are reflected in Figs. 6, 10, 14, the testing RMSE of the fuzzy module tends to decrease as the depth increases, and the fuzzy module at the output layer obtains the lowest RMSE.

- As shown in Tables 2, 4 and 6, the proposed HDFM and its pure versions (pT1DFM and pT2DFM) are superior to the ANFIS, the DCFS, the DIRM-DFM and the IT2DIRM-DFM models in three statistical indicators, RMSE, MAE and R^2 . Taking the RMSE indicator as an example, compared with the above models, it increased by 30.0561%, 22.3129%, 24.9336% and 19.8769% respectively in experiment 1 and 58.4366%, 50.1519%, 48.3381% and 34.1493% respectively in experiment 2. In experiment 3, compared with the above models, the increases were 52.3916%, 37.3027%, 40.2903% and 17.7027%, respectively.
- In the comparison of the three deep fuzzy systems proposed in this paper, according to Figs. 8, 12 and 16, it can be seen intuitively that the HDFM, as a hybrid model, improves the prediction accuracy and needs more training time compared with pT1DFM. Compared with pT2DFM, it saves more training time and sacrifices less precision. Specifically taking Experiment 1 as an example, the HDFM's R^2 improves 1.5333% compared with pT1DFM, and correspondingly only loses 7.9427% in terms of time. However, it takes the pT2DFM 18.5767% longer training time than the HDFM to improve the R^2 indicator by 0.2849%.

6. Conclusions

Inspired by the neural network structure and the idea of parameter optimization, firstly, T1TSKFM and IT2TSKFM are given. Secondly, a

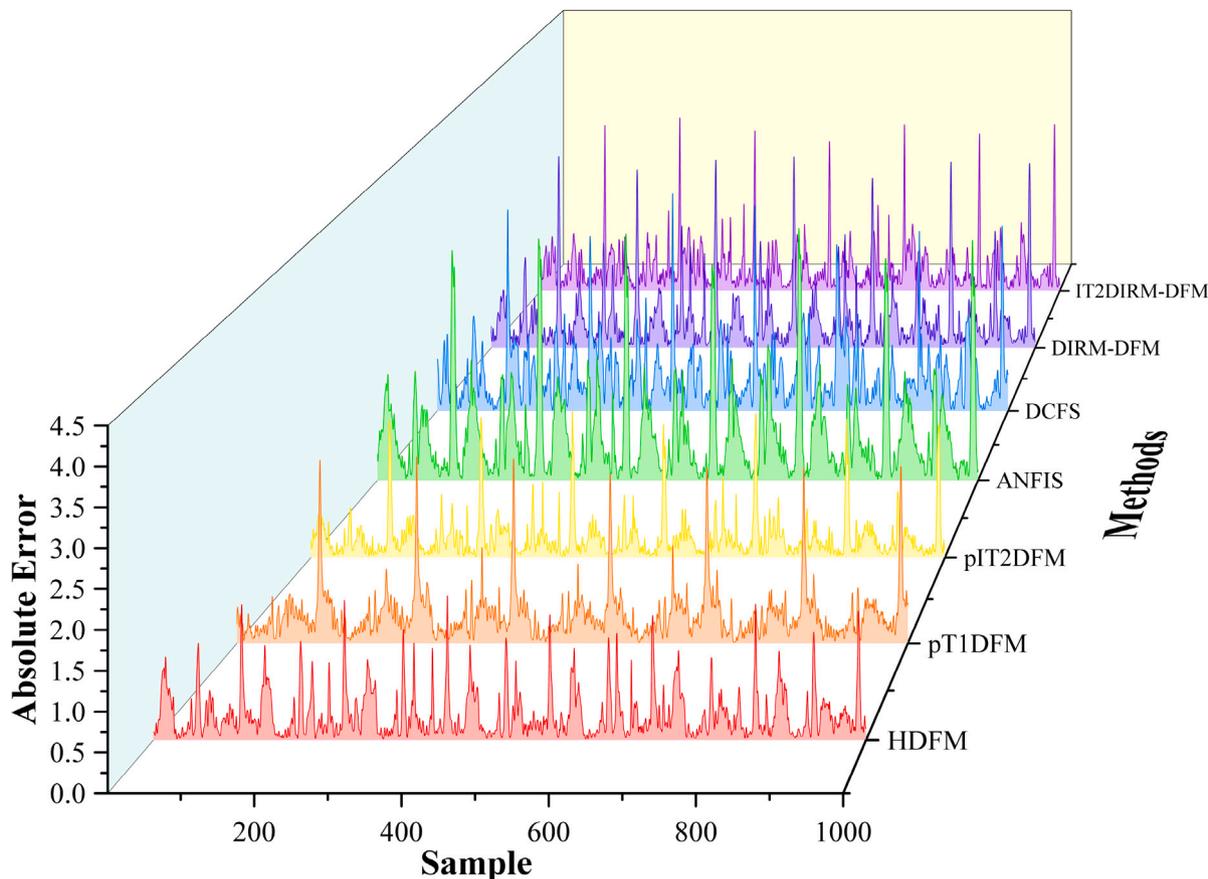


Fig. 17. The absolute errors of all models in experiment 3.

Table 7
The absolute errors of 20 consecutive samples randomly selected in Experiment 3.

Model	HDFM	pT1DFM	pT2DFM	ANFIS	DCFS	DIRM-DFM	IT2DIRM-DFM
1	0.07108728	0.22543513	0.01358658	0.11642098	0.18688408	0.204392146	0.036231505
2	0.16054465	0.30013853	0.08421587	0.02678719	0.31010723	0.033596235	0.141112655
3	0.00269676	0.12137022	0.09342165	0.1562334	0.20411913	0.1358228	0.002195088
4	0.14210437	0.30780727	0.10228551	0.01623085	0.29745328	0.125893297	0.066608076
5	0.17530545	0.29715662	0.08017912	0.04798239	0.38436207	0.051208183	0.192241427
6	0.01688854	0.13597745	0.075423	0.12736449	0.2444917	0.121937201	0.002851593
7	0.1981501	0.3604558	0.15355892	0.08149391	0.36552876	0.056866422	0.115399885
8	0.15484145	0.26187713	0.04233688	0.02421321	0.38436862	0.067842715	0.176738542
9	0.35824261	0.48352896	0.27328167	0.27947076	0.57259402	0.189448673	0.305423498
10	0.31962304	0.38580209	0.15384699	0.21579109	0.58952763	0.314729073	0.297701254
11	0.17205705	0.25323012	0.0232251	0.1452337	0.39218158	0.074155896	0.041285507
12	0.76609142	0.87016818	0.63574389	0.81818061	0.86154571	0.554749805	0.567659156
13	0.50598492	0.53160184	0.25791998	0.60097705	0.82080307	0.825385396	0.56219404
14	0.0581224	0.14295056	0.12290157	0.38917943	0.02343325	0.085885838	0.204872129
15	0.02575412	0.10443271	0.17957438	0.22509162	0.50611291	0.347401807	0.482154067
16	0.08078572	0.21229255	0.06149927	0.19463507	0.47465326	0.289465487	0.22374641
17	0.031471	0.07580262	0.18887325	0.00881688	0.45260133	0.308326599	0.195703465
18	0.09916048	0.02496268	0.22287571	0.36039409	0.45647792	0.487266437	0.312551919
19	0.352691	0.20752173	0.44298912	0.79299079	0.59681312	0.76400142	0.521198441
20	0.90922355	0.67262707	0.87617122	1.37187215	1.08418886	1.455450421	1.019692914

novel hybrid deep fuzzy system structure is presented. In order to accelerate parameter optimization and model convergence, l_2 regularization function is designed, the gradient descent with the Regularization, the DropRule and the AdaBound algorithms is given. Finally, a detailed data-driven model strategy is introduced. In this study, the strategy

used for training parameters of large deep neural networks is applied to the TSK fuzzy system, which realizes effective convergence and accurate optimization of parameter solving. And, this strategy improves the prediction accuracy and generalization performance of the TSK fuzzy system. In order to further increase the complexity of the model

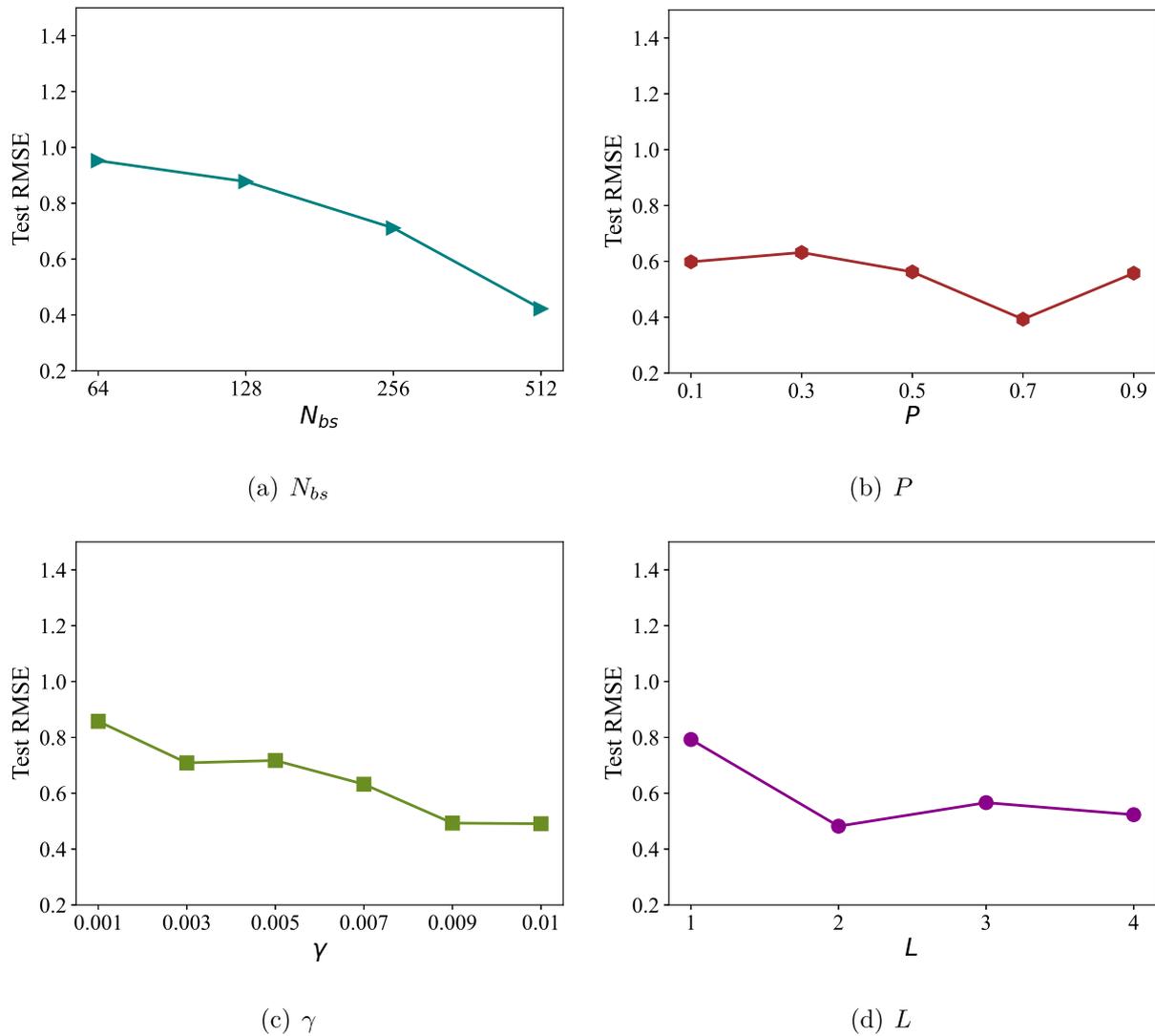


Fig. 18. Test RMSEs of HDFM w.r.t. different hyperparameters in Experiment 3. (a) Different batch size N_{bs} ; (b) different DropRule rate P ; (c) different initial learning rate γ ; (d) different depth L of the stack layer of HDFM. In each subfigure, the configuration of other parameters is consistent with Table 1.

and deal with nonlinear and high randomness problems. This work proposes a novel stack hybrid deep fuzzy model(HDFM), and gives the optimization strategy of the hybrid structure and data driven learning strategy. The proposed HDFM is implemented by stacking TSK fuzzy modules layer by layer from bottom to top. The depth and width of the hybrid deep fuzzy model are adaptive and easy to change. Thanks to the modular design idea, the model effectively avoids the negative impact of rule explosion and has good interpretability. Experimental results on three public datasets show that HDFM, a novel hybrid deep fuzzy model proposed in this study, has better generalization performance and prediction accuracy, which has certain research value.

In future work, on the one hand, we will try to find other advanced optimization algorithms to further improve the performance of the proposed HDFM. On the other hand, other network architecture forms of deep fuzzy systems will also be considered.

CRedit authorship contribution statement

Hui Zhang: Mainly conduct the experiments and simulations, Interpret the results and write the manuscript. **Bo Sun:** Offer help with funding and Supervision. **Wei Peng:** Provide the main idea and methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (61903226, 62076150, 62173216), the Key Research and Development Program of Shandong Province, China (2021CXGC011205), and Shandong Provincial Science, Technology SME Innovation Capability Improving Project, China (2022TSGC2157) and National Innovation and Entrepreneurship Training Program for college students, China (202210430050).

Appendix

See Tables A.1–A.6.

Table A.1
Optimized antecedent parameters in Experiment 1.

	Input 1						Input 2					
	$m_{1,1}$	$\underline{\sigma}_{1,1}$	$\bar{\sigma}_{1,1}$	$m_{1,2}$	$\underline{\sigma}_{1,2}$	$\bar{\sigma}_{1,2}$	$m_{2,1}$	$\underline{\sigma}_{2,1}$	$\bar{\sigma}_{2,1}$	$m_{2,2}$	$\underline{\sigma}_{2,2}$	$\bar{\sigma}_{2,2}$
I-11	-0.01918	1.88628	0.00000	19.72031	1.70600	0.00000	0.16736	1.70600	0.00000	19.89788	2.81046	0.00000
I-12	-0.00270	1.74641	1.29748	19.66017	2.41605	1.76427	0.24163	1.86398	1.07944	19.92352	1.70583	1.02350
I-13	0.24035	1.80917	1.11351	19.94723	2.37406	2.08736	-0.02279	1.88863	1.17341	19.68028	1.70968	1.02334
I-14	0.15481	1.85912	0.00000	19.77210	1.70516	0.00000	0.29346	1.70765	0.00000	20.00650	2.58943	0.00000
I-15	0.21258	1.86690	1.02305	19.96722	1.70508	1.22169	0.21258	1.70508	3.52536	19.96722	2.29741	3.01208
S-11	0.51126	1.11524	0.00000	15.05414	1.39179	0.00000	0.41299	1.39329	0.00000	11.62221	1.15214	0.00000
S-12	0.39022	1.68332	0.00000	11.68017	1.36528	0.00000	0.23647	1.77448	0.00000	10.04979	1.23503	0.00000
S-13	0.42440	1.57890	1.04678	10.21054	1.38225	0.81257	0.27280	1.35736	0.96898	11.62183	1.71526	1.13894
S-14	0.02705	1.49480	0.82866	11.35550	1.48659	0.97293	-0.25516	1.44785	1.06590	12.53386	1.60281	1.06912
S-21	0.43492	1.21793	0.00000	8.99391	1.54099	0.00000	0.26153	1.82943	0.00000	9.33723	1.24137	0.00000
S-22	0.47488	1.77615	1.46456	9.52432	1.43671	0.76708	0.01665	1.27846	1.61212	12.29847	1.85567	0.83758
S-23	0.02448	1.70646	1.39682	12.33188	1.37117	0.86551	0.25190	1.43770	0.89885	11.56388	1.90013	2.14571
S-24	0.19987	1.52681	0.94811	11.55348	1.50663	1.04254	-0.01347	1.90649	0.91735	19.71351	1.67860	1.13131
R-11	0.32496	1.55103	0.00000	9.13945	1.28883	0.00000	0.15488	1.42076	0.00000	12.69629	1.53340	0.00000
R-12	0.04177	1.54596	1.26616	12.58932	1.46908	0.88245	-0.17697	1.34156	1.10682	8.74405	2.28595	1.45140
R-13	0.25678	1.40097	0.94421	9.24761	1.75570	0.98409	0.02548	1.97204	1.15021	14.16117	1.39632	0.83779
R-21	0.50028	1.90267	2.11384	9.25511	1.33586	0.81122	0.07499	1.56693	0.86923	8.30488	1.47845	0.98041
R-22	-0.22999	1.42040	0.83181	8.02144	1.75162	1.04970	0.10964	1.66198	1.07843	11.68972	1.38486	0.97096
O-11	-0.17524	1.54073	1.26293	8.19362	1.84123	1.20714	0.22618	1.97091	1.44704	8.78537	1.37861	1.12532

Table A.2
Optimized consequent parameters in Experiment 1.

	The 1-th rule						The 2-th rule					
	$\underline{\omega}_{1,0}$	$\underline{\omega}_{1,1}$	$\underline{\omega}_{1,2}$	$\bar{\omega}_{1,0}$	$\bar{\omega}_{1,1}$	$\bar{\omega}_{1,2}$	$\underline{\omega}_{2,0}$	$\underline{\omega}_{2,1}$	$\underline{\omega}_{2,2}$	$\bar{\omega}_{2,0}$	$\bar{\omega}_{2,1}$	$\bar{\omega}_{2,2}$
I-11	0.5282	0.0795	0.7175	0.0000	0.0000	0.0000	0.7696	0.4535	0.1621	0.0000	0.0000	0.0000
I-12	0.0894	0.1560	1.5658	0.8279	-0.0535	-0.1245	1.3366	1.2207	0.5754	-0.2570	-0.1852	-0.2267
I-13	0.8859	0.5288	0.4798	-0.1840	-0.2741	1.0612	1.1109	0.7311	0.3081	-0.1237	0.2636	0.2878
I-14	0.2257	0.1598	0.7103	0.0000	0.0000	0.0000	0.3438	0.4137	0.4028	0.0000	0.0000	0.0000
I-15	-0.0572	0.2082	1.1304	0.2179	0.0781	0.3680	-0.1268	0.6221	0.6880	0.4832	0.3379	0.0267
S-11	-0.2322	-0.0898	0.1732	0.0000	0.0000	0.0000	0.5612	0.1802	0.5389	0.0000	0.0000	0.0000
S-12	-0.0522	0.0659	0.9488	0.0000	0.0000	0.0000	0.1268	0.8757	0.2199	0.0000	0.0000	0.0000
S-13	-0.0357	-0.2162	1.0346	-0.2685	0.8210	0.4912	0.7350	-0.4273	0.6448	-0.7078	0.8322	1.0171
S-14	0.4330	-0.0420	0.4455	-0.5846	0.3032	1.5775	0.5106	0.4006	0.5161	0.3247	-0.1991	1.0303
S-21	-0.0764	0.0885	0.9466	0.0000	0.0000	0.0000	-0.0764	0.2965	0.7942	0.0000	0.0000	0.0000
S-22	-0.0701	0.8861	0.1595	-0.1657	-0.1940	1.2994	0.7422	-0.0167	0.9377	0.0556	-0.1333	0.9021
S-23	0.1118	0.1900	0.7872	-0.2548	0.0310	1.1332	0.0750	0.3812	0.5346	-0.0438	0.4472	0.6789
S-24	0.3060	-0.1757	1.0881	-0.0134	0.1958	0.8938	0.1602	1.1880	0.8289	0.1111	-0.0115	-0.3279
R-11	-0.1093	0.1030	0.9317	0.0000	0.0000	0.0000	-0.0171	-0.0034	1.0244	0.0000	0.0000	0.0000
R-12	0.0052	0.4145	0.3132	-0.1117	-0.2572	1.4429	0.0285	0.1886	1.3355	0.1548	0.2648	1.1255
R-13	-0.4919	0.5914	0.6340	0.1032	0.3075	0.7057	0.0238	-0.1831	0.1245	0.3318	0.6255	1.4348
R-21	-0.6795	-0.3636	1.1185	0.3123	0.4625	0.8418	-0.2745	0.1160	0.4531	0.3536	-0.0803	1.5453
R-22	0.4401	0.2917	0.7072	-0.5790	0.2507	0.8317	-0.5594	0.3359	0.7278	0.5192	0.7410	0.1461
O-11	-0.3453	0.1319	1.2270	0.3540	-0.5135	1.1392	0.4071	0.7195	-0.3185	0.2954	0.4283	0.9142
	The 3-th rule						The 4-th rule					
	$\underline{\omega}_{3,0}$	$\underline{\omega}_{3,1}$	$\underline{\omega}_{3,2}$	$\bar{\omega}_{3,0}$	$\bar{\omega}_{3,1}$	$\bar{\omega}_{3,2}$	$\underline{\omega}_{4,0}$	$\underline{\omega}_{4,1}$	$\underline{\omega}_{4,2}$	$\bar{\omega}_{4,0}$	$\bar{\omega}_{4,1}$	$\bar{\omega}_{4,2}$
I-11	0.6088	0.0681	0.5459	0.0000	0.0000	0.0000	0.0673	0.4212	0.6073	0.0000	0.0000	0.0000
I-12	0.7776	0.1859	0.2527	0.2148	-0.0562	1.1814	-0.1845	0.8368	-0.3016	0.4876	0.6188	-0.0730
I-13	0.6300	0.6398	0.6892	-0.0243	-0.5349	0.8104	0.8677	0.1710	1.0365	0.8888	0.1445	-0.2588
I-14	0.3581	0.2249	0.8277	0.0000	0.0000	0.0000	0.5890	0.0983	0.4823	0.0000	0.0000	0.0000
I-15	0.3919	0.5326	1.2257	0.4379	-0.0459	-0.0242	1.0557	0.2122	0.9155	0.2467	0.0984	-0.0136
S-11	0.0249	0.0661	0.9804	0.0000	0.0000	0.0000	0.6345	0.3305	0.3384	0.0000	0.0000	0.0000
S-12	-0.0114	0.3171	0.6489	0.0000	0.0000	0.0000	0.0294	0.2058	0.8255	0.0000	0.0000	0.0000
S-13	0.6633	-0.0963	0.2482	-0.5138	0.2482	1.5247	0.9191	-0.3042	0.5307	-0.3890	0.5075	1.0003
S-14	-0.0687	0.8869	-0.2082	-0.1218	-0.1367	1.5372	0.6900	0.6360	0.4063	-0.4919	0.3720	0.5915
S-21	0.0499	0.5600	0.4376	0.0000	0.0000	0.0000	0.1677	-0.1156	1.1071	0.0000	0.0000	0.0000
S-22	0.2676	-0.0042	0.9041	-0.3503	0.5375	0.6481	-0.1826	0.8338	0.1062	0.6915	-0.0284	0.7580
S-23	0.6904	0.5007	0.1937	0.0851	0.4223	0.4383	0.1177	-0.3347	0.9159	0.4113	0.5852	0.5707
S-24	-0.6632	0.1499	0.5797	0.7930	1.0686	0.1597	-0.2827	0.6879	-0.1506	1.0245	0.7087	0.4464
R-11	0.0306	0.4529	0.5527	0.0000	0.0000	0.0000	0.0825	0.7534	0.1948	0.0000	0.0000	0.0000
R-12	0.6207	-0.0249	1.4598	-0.0339	0.2309	-0.0593	-0.0422	0.8672	0.9101	-0.0714	0.3587	-0.1374
R-13	-0.9030	0.7300	0.5815	0.6601	-0.1598	0.9171	1.0966	0.1059	0.6310	-0.6233	1.4241	-0.3468
R-21	0.8973	-0.0898	0.6998	0.1633	0.3000	0.7017	1.0148	-0.0294	0.4362	-0.0095	0.3263	1.0488
R-22	0.0327	0.6835	1.1012	-0.2469	-0.0378	0.3301	0.3298	0.8401	0.1207	0.2442	0.0891	0.7209
O-11	0.3384	0.7139	0.7823	-0.1973	-0.0666	0.4898	-0.4141	1.3360	0.6105	0.1322	-0.0842	0.2625

Table A.3
Optimized antecedent parameters in Experiment 2.

	Input 1						Input 2					
	$m_{1,1}$	$\underline{\sigma}_{1,1}$	$\bar{\sigma}_{1,1}$	$m_{1,2}$	$\underline{\sigma}_{1,2}$	$\bar{\sigma}_{1,2}$	$m_{2,1}$	$\underline{\sigma}_{2,1}$	$\bar{\sigma}_{2,1}$	$m_{2,2}$	$\underline{\sigma}_{2,2}$	$\bar{\sigma}_{2,2}$
I-11	0.92878	6.21740	4.18601	20.92829	5.01655	3.02985	-0.16959	5.01655	3.07279	19.92097	5.10183	3.48819
I-12	0.72584	6.48793	0.00000	20.54881	5.03409	0.00000	0.56891	6.23408	0.00000	20.42881	5.05494	0.00000
I-13	0.17881	5.14543	3.37668	20.13237	5.17047	3.03421	0.15256	5.09376	3.36436	20.14997	5.02061	3.03604
I-14	0.41913	5.91012	0.00000	20.24460	5.05354	0.00000	0.93345	6.92305	0.00000	20.79555	5.01598	0.00000
I-15	-0.16566	5.01705	3.29659	19.89546	5.35001	3.15482	0.14637	5.01966	3.26007	20.20135	5.01578	3.12444
S-11	-1.47314	4.60185	2.77532	19.01987	4.85240	3.65357	-1.14271	4.96502	3.24153	20.68090	5.02683	2.94780
S-12	-0.60801	6.41894	0.00000	21.25259	4.90243	0.00000	-0.75721	6.00637	0.00000	20.47552	4.80259	0.00000
S-13	-1.34270	4.91388	3.04060	20.04858	5.20383	2.91891	-1.19737	4.85093	2.99164	21.10055	4.85093	3.11411
S-14	-1.49542	5.20848	0.82866	20.65026	5.90624	0.97293	0.18563	5.51543	1.06590	23.46545	5.06991	1.06912
S-21	-0.44531	5.52457	0.00000	21.34928	4.86587	0.00000	-0.65344	5.03791	0.00000	21.12941	5.36487	0.00000
S-22	-0.61536	5.08640	0.00000	21.20777	5.08180	0.00000	-0.60606	5.08397	0.00000	20.71637	5.84396	0.00000
S-23	-0.58767	4.99267	3.09406	20.93077	4.96321	3.24767	-0.44565	4.95889	2.99873	23.75410	5.44410	3.22900
S-24	0.35748	5.86207	4.04942	24.59490	5.00391	3.04304	0.29246	5.36778	3.09818	20.24815	5.00391	3.04954
R-11	-0.91028	4.94046	0.00000	21.20072	5.21019	0.00000	-0.40358	5.10940	0.00000	24.04453	4.97863	0.00000
R-12	-1.00286	4.93584	3.04326	23.59413	5.63655	3.42994	0.04066	5.66102	3.65279	21.80994	4.93670	2.96331
R-13	0.00949	5.83229	0.00000	21.71007	4.99733	0.00000	0.57881	6.68169	0.00000	20.38438	4.97452	0.00000
R-21	1.16292	8.64659	0.00000	23.65536	5.10991	0.00000	0.15617	6.58971	0.00000	22.08906	4.94629	0.00000
R-22	-0.47773	5.49958	0.00000	21.53236	5.66371	0.00000	-0.22386	5.10796	0.00000	20.56074	5.57747	0.00000
O-11	-0.37018	5.06681	0.00000	28.21539	5.10468	0.00000	0.06195	5.39148	0.00000	20.78743	4.94124	0.00000

Table A.4
Optimized consequent parameters in Experiment 2.

	The 1-th rule						The 2-th rule					
	$\underline{\omega}_{1,0}$	$\underline{\omega}_{1,1}$	$\underline{\omega}_{1,2}$	$\bar{\omega}_{1,0}$	$\bar{\omega}_{1,1}$	$\bar{\omega}_{1,2}$	$\underline{\omega}_{2,0}$	$\underline{\omega}_{2,1}$	$\underline{\omega}_{2,2}$	$\bar{\omega}_{2,0}$	$\bar{\omega}_{2,1}$	$\bar{\omega}_{2,2}$
I-11	0.4188	-1.2300	1.7935	0.3415	-1.9618	2.9305	0.0653	0.0041	0.8902	0.5475	-1.7428	2.5070
I-12	0.2125	-1.7485	2.7555	0.0000	0.0000	0.0000	0.3200	-1.1637	2.0523	0.0000	0.0000	0.0000
I-13	-0.6674	0.1480	0.1740	0.4648	-1.5152	2.7928	-0.1153	-1.0508	0.5753	0.3484	-0.0482	1.5017
I-14	0.0917	-0.9027	2.0185	0.0000	0.0000	0.0000	0.0967	-0.0770	0.9920	0.0000	0.0000	0.0000
I-15	0.1267	-0.4731	0.9094	-0.0101	0.0007	1.2614	0.1206	-0.4771	0.4320	-0.0356	0.4714	0.9239
S-11	-0.4691	-0.3671	0.9738	0.1389	0.2680	0.8941	0.2277	-0.5484	0.8860	-0.2421	0.6112	0.6018
S-12	-0.0256	0.2244	0.7728	0.0000	0.0000	0.0000	-0.0417	0.2098	0.7781	0.0000	0.0000	0.0000
S-13	-0.4715	0.2557	-0.0428	0.1422	0.0646	1.4245	0.3009	-0.4223	0.1014	-0.0816	0.5737	1.0648
S-14	-0.0008	-0.0403	1.0146	0.0000	0.0000	0.0000	0.0017	-0.0175	0.9939	0.0000	0.0000	0.0000
S-21	-0.0065	-0.0881	1.0378	0.0000	0.0000	0.0000	0.0237	0.1153	0.9288	0.0000	0.0000	0.0000
S-22	0.0088	-0.1568	1.0699	0.0000	0.0000	0.0000	-0.0094	-0.0501	1.0064	0.0000	0.0000	0.0000
S-23	-0.2350	0.5013	0.0728	0.1055	-0.3426	1.5307	-0.3499	-0.0033	0.7946	0.0621	0.4500	0.5920
S-24	-0.1780	0.1016	0.6611	0.0840	-0.0035	1.1233	0.5839	-0.2389	0.7340	-0.2194	0.4550	0.7424
R-11	-0.0371	-0.0251	0.9723	0.0000	0.0000	0.0000	0.0110	0.2202	0.8039	0.0000	0.0000	0.0000
R-12	-0.5576	-0.7354	0.7660	0.2658	0.2145	1.1715	-0.3889	0.8165	-0.2643	0.1556	-0.2097	1.4047
R-13	0.0156	0.1814	0.7899	0.0000	0.0000	0.0000	-0.0039	0.8591	0.0954	0.0000	0.0000	0.0000
R-21	0.0160	-0.0729	1.0980	0.0000	0.0000	0.0000	0.0266	0.0179	0.9856	0.0000	0.0000	0.0000
R-22	-0.0032	0.4403	0.5642	0.0000	0.0000	0.0000	0.0185	0.0310	1.0561	0.0000	0.0000	0.0000
O-11	-0.0102	0.3271	0.6483	0.0000	0.0000	0.0000	-0.0062	0.1387	0.8114	0.0000	0.0000	0.0000
	The 3-th rule						The 4-th rule					
	$\underline{\omega}_{3,0}$	$\underline{\omega}_{3,1}$	$\underline{\omega}_{3,2}$	$\bar{\omega}_{3,0}$	$\bar{\omega}_{3,1}$	$\bar{\omega}_{3,2}$	$\underline{\omega}_{4,0}$	$\underline{\omega}_{4,1}$	$\underline{\omega}_{4,2}$	$\bar{\omega}_{4,0}$	$\bar{\omega}_{4,1}$	$\bar{\omega}_{4,2}$
I-11	-0.8051	-0.1858	0.1244	0.8658	-0.4814	1.6465	-0.1663	0.1859	0.8732	0.6090	-0.7418	1.5525
I-12	0.2312	-1.0566	1.9838	0.0000	0.0000	0.0000	0.2386	-0.3581	1.3147	0.0000	0.0000	0.0000
I-13	0.4893	-0.6837	0.9134	-0.1584	0.1138	1.0497	-0.3455	-0.1434	0.7292	-0.0123	-0.0255	1.0986
I-14	0.0557	-0.3538	1.3723	0.0000	0.0000	0.0000	0.1800	0.0113	0.8803	0.0000	0.0000	0.0000
I-15	0.6362	-0.0598	0.5047	-0.2490	0.0050	1.2342	-0.0046	0.4513	0.4119	0.0436	-0.2282	1.3292
S-11	-0.0960	0.0844	0.2951	-0.0957	0.2951	0.9332	0.5665	-0.5296	0.8009	-0.3709	0.7500	0.6151
S-12	-0.0148	0.1308	0.9074	0.0000	0.0000	0.0000	-0.0938	0.1874	0.8528	0.0000	0.0000	0.0000
S-13	0.0337	-0.4477	0.4628	-0.0813	0.5141	0.9311	-0.5036	0.4317	-0.3327	0.2592	0.2548	1.0847
S-14	0.0078	-0.0105	1.0272	0.0000	0.0000	0.0000	0.0232	0.1774	0.8063	0.0000	0.0000	0.0000
S-21	-0.0022	0.1656	0.9311	0.0000	0.0000	0.0000	0.1949	0.1747	0.7544	0.0000	0.0000	0.0000
S-22	-0.0276	0.1710	0.6442	0.0000	0.0000	0.0000	0.0000	0.2011	0.8476	0.0000	0.0000	0.0000
S-23	0.4787	1.2614	-0.6127	-0.1637	-0.3288	1.4975	0.7858	0.3026	0.2663	-0.3880	0.1480	0.9585
S-24	-0.5670	0.0196	0.6002	0.2198	0.0529	1.0689	1.1855	0.2938	-0.0179	-0.6510	0.2063	1.0426
R-11	-0.0357	0.1694	0.7461	0.0000	0.0000	0.0000	-0.1042	0.4598	0.5699	0.0000	0.0000	0.0000
R-12	0.1641	-0.2642	0.4429	-0.0371	0.2735	1.0529	-0.4007	0.2521	0.2473	-0.0863	0.2115	1.0730
R-13	0.0279	0.1857	0.7781	0.0000	0.0000	0.0000	-0.0105	0.1896	0.7998	0.0000	0.0000	0.0000
R-21	0.0078	-0.0157	0.9587	0.0000	0.0000	0.0000	0.0348	0.0960	0.9215	0.0000	0.0000	0.0000
R-22	0.0181	0.7473	0.2675	0.0000	0.0000	0.0000	0.1845	0.2000	0.7777	0.0000	0.0000	0.0000
O-11	-0.0137	0.0597	0.8798	0.0000	0.0000	0.0000	0.0952	0.3396	0.7151	0.0000	0.0000	0.0000

Table A.5
Optimized antecedent parameters in Experiment 3.

	Input 1						Input 2					
	$m_{1,1}$	$\underline{c}_{1,1}$	$\bar{c}_{1,1}$	$m_{1,2}$	$\underline{c}_{1,2}$	$\bar{c}_{1,2}$	$m_{2,1}$	$\underline{c}_{2,1}$	$\bar{c}_{2,1}$	$m_{2,2}$	$\underline{c}_{2,2}$	$\bar{c}_{2,2}$
I-11	1.40288	2.11919	1.37366	10.23101	2.29749	1.96052	2.45031	2.63442	2.48064	10.74982	2.34651	1.27151
I-12	2.06049	2.44210	0.00000	10.43324	2.50311	0.00000	2.99464	4.34986	0.00000	11.10245	2.32252	0.00000
I-13	2.28725	2.80295	0.00000	10.49115	2.58045	0.00000	2.32071	2.96513	0.00000	10.44269	2.45734	0.00000
I-14	2.24582	2.78630	0.00000	10.48038	2.72759	0.00000	1.73754	2.12342	0.00000	10.27022	2.72559	0.00000
I-15	0.75804	2.16184	0.00000	9.18770	5.00243	0.00000	2.84288	4.24776	0.00000	10.44990	3.14525	0.00000
S-11	2.33434	2.69418	2.03542	10.85632	1.97127	1.18276	2.03509	2.15295	1.28825	10.68302	2.37737	1.57424
S-12	2.32970	2.65024	0.00000	10.94343	2.13719	0.00000	1.98549	2.05357	0.00000	10.49317	2.45491	0.00000
S-13	2.71472	3.30341	1.71176	11.14851	2.28409	1.22903	1.84138	2.37097	1.23358	10.40742	2.94507	1.59170
S-14	1.72669	2.07811	1.24687	10.33173	2.74457	2.09341	2.59511	2.83659	2.38348	11.14982	2.07811	1.27251
S-21	0.79071	2.04073	1.23979	10.23140	3.38424	2.92160	3.09375	3.62263	3.11712	11.77367	2.02848	1.21890
S-22	1.68981	2.05393	0.00000	10.53252	2.79766	0.00000	2.68235	3.94902	0.00000	11.59900	2.05393	0.00000
S-23	2.54570	3.00809	2.93099	11.33455	2.10585	1.28767	1.24866	2.13318	1.26351	9.90804	3.15347	2.82235
S-24	3.59383	5.39714	0.00000	12.11748	2.12096	0.00000	1.68009	2.38020	0.00000	10.11705	3.28381	0.00000
R-11	1.38494	2.04278	1.28530	10.33411	3.13049	2.45226	2.18242	2.33984	1.86801	11.04051	2.42411	1.23944
R-12	0.98128	2.09974	0.00000	10.04847	4.24432	0.00000	2.85689	4.09696	0.00000	11.11380	2.46010	0.00000
R-13	2.90289	4.02272	0.00000	11.30268	2.10762	0.00000	1.35546	2.17592	0.00000	10.12070	3.86557	0.00000
R-21	2.11246	2.22068	0.00000	10.80650	2.08022	0.00000	2.25545	2.40719	0.00000	10.89074	2.11495	0.00000
R-22	2.22004	2.35993	0.00000	10.54374	2.42113	0.00000	2.03560	2.27649	0.00000	10.86756	2.47930	0.00000
O-11	3.17693	3.66085	3.35110	11.72851	2.09259	1.25744	1.27847	2.11830	1.25559	10.09660	2.96365	2.69963

Table A.6
Optimized consequent parameters in Experiment 3.

	The 1-th rule						The 2-th rule					
	$\underline{\omega}_{1,0}$	$\underline{\omega}_{1,1}$	$\underline{\omega}_{1,2}$	$\bar{\omega}_{1,0}$	$\bar{\omega}_{1,1}$	$\bar{\omega}_{1,2}$	$\underline{\omega}_{2,0}$	$\underline{\omega}_{2,1}$	$\underline{\omega}_{2,2}$	$\bar{\omega}_{2,0}$	$\bar{\omega}_{2,1}$	$\bar{\omega}_{2,2}$
I-11	-0.3362	0.8430	0.5677	0.7582	-0.7151	1.4417	0.4326	0.8102	0.8157	0.7699	-0.1060	0.7009
I-12	0.3734	-0.7757	1.7092	0.0000	0.0000	0.0000	0.5833	-0.6824	1.6047	0.0000	0.0000	0.0000
I-13	0.1791	-0.3046	1.2684	0.0000	0.0000	0.0000	0.1731	-0.2594	1.2185	0.0000	0.0000	0.0000
I-14	0.1397	0.3163	0.6673	0.0000	0.0000	0.0000	0.1030	-0.3066	1.2864	0.0000	0.0000	0.0000
I-15	0.0976	0.0243	0.9509	0.0000	0.0000	0.0000	0.0396	0.3778	0.6198	0.0000	0.0000	0.0000
S-11	-0.4359	-0.8202	1.1489	0.1389	0.0013	0.0812	0.5852	0.1785	0.1555	-0.2546	-0.3503	1.6939
S-12	-0.0378	-0.0007	0.9389	0.0000	0.0000	0.0000	-0.0575	-0.2152	1.2091	0.0000	0.0000	0.0000
S-13	0.3783	-1.0497	1.0033	-0.3777	0.6097	0.8875	-0.0924	0.1629	-0.1047	0.2018	-0.2428	1.6146
S-14	0.5978	-0.1017	0.2930	-0.3369	0.3432	1.0099	1.0462	0.6359	-0.3848	-0.4948	-0.0975	1.4166
S-21	0.3496	-0.3529	0.8316	-0.0620	0.3303	0.9310	0.8587	-0.4337	0.9306	0.2145	0.4892	0.6410
S-22	0.0032	-0.0468	1.0596	0.0000	0.0000	0.0000	0.1444	0.3081	0.6595	0.0000	0.0000	0.0000
S-23	0.5952	-0.4361	0.9246	-0.1324	0.2907	0.9136	0.2558	-0.2874	0.9772	0.0347	0.0903	1.0308
S-24	-0.0189	0.4075	0.5927	0.0000	0.0000	0.0000	-0.0442	1.0516	-0.0280	0.0000	0.0000	0.0000
R-11	0.8381	0.7548	0.1959	-0.1829	-0.0434	0.9893	-0.4808	0.8832	0.4694	0.0023	0.0068	0.8700
R-12	-0.1484	-0.2510	1.2707	0.0000	0.0000	0.0000	0.3195	0.5060	0.4596	0.0000	0.0000	0.0000
R-13	-0.0897	0.7744	0.2370	0.0000	0.0000	0.0000	-0.0003	-0.0834	1.0955	0.0000	0.0000	0.0000
R-21	0.0387	-0.1391	1.1173	0.0000	0.0000	0.0000	0.1439	0.1888	0.7926	0.0000	0.0000	0.0000
R-22	-0.0559	0.8655	0.1343	0.0000	0.0000	0.0000	0.0440	0.6854	0.3247	0.0000	0.0000	0.0000
O-11	-0.6285	0.1557	0.4538	0.1286	0.6718	0.5289	0.0465	0.6464	-0.1267	0.0536	0.4990	0.7096
	The 3-th rule						The 4-th rule					
	$\underline{\omega}_{3,0}$	$\underline{\omega}_{3,1}$	$\underline{\omega}_{3,2}$	$\bar{\omega}_{3,0}$	$\bar{\omega}_{3,1}$	$\bar{\omega}_{3,2}$	$\underline{\omega}_{4,0}$	$\underline{\omega}_{4,1}$	$\underline{\omega}_{4,2}$	$\bar{\omega}_{4,0}$	$\bar{\omega}_{4,1}$	$\bar{\omega}_{4,2}$
I-11	-0.3703	0.4522	0.5892	0.2330	-0.7226	1.7101	0.2795	0.1408	0.5590	0.1451	-0.3391	1.3613
I-12	0.3783	-1.0040	1.9184	0.0000	0.0000	0.0000	0.0697	-0.5288	1.4609	0.0000	0.0000	0.0000
I-13	0.3046	-0.2632	1.2445	0.0000	0.0000	0.0000	0.1462	-0.5964	1.5439	0.0000	0.0000	0.0000
I-14	0.1586	-0.5443	1.5209	0.0000	0.0000	0.0000	0.3662	-0.8518	1.7788	0.0000	0.0000	0.0000
I-15	0.0112	-0.2840	1.2739	0.0000	0.0000	0.0000	0.1710	-0.0207	1.0023	0.0000	0.0000	0.0000
S-11	-0.2429	-0.1904	0.4335	0.1653	0.4224	0.9043	0.0836	-0.0899	0.3799	0.1601	0.3779	0.8928
S-12	0.1498	0.0221	0.9407	0.0000	0.0000	0.0000	0.0450	0.0519	0.9509	0.0000	0.0000	0.0000
S-13	-0.5643	0.2105	-0.2629	-0.2033	0.2977	1.2450	-0.4033	-0.1120	0.3524	-0.0444	0.6113	0.7474
S-14	-0.0090	-0.3952	0.6932	0.1722	0.0508	1.2248	1.0591	0.2904	-0.1630	-0.6496	0.0213	1.3830
S-21	-0.6179	0.5105	0.2505	0.2133	0.0669	1.0601	-0.8211	0.7785	0.0074	0.1882	0.1319	0.9656
S-22	0.0401	-0.2328	1.1797	0.0000	0.0000	0.0000	0.0527	0.0508	0.9374	0.0000	0.0000	0.0000
S-23	0.2315	0.2765	0.1430	-0.1035	0.5457	0.7140	0.6552	0.0233	0.6394	0.3152	0.3150	0.7516
S-24	-0.1831	0.5762	0.4450	0.0000	0.0000	0.0000	0.3089	0.6765	0.2870	0.0000	0.0000	0.0000
R-11	-0.1363	0.4252	0.6930	0.0356	-0.0785	1.0594	0.4930	-0.1093	1.0944	0.2850	-0.0095	0.9829
R-12	-0.0947	-0.0403	1.0565	0.0000	0.0000	0.0000	0.0815	-0.1333	1.1354	0.0000	0.0000	0.0000
R-13	0.2591	0.1188	0.8558	0.0000	0.0000	0.0000	0.0556	0.2050	0.7859	0.0000	0.0000	0.0000
R-21	-0.1923	0.3449	0.6752	0.0000	0.0000	0.0000	0.2183	0.0782	0.9253	0.0000	0.0000	0.0000
R-22	-0.1824	0.7610	0.2829	0.0000	0.0000	0.0000	0.1438	0.1835	0.7978	0.0000	0.0000	0.0000
O-11	0.6815	0.3004	0.1424	0.1683	0.5235	0.6362	-0.7800	0.6315	0.0387	0.0720	0.1635	1.0062

References

- Begian, M. B., Melek, W. W., & Mendel, J. M. (2008). Stability analysis of type-2 fuzzy systems. In *2008 IEEE international conference on fuzzy systems* (pp. 947–953). IEEE.
- Bilgili, M., Ilhan, A., & Ünal, Ş. (2022). Time-series prediction of hourly atmospheric pressure using ANFIS and LSTM approaches. *Neural Computing and Applications*, *34*(18), 15633–15648.
- Cervone, G., Clemente-Harding, L., Alessandrini, S., & Delle Monache, L. (2017). Short-term photovoltaic power forecasting using Artificial Neural Networks and an Analog Ensemble. *Renewable Energy*, *108*, 274–286.
- Cheng, Y., Wan, S., Choo, K.-K. R., et al. (2018). Deep belief network for meteorological time series prediction in the Internet of Things. *IEEE Internet of Things Journal*, *6*(3), 4369–4376.
- Chu, Y., Urquhart, B., Gohari, S. M., Pedro, H. T., Kleissl, J., & Coimbra, C. F. (2015). Short-term reforecasting of power output from a 48 MWe solar PV plant. *Solar Energy*, *112*, 68–77.
- Coupland, S., & John, R. (2008). Type-2 fuzzy logic and the modelling of uncertainty. In H. Bustince, F. Herrera, & J. Montero (Eds.), *Fuzzy sets and their extensions: Representation, aggregation and models* (pp. 3–22). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hossain, M., Mekhilef, S., Danesh, M., Olatomiwa, L., & Shamshirband, S. (2017). Application of extreme learning machine for short term output power forecasting of three grid-connected PV systems. *Journal of Cleaner Production*, *167*, 395–405.
- Karevan, Z., & Suykens, J. A. (2020). Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, *125*, 1–9.
- Li, C., Gao, J., Yi, J., & Zhang, G. (2016). Analysis and design of functionally weighted single-input-rule-modules connected fuzzy inference systems. *IEEE Transactions on Fuzzy Systems*, *26*(1), 56–71.
- Li, Y., Su, Y., & Shu, L. (2014). An ARMAX model for forecasting the power output of a grid connected photovoltaic system. *Renewable Energy*, *66*, 78–89.
- Li, L.-L., Wen, S.-Y., Tseng, M.-L., & Wang, C.-S. (2019). Renewable energy prediction: A novel short-term prediction model of photovoltaic output power. *Journal of Cleaner Production*, *228*, 359–375.
- Li, C., & Yi, J. (2010). SIRMs based interval type-2 fuzzy inference systems: properties and application. *International Journal of Innovative Computing, Information and Control*, *6*(9), 4019–4028.
- Li, C., Zhou, C., Peng, W., Lv, Y., & Luo, X. (2020). Accurate prediction of short-term photovoltaic power generation via a novel double-input-rule-modules stacked deep fuzzy method. *Energy*, *212*, Article 118700.
- Liu, Y., Gong, C., Yang, L., & Chen, Y. (2020). DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with Applications*, *143*, Article 113082.
- Liu, P., Liu, J., & Wu, K. (2020). CNN-FCM: System modeling promotes stability of deep learning in time series prediction. *Knowledge-Based Systems*, *203*, Article 106081.
- Luo, L., Xiong, Y., Liu, Y., & Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. In *International conference on learning representations*.
- Mendel, J. M. (2017). Uncertain rule-based fuzzy systems. In *Introduction and new directions*, Vol. 684. Springer.
- Nguyen, A.-T., Taniguchi, T., Eciolaza, L., Campos, V., Palhares, R., & Sugeno, M. (2019). Fuzzy control systems: Past, present and future. *IEEE Computational Intelligence Magazine*, *14*(1), 56–68.
- Peng, W., Zhou, C., Li, C., Deng, X., & Zhang, G. (2021). Double-input rule modules stacked deep interval type-2 fuzzy model with application to time series forecasting. *International Journal of Fuzzy Systems*, *23*(5), 1326–1346.
- Seki, H., Ishii, H., & Mizumoto, M. (2008). On the generalization of single input rule modules connected type fuzzy reasoning method. *IEEE Transactions on Fuzzy Systems*, *16*(5), 1180–1187.
- Singh, P. (2020). A novel hybrid time series forecasting model based on neutrosophic-PSO approach. *International Journal of Machine Learning and Cybernetics*, *11*(8), 1643–1658.
- Singh, P. (2021). FQTSFM: A fuzzy-quantum time series forecasting model. *Information Sciences*, *566*, 57–79.
- Singh, P., & Huang, Y.-P. (2019). A high-order neutrosophic-neuro-gradient descent algorithm-based expert system for time series forecasting. *International Journal of Fuzzy Systems*, *21*, 2245–2257.
- Singh, R., & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications*, *76*, 18569–18584.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.
- Tak, N., Evren, A. A., Tez, M., & Egrioglu, E. (2018). Recurrent type-1 fuzzy functions approach for time series forecasting. *Applied Intelligence*, *48*, 68–77.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(12).
- Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International conference on machine learning* (pp. 1058–1066). PMLR.
- Wang, L.-X. (2019). Fast training algorithms for deep convolutional fuzzy systems with application to stock index prediction. *IEEE Transactions on Fuzzy Systems*, *28*(7), 1301–1314.
- Wu, D., Yuan, Y., Huang, J., & Tan, Y. (2019). Optimize TSK fuzzy systems for regression problems: Minibatch gradient descent with regularization, DropRule, and AdaBound (MBGD-RDA). *IEEE Transactions on Fuzzy Systems*, *28*(5), 1003–1015.
- Yi, J., Yubazaki, N., & Hirota, K. (2001). Upswing and stabilization control of inverted pendulum system based on the SIRMs dynamically connected fuzzy inference model. *Fuzzy Sets and Systems*, *122*(1), 139–152.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353.
- Zadeh, L. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, *8*(3), 199–249.
- Zhang, H., & Peng, W. (2021). Time series forecasting based on interval type-2 fuzzy logic system with PSO. In *2021 China automation congress* (pp. 3090–3097). IEEE.
- Zhou, Y., Zhou, N., Gong, L., & Jiang, M. (2020). Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine. *Energy*, *204*, Article 117894.