Journal of
Journal of Artificial Intelligence and Soft Computing Techniques

Research Article Open Access

Interval Type-3 Fuzzy Broad Learning System with Its Application in Uncertain System Regression Modeling

Hao Tian[1,2], Jian Tang[1,2*] and Luhong Guo[3]

[1]School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China
[2]Beijing Laboratory of Smart Environmental Protection, Beijing 100124, China
[3]Beijing-Dublin International College at BJUT, Beijing University of Technology, Beijing 100124, China

[*]**Corresponding Author:** Jian Tang, School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China, E-mail: freeflytang@bjut.edu.cn

Received Date: June 20, 2025 **Accepted Date:** July 04, 2025 **Published Date:** July 07, 2025

Citation: Hao Tian, Jian Tang, Luhong Guo (2025) Interval Type-3 Fuzzy Broad Learning System with Its Application in Uncertain System Regression Modeling.J Artif Intel Sost Comp Tech 2: 1-31

Abstract

Traditional Type-1 and Type-2 fuzzy systems face limitations in addressing complex uncertainties. To overcome these challenges, a novel framework integrating the Interval Type-3 Fuzzy System (IT3FS) with a Broad Learning System (BLS) was developed, referred to as the Interval Type-3 Fuzzy Broad Learning System (IT3FBLS). This approach combines the advanced uncertainty representation capabilities of Type-3 fuzzy logic with the computational efficiency of BLS, effectively addressing the modeling difficulties associated with complex, nonlinear, and uncertain data. In the IT3FBLS, an entropy-based adaptive regularization and gradient quadratic optimization strategy is introduced for parameter learning. This strategy dynamically adjusts node distribution and feature contribution, thereby improving robustness and generalization in the context of high-dimensional, multi-source heterogeneous data. Modeling experiments were conducted using low-to-medium-dimensional public datasets, uncertain functions, and real-world industrial process data. The results, when compared to several conventional and advanced BLS and fuzzy models,

Keywords: Interval Type-3 Fuzzy System (IT3FS); Broad Learning System (BLS); Entropy-Based Parameter Learning; Interval Type-3 Fuzzy BLS (IT3FBLS); Industrial Process Modeling

Introduction

With the rapid development of sensor networks and information technology, industrial production processes are generating vast amounts of multi-source, heterogeneous data [1]. These data often exhibit prominent non-Gaussian characteristics and dynamic coupling effects [2], while also being influenced by various stochastic factors, resulting in significant uncertainty in both data distribution and system behavior [3]. Traditional modeling approaches typically rely on prior assumptions about system mechanisms [4], making them inadequate for capturing the highly nonlinear and multi-layered uncertainties inherent in industrial processes. At the same time, the increasing demand for real-time performance and accuracy has highlighted a critical challenge: how to develop regression models that can operate efficiently and reliably in high-dimensional, dynamic, and noisy data environments. This challenge has increasingly become a bottleneck in the intelligent transformation of industrial processes [5]. Therefore, the development of more flexible, efficient, and adaptive data-driven modeling theories and methods, capable of handling complex and uncertain datasets, has become a key focus of both academic research and industrial applications.

Amid the continuous advancement of industrial big data and artificial intelligence (AI) technologies [6], data-driven modeling strategies are emerging as a powerful solution to the challenges of complex industrial process modeling. Among these, fuzzy logic systems (FLS) [7] have gained widespread adoption in industrial research and optimization due to their ability to handle nonlinear problems without requiring explicit mathematical models. Building on this foundation, fuzzy neural networks (FNN) [8] combine the fuzzy inference and uncertainty-handling capabilities of FLS with the adaptive learning potential of neural networks, enabling automatic parameter adjustment. Applications of FNN in areas such as municipal solid waste incineration (MSWI) and wastewater treatment processes (WWTP) have demonstrated promising results [9,10], highlighting its strong potential for improving process modeling and control.

However, the widely adopted early-generation Type-1 Fuzzy Systems (T1FS) exhibit inherent limitations in handling uncertainty, as their membership functions can only depict single-layer fuzzy boundaries, rendering them ineffective in complex environments characterized by severe noise and disturbances [11]. To address these shortcomings, researchers introduced interval type-2 fuzzy systems (IT2FS), which can incorporate uncertainty boundaries, thereby significantly expanding the applicability of fuzzy logic in complex and uncertain environments [12]. Leveraging the enhanced uncertainty modeling capabilities of IT2FS, interval type-2 fuzzy neural networks (IT2FNN) have found widespread applications in industrial process control [13].

Despite the significant advancements brought by IT2FS in extending fuzzy boundaries, their secondary membership functions are typically confined to interval forms, limiting their capacity to capture multi-level, unstable, or time-varying uncertainties. This limitation becomes particularly pronounced when dealing with non-Gaussian noise and multi-modal fuzzy distributions, leading to potential modeling biases [14]. Consequently, improving the adaptability of fuzzy systems to handle multi-level uncertainties without significantly increasing system complexity has become a critical research direction.

To further enhance the modeling capabilities of fuzzy systems, general type-2 fuzzy systems (GT2FS) were proposed, extending secondary membership functions into Type-1 fuzzy sets, thereby theoretically enabling a more precise characterization of high-dimensional uncertainties [15,16]. However, the complex structure and substantial computational overhead associated with GT2FS have hindered their practical deployment in industrial scenarios requiring real-time processing [17]. As an improvement, interval type-3 fuzzy systems (IT3FS), based on type-3 FLS (T3FLS), were introduced [18]. These systems utilize interval type-2 fuzzy sets to construct secondary membership functions, effectively balancing the enhanced uncertainty-handling capacity of type-3 fuzzy systems with reduced computational complexity. IT3FS has already demonstrated promising potential in nonlinear dynamic system control and predictive modeling [19,20,21].

Nevertheless, while IT3FS represents a significant advancement in uncertainty management, its complex system architecture also introduces an expansive parameter space. In practical applications, particularly within adaptive control domains, fuzzy models should satisfy real-time online learning and dynamic updates. The substantial parameter tuning and high-dimensional search requirements of IT3FS not only complicate algorithmic convergence but also impose considerable computational burdens, thereby constraining its feasibility for real-time industrial applications [22]. There-

fore, designing more efficient parameter learning algorithms to unlock the full modeling potential of IT3FS, while reducing computational costs, remains an open issue to be solved.

To address the challenges associated with parameter learning in IT3FS, the recently popular broad learning system (BLS) [23] offers a promising solution. BLS focuses on rapid learning through feature mapping and an enhanced node structure. Unlike conventional deep learning frameworks, BLS eliminates the need for multi-layer iterative training, requiring only output layer weight updates to accomplish large-scale data learning, thus significantly reducing computational complexity [24,25]. Moreover, its parallel computing and incremental learning mechanisms enable more efficient and feasible online updates in dynamic environments. By integrating the advanced uncertainty representation of IT3FS with the efficient learning mechanism of BLS, it becomes feasible to develop a fuzzy system characterized by strong adaptability and rapid convergence under complex uncertain conditions.

In training regression models for complex datasets, conventional performance evaluation metrics, such as the mean and variance, often fail to fully capture the intricate characteristics of non-Gaussian or multi-modal dynamic systems. Entropy, as a critical measure of data diversity, uncertainty, and randomness [26], offers a more effective means of revealing higher-order statistical information. In modeling error analysis, lower entropy typically indicates a more concentrated error distribution with reduced volatility, reflecting greater system stability [27]. Therefore, to optimize the parameters of IT3FBLS more precisely, leveraging entropy-based metrics for weight adjustment may enhance the system's adaptability and robustness in uncertain environments.

In summary, to address the challenges of modeling and controlling complex uncertain datasets, this article proposes a novel neural model framework — the interval type-3 fuzzy broad learning system (IT3FBLS) — which integrates IT3FS with BLS. Within this framework, the IT3FS module replaces the feature mapping unit of BLS, retaining the structural simplicity and efficient learning capabilities of BLS while overcoming the high-dimensional parameter learning bottleneck of IT3FS through multi-level uncertainty modeling and rapid learning mechanisms. Additionally, two entropy-based parameter optimization strategies are introduced, leveraging entropy metrics for model evaluation and adjustment to significantly reduce modeling errors. The effectiveness of the proposed algorithm is ultimately validated through its application in uncertain data regression modeling with different datasets. The primary contributions of this article are as follows:

(1) To address the challenges of complex uncertainty and real-time regression modeling, an IT3FBLS framework integrating IT3FS and the BLS is proposed. This framework achieves precise regression and adaptive control of complex dynamic systems with high uncertainty representation capability while requiring only limited training samples.

(2) By retaining the efficient feature mapping and enhanced node structure of BLS, the IT3FBLS framework further improves modeling performance for high-dimensional and multi-source heterogeneous data. This article also provides an initial exploration of integrating advanced fuzzy logic systems within the BLS framework, offering insights for future advancements in uncertainty modeling.

(3) Two entropy-weighted secondary parameter optimization algorithms are designed, including an adaptive regularization strategy and a gradient descent-based secondary optimization approach. These algorithms effectively mitigate the impact of random noise and non-stationary disturbances while ensuring dynamic fine-tuning and effective feature extraction, laying the foundation for high-precision regression modeling and robust control.

The remainder of this article is organized as follows: Section 2 introduces the preliminaries, including IT3FS and BLS. Section 3 provides details on the IT3FBLS structure and implementation, entropy-based parameter learning, computational complexity analysis, and algorithm pseudocode. Section 4 presents experiments and discussions, while Section 5 summarizes the research.

2. Preliminaries

2.1 Interval Type-3 Fuzzy System (IT3FS)

The IT3FS is a fuzzy logic system designed to handle complex uncertainties. By employing IT2FS to construct secondary membership function (SMF), IT3FS enhances its ability to model uncertainties without significantly increasing computational complexity. The computational process of IT3FS can be divided into five functional layers: input layer, membership function computation layer, rule inference layer, type-reduction layer, and output layer. The description of each layer is provided in detail as follows.

(1) Input layer: The input layer receives external input data and passes it to the membership function computation layer. The system inputs are denoted as $x_1, x_2, \ldots, x_I$, where x_i represents the i-th input variable. The primary function of this layer is

to prepare and provide input data for subsequent computations.

(2) Membership function computation layer: The membership function computation layer fuzzifiers the input variables by mapping each input x_i to the upper and lower membership degrees $\overline{\mu}_{\tilde{A}_i^j}$, and $\underline{\mu}_{\tilde{A}_i^j}$ of the fuzzy set $\tilde{A}_i^j$. The membership functions are defined based on Gaussian distributions, with the calculations depending on the specific horizontal slice μ_s, when the specific value of input x_i is x_i , the calculation process is as follows.

For the baseline horizontal slice ($\mu_s = \beta_0$), they are calculated as:

$$\overline{\mu}_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)=\exp\left(-\frac{\left(x_i'-C_{\tilde{A}_i^j}\right)^2}{\overline{\sigma}_{\tilde{A}_i^j}^2}\right) \tag{1}$$

$$\underline{\mu}_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)=\exp\left(-\frac{\left(x_i'-C_{\tilde{A}_i^j}\right)^2}{\underline{\sigma}_{\tilde{A}_i^j}^2}\right) \tag{2}$$

For other horizontal slices ($\mu_s=\overline{\beta}_k$ and $\mu_s=\underline{\beta}_k$), they are calculated as:

$$\overline{\mu}_{\tilde{A}_i^j|\mu_s=\overline{\beta}_k}\left(x_i'\right)=m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)+\sqrt{\ln\left(\frac{1}{\overline{\beta}_k}\right)\cdot\sigma_{V_i}^2\left(x_i'\right)} \tag{3}$$

$$\underline{\mu}_{\tilde{A}_i^j|\mu_s=\overline{\beta}_k}\left(x_i'\right)=m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)-\sqrt{\ln\left(\frac{1}{\overline{\beta}_k}\right)\cdot\sigma_{V_i}^2\left(x_i'\right)} \tag{4}$$

$$\overline{\mu}_{\tilde{A}_i^j|\mu_s=\overline{\beta}_k}\left(x_i'\right)=m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)+\sqrt{\ln\left(\frac{1}{\overline{\beta}_k}\right)\cdot\sigma_{V_i}^2\left(x_i'\right)} \tag{5}$$

$$\underline{\mu}_{\tilde{A}_i^j|\mu_s=\overline{\beta}_k}\left(x_i'\right)=m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)-\sqrt{\ln\left(\frac{1}{\overline{\beta}_k}\right)\cdot\sigma_{V_i}^2\left(x_i'\right)} \tag{6}$$

where $C_{\tilde{A}_i^j}$ represents the center of the fuzzy set $\tilde{A}_i^j$; $\overline{\sigma}_{\tilde{A}_i^j}^2$ and $\underline{\sigma}_{\tilde{A}_i^j}^2$ are the upper and lower widths of the fuzzy set $\tilde{A}_i^j$, respectively; $j=1,\ldots,J$ represents the number of fuzzy rules; $k=1,\ldots,K$ represents the number of horizontal slices; $\beta_0>0$, $\overline{\beta}_k=\beta_0^{1/\Delta}$, $\beta_k=\beta_0^{\Delta}$ $(\Delta\geq 1)$ are the upper and lower boundaries of horizontal slices; $m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)$ and $\sigma_{V_i}^2\left(x_i'\right)$represents the center value of the membership function and the variance of the input, which are calculated as follows:

$$m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)=\frac{\overline{\mu}_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)+\underline{\mu}_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)}{2} \tag{7}$$

$$\sigma_{V_i}^2\left(x_i'\right)=\frac{\left(\underline{\mu}_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)-m_{\tilde{A}_i^j|\mu_s=\beta_0}\left(x_i'\right)\right)^2}{\ln\left(1/\delta\right)} \tag{8}$$

where δ represents a small positive number.

(3) Rule inference layer: The rule inference layer calculates the firing degrees of fuzzy rules based on the membership degrees of input variables. The j-th fuzzy rule is represented as:

$$\text{If } x_1 \text{ is } \tilde{A}_1^j \text{ and } x_2 \text{ is } \tilde{A}_2^j \text{ and } \cdots x_I \text{ is } \tilde{A}_I^j, \quad (9)$$
$$\text{Then } w \subset [\underline{w}_j, \overline{w}_j]$$

where $\underline{w}_j$ and $\overline{w}_j$ are the upper and lower bounds of the j-th rule consequents, respectively.

The firing degrees for j-th rule are computed using a T-norm operation is shown:

$$\overline{z}^j_{\mu_s=\overline{\beta}_k} = T\left(\overline{\mu}_{\tilde{A}_1^j|\mu_s=\overline{\beta}_k}, \overline{\mu}_{\tilde{A}_2^j|\mu_s=\overline{\beta}_k}, \ldots, \overline{\mu}_{\tilde{A}_I^j|\mu_s=\overline{\beta}_k}\right) \quad (10)$$

$$\underline{z}^j_{\mu_s=\overline{\beta}_k} = T\left(\underline{\mu}_{\tilde{A}_1^j|\mu_s=\overline{\beta}_k}, \underline{\mu}_{\tilde{A}_2^j|\mu_s=\overline{\beta}_k}, \ldots, \underline{\mu}_{\tilde{A}_I^j|\mu_s=\overline{\beta}_k}\right) \quad (11)$$

$$\overline{z}^j_{\mu_s=\underline{\beta}_k} = T\left(\overline{\mu}_{\tilde{A}_1^j|\mu_s=\underline{\beta}_k}, \overline{\mu}_{\tilde{A}_2^j|\mu_s=\underline{\beta}_k}, \ldots, \overline{\mu}_{\tilde{A}_I^j|\mu_s=\underline{\beta}_k}\right) \quad (12)$$

$$\underline{z}^j_{\mu_s=\underline{\beta}_k} = T\left(\underline{\mu}_{\tilde{A}_1^j|\mu_s=\underline{\beta}_k}, \underline{\mu}_{\tilde{A}_2^j|\mu_s=\underline{\beta}_k}, \ldots, \underline{\mu}_{\tilde{A}_I^j|\mu_s=\underline{\beta}_k}\right) \quad (13)$$

where $\overline{z}^j_{\mu_s=\overline{\beta}_k}$ and $\overline{z}^j_{\mu_s=\underline{\beta}_k}$ are the upper bound firing degrees of j-th rule at horizontal slices in terms of $\mu_s=\overline{\beta}_k$ and $\mu_s=\underline{\beta}_k$, respectively; $\underline{z}^j_{\mu_s=\overline{\beta}_k}$ and $\underline{z}^j_{\mu_s=\underline{\beta}_k}$ are the lower bound firing degrees of j-th rule at horizontal slices in terms of $\mu_s=\overline{\beta}_k$ and $\mu_s=\underline{\beta}_k$, respectively; and $T(\cdot)$ is the T-norm operator for aggregating input membership degrees.

(4) Type-reduction layer: The type-reduction layer combines the firing degrees and rule consequents to compute the consequent values for each horizontal slice. The calculations are as follows:

$$\overline{f}_k = \frac{\sum_{j=1}^{J}\left(\overline{z}^j_{\mu_s=\overline{\beta}_k}\cdot\overline{w}_l + \underline{z}^j_{\mu_s=\overline{\beta}_k}\cdot\underline{w}_l\right)}{\sum_{j=1}^{J}\left(\overline{z}^j_{\mu_s=\overline{\beta}_k} + \underline{z}^j_{\mu_s=\overline{\beta}_k}\right)} \quad (14)$$

$$\underline{f}_k = \frac{\sum_{j=1}^{J}\left(\overline{z}^j_{\mu_s=\underline{\beta}_k}\cdot\overline{w}_l + \underline{z}^j_{\mu_s=\underline{\beta}_k}\cdot\underline{w}_l\right)}{\sum_{l=1}^{M}\left(\overline{z}^l_{\mu_s=\underline{\beta}_k} + \underline{z}^l_{\mu_s=\underline{\beta}_k}\right)} \quad (15)$$

where $\overline{f}_k$ and $\underline{f}_k$ are the consequent values at horizontal slices in terms of $\mu_s=\overline{\beta}_k$ and $\mu_s=\underline{\beta}_k$.

(5) Output layer: The output layer aggregates all the consequent values from the type-reduction layer to produce the final system output. The calculation is given by:

$$p = \frac{\sum_{k=1}^{K}(\overline{\beta}_k\overline{f}_k + \underline{\beta}_k\underline{f}_k)}{\sum_{k=1}^{K}(\overline{\beta}_k + \underline{\beta}_k)} \quad (16)$$

where p is the final system output of IT3FLS with its input is $[x_1, x_2, \ldots, x_I]$.

2.2 Broad Learning System (BLS)

The BLS is an efficient learning framework designed to achieve fast modeling and training through a shallow and parallel network structure. BLS constructs broad features using feature mapping nodes and enhancement nodes, while optimizing only the weight matrix of the output layer. This avoids the iterative training process required in conventional deep learning methods. Below, the computational process of BLS is described layer by layer, including

the input layer, feature mapping layer, enhancement layer, and output layer.

(1) Input layer: The input layer receives the original data and forwards it to the feature mapping layer. Assume the value of input data is $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, where N is the number of samples and D is the dimensionality of the input features. The input layer simply passes the data to the next layer without performing additional processing.

(2) Feature mapping layer: The feature mapping layer performs nonlinear feature transformations on the input data to generate a high-dimensional feature space, as follows,

$$\boldsymbol{P} = \Omega(\boldsymbol{X}, \boldsymbol{\Phi}) \tag{17}$$

where $\boldsymbol{P} \in \mathbb{R}^{N \times M}$ is the output matrix of feature mapping layer with input $\boldsymbol{X}$, $\boldsymbol{M}$ is the number of feature mapping nodes in this layer, $\boldsymbol{\Phi}$ are all parameters of the feature map layer, and $\Omega(\cdot)$ is a nonlinear activation function.

(3) Enhancement layer: The enhancement layer further enriches the feature space by providing additional redundant and nonlinear representations. It takes the feature node matrix $\boldsymbol{P}$ as input and generates the enhancement node matrix $\boldsymbol{H} \in \mathbb{R}^{N \times L}$ using a randomly initialized weight matrix $\boldsymbol{\Theta} \in \mathbb{R}^{M \times L}$ and a bias vector $\boldsymbol{c} \in \mathbb{R}^{L}$. The computation process of $\boldsymbol{H}$ is defined as:

$$\boldsymbol{H} = g(\boldsymbol{P}\boldsymbol{\Theta} + \boldsymbol{c}) \tag{18}$$

where L is the number of enhancement nodes and $g(\cdot)$ is the nonlinear activation function.

(4) Output layer: The output layer combines the feature node matrix $\boldsymbol{P}$ and the enhancement node matrix $\boldsymbol{H}$ into a unified feature matrix $\boldsymbol{Q} = [\boldsymbol{P}, \boldsymbol{H}] \in \mathbb{R}^{N \times (M+L)}$. For the target output $\boldsymbol{Y} \in \mathbb{R}^{N \times c}$, where c is the output dimension, the predicted output $\hat{\boldsymbol{Y}}$ is calculated as:

$$\hat{\boldsymbol{Y}} = \boldsymbol{Q}\boldsymbol{W}_o \tag{19}$$

where $\boldsymbol{W}_o \in \mathbb{R}^{(M+L) \times c}$ represents the weight matrix, which is optimized using ridge regression as follows,

$$\boldsymbol{W}_o = (\boldsymbol{Q}^{\cdot}\boldsymbol{Q} + \lambda \boldsymbol{I})^{-1}\boldsymbol{Q}^{\cdot}\boldsymbol{Y} \tag{20}$$

where the λ represents the regularization parameter, and $\boldsymbol{I}$ is the identity matrix.

3. Interval Type Three Fuzzy Broad Learning System (IT3FBLS)

The proposed IT3FBLS algorithm consists of 4 layers: the input layer, the interval type-3 fuzzy mapping layer, the enhancement node layer, and the output layer.

3.1 Structure and Implementation

These layers together compute the output of the model to achieve the regression task. The detailed implementation description of each layer is as follows (1) Input layer: The input sample of the input layer can be described as

$$\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \cdots, \boldsymbol{x}_N], \text{ where } \boldsymbol{x}_n = \left[x_{1,n}, \cdots, x_{d,n}, \cdots, x_{D,n}\right]^{\mathrm{T}} \in \mathbb{R}^{D \times 1} \tag{21}$$

where D is the number of features of the sample. The input $\boldsymbol{X}$ is passed to the interval type-3 fuzzy mapping layer for further processing.

(2) Interval type-3 fuzzy mapping layer: The interval type-3 fuzzy mapping layer consists of M independent IT3FS units. Each IT3FS receives the input $\boldsymbol{X}$ and computes a mapping output $\boldsymbol{p}_m = \left[p_m^1, p_m^2, \cdots, p_m^N\right]^T \in \mathbb{R}^{N \times 1}$. The outputs of all IT3FS units form the feature matrix in (1)~(16):

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M] \in \mathbb{R}^{N \times M} \tag{22}$$

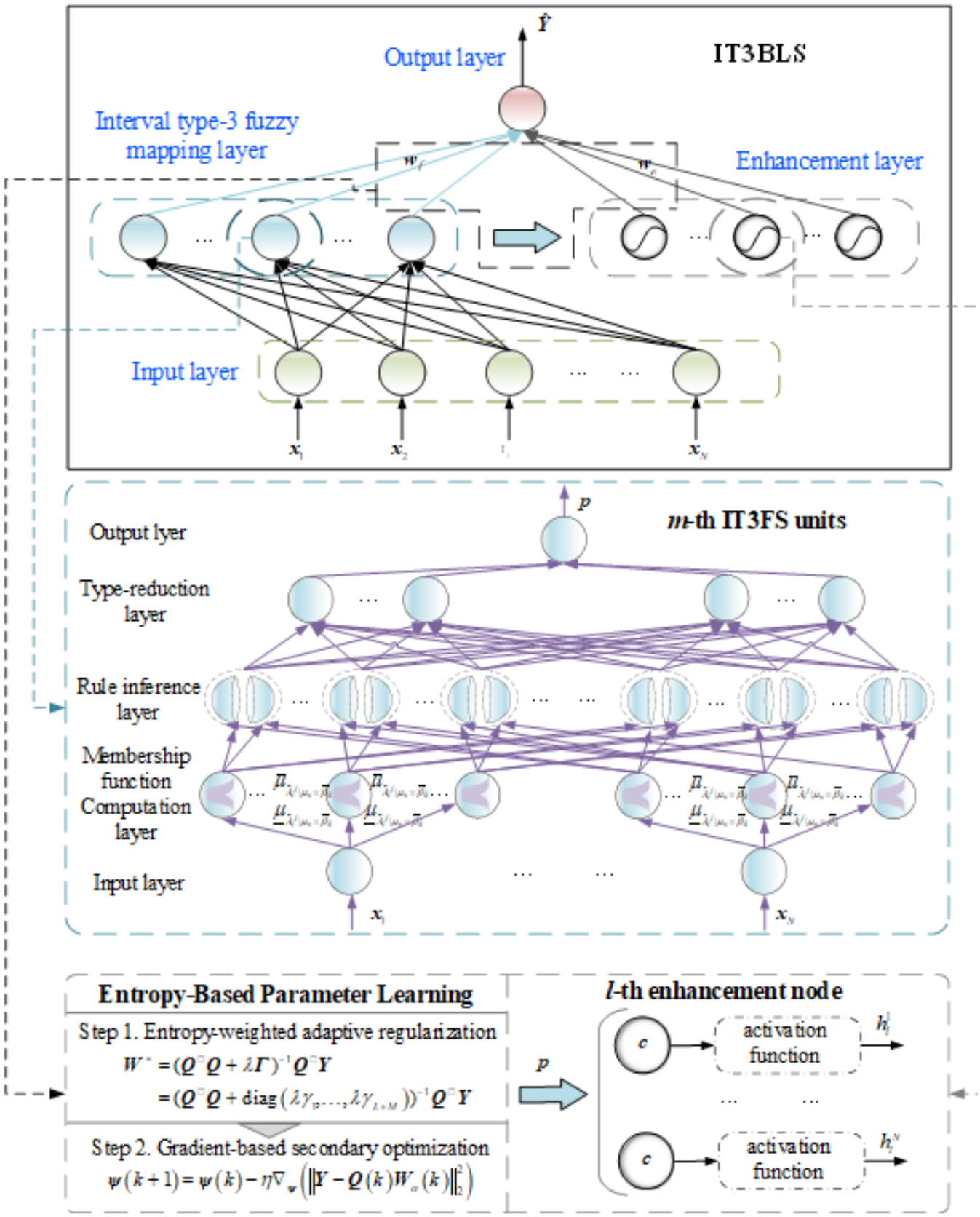


Figure 1: IT3FBLS structure

This layer provides robust feature mapping under uncertainties and serves as the foundation for the subsequent enhancement layer.

(3) Enhancement layer: The enhancement layer expands the feature space, complementing the outputs of the interval type-3 fuzzy mapping layer. With L enhancement nodes, the enhancement node output matrix $\boldsymbol{H}=\left[\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_L\right]\in\boxtimes^{N\times L}$ is computed using $\boldsymbol{V}$ and c in (18), where $\boldsymbol{h}_i=\left[h_i^1,h_i^2,\cdots,h_i^N\right]^T\in\boxtimes^{N\times 1}$.

Then, the final feature matrix $\boldsymbol{Q}=[\boldsymbol{P},\boldsymbol{H}]\in\boxtimes^{N\times(M+L)}$ is obtained by combining $\boldsymbol{P}$ and $\boldsymbol{H}$.

(4) Output layer: Utilizing the IT3FS module for feature mapping enables a robust representation of data uncertainty, thereby enhancing the performance of regression modeling tasks [28]. The output of IT3FBLS is:

$$\begin{aligned}\hat{\boldsymbol{Y}}&=\boldsymbol{Q}\boldsymbol{W}_o=\left[\boldsymbol{P},\boldsymbol{H}\right]\cdot\left[\boldsymbol{w}_f,\boldsymbol{w}_e\right]\\&=\left[\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_M\,|\,\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_L\right]\cdot\left[w_{f1},w_{f2},\cdots,w_{fM}\,|\,w_{e1},w_{e2},\cdots,w_{eL}\right]^{\mathrm{T}}\end{aligned}\tag{23}$$

where $\boldsymbol{W}_o\in\boxtimes^{(M+L)\times 1}$ is a weight matrix, connecting the interval type-3 fuzzy mapping layer, enhancement layer, and output layer.

3.2 Entropy-Based Parameter Learning

To enhance the robustness and generalization performance of the BLS, this article introduces the concept of node-level entropy, which is used to measure the discreteness of the output distribution of each fuzzy subsystem and enhanced node. This entropy measure is then used to adaptively adjust the regularization strength for each node within the penalty term.

3.2.1 Entropy-Weighted Adaptive Regularization

The output layer weights of BLS are typically computed through classical ridge regression algorithm, with the objective function expressed as:

$$\min_{\boldsymbol{W}_o}\left(\left\|\boldsymbol{Y}-\boldsymbol{Q}\boldsymbol{W}_o\right\|_2^2+\lambda\left\|\boldsymbol{W}_o\right\|_2^2\right)\tag{24}$$

where $\lambda\geq 0$ is the regularization coefficient, $\boldsymbol{Y}$ is the true output of the samples, and the solution for $\boldsymbol{W}_o$ can be derived from (20).

The output distributions of fuzzy subsystems within the IT3FS-based mapping layer and enhancement nodes within the enhancement layer often exhibit significant differences. To better capture these localized mapping characteristics, information entropy is introduced to quantify the output distribution of each subsystem or node. This entropy measure is then used to weight the regularization strength, replacing the uniform penalty term with a component-wise weighted form. The modified objective function is given by:

$$\min_{\boldsymbol{W}_o}\left(\left\|\boldsymbol{Y}-\boldsymbol{Q}\boldsymbol{W}_o\right\|_2^2+\lambda\sum_{i=1}^{M+L}\gamma_i w_i^2\right)\tag{25}$$

where γ_i represents the regularization weight for the ith fuzzy subsystem or enhancement node, reflecting the impact of its distribution entropy on the regularization strength. With $\gamma_i>0$, a larger γ_i imposes a stronger penalty on the corresponding weight, while a smaller value results in a weaker penalty. The specific implementation is as follows:

First, let $\boldsymbol{q}_i=\left[q_i^1,q_i^2,\cdots,q_i^N\right]^{\mathrm{T}}\in\boxtimes^N$ denote the ith column of matrix $\boldsymbol{Q}$, representing the output of the ith fuzzy subsystem or enhancement node for a dataset with N samples. Define q_i as a random variable representing the distribution of $\boldsymbol{q}_i$, whose probability density function (PDF) can be estimated using a Gaussian kernel as:

$$\hat{\gamma}_{q_i}\left(q_i\right)=N^{-1}\sum_{n=1}^{N}\kappa_{q_i}\left(q_i-q_i^n\right)\tag{26}$$

where q_i^n is the nth value of $\boldsymbol{q}_i$, and $\kappa_{q_i}\left(q_i - q_i^n\right)$ is calculated as:

$$\kappa_{q_i}\left(q_i - q_i^n\right) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{\left(q_i - q_i^n\right)^2}{2\sigma_i^2}\right) \tag{27}$$

where σ_i the standard deviation of the sample data.

After obtaining the PDF through kernel density estimation (KDE), the marginal information potential can be expressed as:

$$V(q_i) = \int \hat{\gamma}_{q_i}\left(q_i\right)^2 dq_i \tag{28}$$

Substituting (26) and (27) into (28) yields:

$$\begin{aligned} V(q_i) &= \int\left(\frac{1}{N}\sum_{l=1}^{N}\kappa_{q_i}(q_i - q_{i,l})\right)^2 \mathrm{d}q_i \\ &= \frac{1}{N^2}\int\sum_{l=1}^{N}\sum_{j=1}^{N}\kappa_{q_i}(q_i - q_{i,l})\kappa_{q_i}(q_i - q_{i,j})\mathrm{d}q_i \\ &= \frac{1}{N^2}\sum_{l=1}^{N}\sum_{j=1}^{N}\int\kappa_{q_i}(q_i - q_{i,l})\kappa_{q_i}(q_i - q_{i,j})\mathrm{d}q_i \end{aligned} \tag{29}$$

According to (27), $\int\kappa_{q_i}(q_i - a)\kappa_{q_i}(q_i - b)\mathrm{d}q_i$ can be computed as:

$$\begin{aligned} &\int\kappa_{q_i}(q_i - a)\kappa_{q_i}(q_i - b)\mathrm{d}q_i = \int\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(q_i - a)^2}{2\sigma_i^2}\right)\cdot\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(q_i - b)^2}{2\sigma_i^2}\right)\mathrm{d}q_i \\ &= \frac{1}{2\pi\sigma_i^2}\int\exp\left(-\frac{(q_i - a)^2 + (q_i - b)^2}{2\sigma_i^2}\right)\mathrm{d}q_i = \frac{1}{2\pi\sigma_i^2}\exp\left(-\frac{(a-b)^2}{4\sigma_i^2}\right)\int\exp\left(-\frac{(q_i - \frac{a+b}{2})^2}{\sigma_i^2}\right)\mathrm{d}q_i \\ &= \frac{1}{2\pi\sigma_i^2}\exp\left(-\frac{(a-b)^2}{4\sigma_i^2}\right)\cdot\sqrt{\pi}\sigma_i = \frac{1}{2\sqrt{\pi}\sigma_i}\exp\left(-\frac{(a-b)^2}{4\sigma_i^2}\right) \end{aligned} \tag{30}$$

Substituting (30) into (29) gives the discrete form of the marginal information potential q_i under the Gaussian kernel as follows:

$$\begin{aligned} V(q_i) &= \frac{1}{N^2}\sum_{l=1}^{N}\sum_{j=1}^{N}\frac{1}{2\sqrt{\pi}\sigma_i}\exp\left(-\frac{(q_{i,l} - q_{i,j})^2}{4\sigma_i^2}\right) \\ &= N^{-2}\frac{1}{2\sqrt{\pi}\sigma_i}\sum_{l=1}^{N}\sum_{j=1}^{N}\exp\left(-\frac{(q_{i,l} - q_{i,j})^2}{4\sigma_i^2}\right). \end{aligned} \tag{31}$$

This calculation requires only N^2 kernel function evaluations to obtain the information potential $V(q_j)$.

Next, for each fuzzy subsystem or enhancement node, after calculating its information potential, an exponential mapping function is applied to transform the potential into a regularization weight:

$$\gamma_i = \exp\left(-\alpha \cdot V(q_i)\right) \tag{32}$$

where α is a factor with $0 < \alpha < 1$.:

Then, using the diagonal matrix $\boldsymbol{\Gamma} = \mathrm{diag}\left(\gamma_1, \ldots, \gamma_{L+M}\right)$, the regularized term in (25) can be reformulated as:

$$\lambda \sum_{i=1}^{M+L} \gamma_i w_i^2 = \lambda \cdot \boldsymbol{W}_o^{\cdot} \boldsymbol{\Gamma} \boldsymbol{W}_o \tag{33}$$

Simplifying the first term in (25) yields:

$$\begin{aligned} &\left\| \boldsymbol{Y} - \boldsymbol{Q}\boldsymbol{W}_o \right\|_2^2 = \left(\boldsymbol{Y} - \boldsymbol{Q}\boldsymbol{W}_o \right)^{\cdot} \left(\boldsymbol{Y} - \boldsymbol{Q}\boldsymbol{W}_o \right) \\ &= \boldsymbol{Y}^{\cdot} \boldsymbol{Y} - 2\boldsymbol{W}_o^{\cdot} \boldsymbol{Q}^{\cdot} \boldsymbol{Y} + \boldsymbol{W}_o^{\cdot} \boldsymbol{Q}^{\cdot} \boldsymbol{Q} \boldsymbol{W}_o \end{aligned} \tag{34}$$

Finally, substitute (34) into (25) , and let be the optimal solution.

By setting the follows:

$$\nabla_{\boldsymbol{W}_o = \boldsymbol{W}^*} \left(\left\| \boldsymbol{Y} - \boldsymbol{Q}\boldsymbol{W}_o \right\|_2^2 + \lambda \cdot \boldsymbol{W}_o^{\cdot} \boldsymbol{\Gamma} \boldsymbol{W}_o \right) = 0 \tag{35}$$

Let $\boldsymbol{Q}^{\cdot} \boldsymbol{Q} = \boldsymbol{A}$, we get:

$$\nabla \left(\left\| \boldsymbol{Y} - \boldsymbol{Q}\boldsymbol{W}_o \right\|_2^2 + \lambda \cdot \boldsymbol{W}_o^{\cdot} \boldsymbol{\Gamma} \boldsymbol{W}_o \right) = -2\boldsymbol{Q}^{\cdot} \boldsymbol{Y} + 2\boldsymbol{A}\boldsymbol{W}_o^{\cdot} + 2\lambda \boldsymbol{\Gamma} \boldsymbol{W}_o^{\cdot} \tag{36}$$

Substituting (35) into (36), the optimal weight of $\boldsymbol{W}_o$ can be calculated as follows:

$$\begin{aligned} \boldsymbol{W}^* &= (\boldsymbol{Q}^{\cdot} \boldsymbol{Q} + \lambda \boldsymbol{\Gamma})^{-1} \boldsymbol{Q}^{\cdot} \boldsymbol{Y} \\ &= (\boldsymbol{Q}^{\cdot} \boldsymbol{Q} + \operatorname{diag}(\lambda \gamma_1, \ldots, \lambda \gamma_{L+M}))^{-1} \boldsymbol{Q}^{\cdot} \boldsymbol{Y} \end{aligned} \tag{37}$$

Notably, by measuring the output distribution entropy of each node and adaptively adjusting the corresponding regularization strength, the proposed approach effectively suppresses the influence of abnormal or unstable nodes, thereby improving the generalization performance of the IT3FBLS model.

3.2.2 Gradient-Based Secondary Optimization

In the entropy-weight adaptive regularization algorithm described earlier, the output layer weights $\boldsymbol{W}_o$ of IT3FBLS can be efficiently computed through entropy-enhanced ridge regression algorithm. However, the internal parameters of IT3FBLS, including the weight matrix $\boldsymbol{\Theta}$ and the bias vector *c* defined in (18), play a crucial role in network performance but cannot be directly solved using the modified ridge regression algorithm.

If these internal parameters remain initialized through random search or fixed settings, the model's potential cannot be fully exploited. To address this, a gradient descent (GD)-based secondary optimization strategy is introduced, enabling collaborative optimization between output weights and internal parameters during each iteration. This approach enhances both approximation accuracy and convergence speed.

The specific implementation steps for the *k*th iteration are as follows:

First, given the training data **X** and **Y** , and the current weight matrix $\boldsymbol{P}(k)$ and $\boldsymbol{c}(k)$, the outputs of the fuzzy subsystems $\boldsymbol{\Theta}(k)$ and enhancement nodes $\boldsymbol{H}(k)$ can be calculated using (22). As indicated by (18), the values of $\boldsymbol{P}(k)$ depend solely on $\boldsymbol{X}$, while $\boldsymbol{H}(K)$ is determined by $\boldsymbol{P}(K)$, $\boldsymbol{\Theta}(k)$, and $\boldsymbol{C}(K)$.

Second, using KDE as defined in (26)~(31), the marginal information potential for each node is estimated, and the corresponding weighting coefficients $\gamma_1(k), \ldots, \gamma_{L+M}(k)$ are derived via (32). These coefficients are then used to construct the weighted diagonal matrix $\boldsymbol{\Gamma}(k)$. The output layer weights $\boldsymbol{W}_o(k)$ for the current iteration are computed directly via (37).

Next, for internal parameters $\boldsymbol{\psi}(k) = \left[\boldsymbol{\Theta}(k), \boldsymbol{c}(k) \right]$, which cannot be updated through entropy-weighted ridge regression algorithm, GD is applied as follows:

$$\psi(k+1)=\psi(k)-\eta\nabla_{\psi}\left(\left\|\boldsymbol{Y}-\boldsymbol{Q}(k)\boldsymbol{W}_{o}(k)\right\|_{2}^{2}\right) \quad (38)$$

where $\eta>0$ is the learning rate, and the specific calculation process follows the [13].

Then, once $\psi(k)=\left[\boldsymbol{\Theta}(k),c(k)\right]$ are updated, the distribution characteristics of the fuzzy subsystems and enhancement nodes change accordingly. Therefore, the marginal information potential and weighting coefficients are recalculated, leading to an updated weighted diagonal matrix $\boldsymbol{\Gamma}(k+1)$ $\boldsymbol{W}_{o}(k+1)$ The are then updated via (37), and $\psi(k+1)$ are then updated via (38).

Finally, the above steps are repeated until the maximum number of iterations is reached or the convergence criterion is satisfied, thereby completing the gradient-based secondary optimization process for IT3FBLS parameters.

It is worth noting that, under traditional supervised learning frameworks, each update of internal parameters typically necessitates a complete forward pass through the entire network, followed by backpropagation to compute gradients—a process that demands substantial computational resources. In contrast, the proposed parameter learning framework employs an efficient hybrid approach: the output layer weights are computed via a closed-form ridge regression solution, while the internal parameters are locally adjusted using gradient descent. Information potential and the weighted diagonal matrix are recalculated only when necessary. By handling parameter updates in each iteration, the approach leverages the computational efficiency of ridge regression algorithm alongside the flexibility of gradient descent, allowing the model to rapidly approach an optimal solution.

3.3 Computational Complexity analysis

The compSZutational complexity of the IT3FBLS model is determined by the three main components of its architecture: the interval type-3 fuzzy mapping layer, the enhancement layer, and the output layer. The detailed complexity of each component is analyzed in details as follows

3.3.1 Complexity of Interval Type-3 Fuzzy Mapping Layer

The interval type-3 fuzzy mapping layer contains independent IT3FS. Each IT3FS computes its output through several stages.

Membership function computation layer computes the membership functions of all input features over j fuzzy rules and k horizontal slices. The complexity is as follows:

$$O_{\text{Membership}}=O(N\times J\times K) \quad (39)$$

For industrial control tasks, N=1.

Rule inference layer: Inference over j rules has a complexity as follows:

$$O_{\text{Inference}}=O(J) \quad (40)$$

Type-reduction layer and output layer: The type-reduction process with the final output computation calculates aggregated outputs over j rules and k slices, with a complexity as follows:

$$O_{\text{Output}}=O(J\times K) \quad (41)$$

The total complexity of a single IT3FS is:

$$O_{\text{IT3FS}}=O(J\times K)+O(J)+O(J\times K) \quad (42)$$

With M IT3FS units in the interval type-3 fuzzy mapping layer, the total complexity becomes:

$$O_{\text{Mapping}} = M \times (2 \cdot O(J \times K) + O(J)) \quad (43)$$

3.3.2 Complexity of Enhancement Layer

The enhancement node layer computes additional feature transformations through L enhancement nodes:

Matrix multiplication: Combining the outputs of the mapping layer with requires:

$$O_{\text{Enhance-Multiply}} = O(M \times L) \quad (44)$$

Activation function: Applying the activation function to L enhancement nodes results in:

$$O_{\text{Enhance-Activation}} = O(L) \quad (45)$$

The total complexity of the enhancement node layer is as follows:

$$O_{\text{Enhance}} = O(M \times L) + O(L) \quad (46)$$

3.3.3 Complexity of Output Layer

The output layer computes the output i based on $\boldsymbol{Q}$. The complexity of the matrix-vector multiplication with $\boldsymbol{W}_o$ is:

$$O_{\text{Output}} = O(M + L) \quad (47)$$

Therefore, the total computational complexity of IT3FBLS is the sum of the three components as follows:

$$\begin{aligned} O_{\text{IT3FBLS}} &= O_{\text{Mapping}} + O_{\text{Enhance}} + O_{\text{Output}} \\ &= M \times (2 \cdot O(J \times K) + O(J)) + \\ &\quad 2 \cdot O(M \times L) + O(L) \end{aligned} \quad (48)$$

In summary, the parameter choices for M, J, K, and L must balance computational complexity and system performance. Among these factors, M is the primary determinant of complexity, and its adjustment has the most significant impact on computational cost. It should be selected within a reasonable range. The values of K and J primarily affect the complexity of the mapping layer and have a greater influence on real-time performance. These parameters can be reduced when modeling performance demands are low. The growth L of has the least effect on real-time performance and can be increased to enhance the model's adaptability and generalization capabilities.

Remark: In the proposed gradient quadratic optimization algorithm, since the most computationally interval type-3 fuzzy mapping layer does not require repeated forward computation, redundant calculations are minimized. This not only reduces computational costs but also ensures the scalability of the proposed model for large-scale and real-time applications.

3.4 Algorithm Pseudocode

The pseudocode of the proposed IT3FBLS model is as follows:

Algorithm 1 IT3FBLS modeling algorithm	
Input: Training data $\{X, Y\}$; Initial model parameters; Regularization parameter; Learning rate ; Maximum iteration number	
Output: Model prediction output $\hat{Y}$	
(1). Initialization	Compute the outputs of the fuzzy subsystems P and enhancement nodes H // (21)~(22)
	Concatenate P and H to construct matrix Q, and calculate the initial model prediction output $\hat{Y}$ // (23)
(2). Calculate information entropy and regularity coefficient	For each column of Q, estimate the PDF using KDE and calculate the corresponding information entropy // (26)~(31)
	Apply the exponential mapping function to transform the entropy into regularization weights and construct the diagonal matrix $\Gamma(k)$ // (32)
(3). Update $W_o(k)$	Update the output layer weights $W_o(k)$ using the closed-form solution derived from the entropy-weighted adaptive regularization approach // (37)
(4). Gradient descent for internal parameters:	Compute the gradient of loss function with respect to $\psi(k)=[\Theta(k),c(k)]$, and update these parameters using gradient descent // (38)
(5). Gradient-based secondary optimization	After updating $\psi(k)$, recalculate the outputs of the enhancement nodes $H(k+1)$ and the corresponding information entropy
	Update the output layer weights again using the updated
	Repeat Steps 2 through 5 until the maximum number of iterations is reached.
(6). Output results	After IT3FBLS training is completed, the model prediction output is
	calculated as the final output of the model

4. Experiments and Discussion

In this section, a systematic study is conducted to evaluate the effectiveness of the proposed fuzzy broad learning system and its associated learning algorithms.

4.1 Modeling Datasets and Evaluation Indicators Description

To ensure a rigorous and comprehensive evaluation, three categories of datasets are designed:

(1) Benchmark dataset. This dataset employs three representative publicly available benchmark datasets for data-driven regression modeling. These datasets include one medium-dimensional and two low-dimensional datasets sourced from the University of California, Irvine (UCI) Machine Learning Repository [29]. The aim is to evaluate the proposed algorithm's modeling capability across different data dimensionalities and complexities.

(2) Uncertain function dataset. This experiment investigates the robustness and generalization ability of the proposed algorithm in handling dynamic nonlinearity and high-uncertainty environments. Two classical chaotic systems, the Mackey–Glass time series [30] and the Rossler attractor [31], are used to test the model's effectiveness in capturing uncertainty-driven behaviors.

(3) Complex industrial process dataset. To examine the practical applicability and feasibility of the proposed approach in real-world industrial settings, this experiment applies the algorithm to data-driven modeling tasks in industrial processes. Specifically, it focuses on furnace temperature modeling in MSWI [32] and process modeling in the Tennessee Eastman (TE) chemical process [33]. All process data are collected in real time from actual industrial production environments, ensuring a realistic assessment of the algorithm's industrial applicability.

For these datasets, an equal-interval selection strategy is employed, and all datasets are standardized and divided into three subsets: training, validation, and testing. The details of this partition are presented in Table 1.

Table 1: Dataset details for regression modeling

Datasets	Details of datasets	No. of Samples			Input variables
		Train	Validation	Test	
1	Concrete compressive strength (CCS)	515	257	258	8
	Abalone	2088	1044	1045	8
	Housing	253	126	127	13
2	Mackey–Glass	400	200	200	4
	Rossler attractor	500	250	250	3
3	MSWI process	428	214	215	5
	TE chemical process	250	125	125	11

To evaluate model performance, three commonly used regression indicators are adopted: root mean square error (RMSE), mean absolute error (MAE), and R2. To ensure a fair evaluation of the simulation results, all experiments are repeated 20 times, and the mean, variance, and optimal values of the evaluation metrics are recorded.

4.2 Hyperparameter Setting Description

Furthermore, to validate the effectiveness of the

proposed parameter learning algorithm, a comparison is made between IT3FBLS employing traditional ridge regression algorithm (denoted as IT3FBLS-1) and IT3FBLS utilizing entropy-weighted adaptive regularization (denoted as IT3FBLS-2). The hyperparameters of IT3FBLS include the number of fuzzy rules J, number of horizontal slices K, number of fuzzy subsystems M, number of enhancement nodes L, learning rate , and regularization coefficient λ . These hyperparameters are determined through a grid search approach, following these steps: First, identify the hyperparameters that require tuning and define a candidate value range for each. Next, enumerate all possible hyperparameter combinations and train the model for each configuration. Finally, compare the evaluation results across all configurations and select the combination that yields the best average performance as the final hyperparameter setting. For fairness, the hyperparameter configurations of other benchmark models in comparison experiments are aligned with those of IT3FBLS as closely as possible. The specific hyperparameter settings used in the experiments are detailed in Table 2.

Table 2: Hyperparameter settings of the proposed algorithms on different datasets

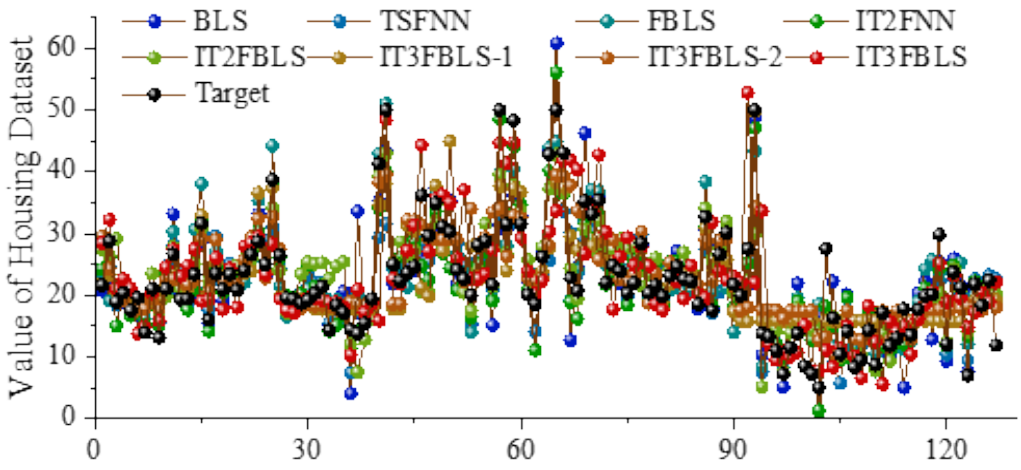
Datasets	Hyperparameters of IT3FBLS					
	J	K	M	L	η	λ
Range	[1,100]	[1,50]	[1,1000]	[1,2050]	[0,1]	[0,1]
CCS	66	7	460	140	0.7	0.07
Abalone	8	4	835	176	0.3	0.2
Housing	48	11	271	490	0.45	0.4
Mackey–Glass	8	40	20	800	0.3	0.3
Rossler attractor	6	4	10	800	0.3	0.3
MSWI process	42	6	125	1610	0.22	0.16
TE chemical process	4	40	10	600	0.3	0.2

To comprehensively assess its performance, comparative experiments are performed against the original BLS [23] , classical regression models including FNN [34] and FBLS 35], as well as advanced regression algorithms integrating fuzzy systems and broad learning, including IT2FNN [36] and IT2FBLS [37].

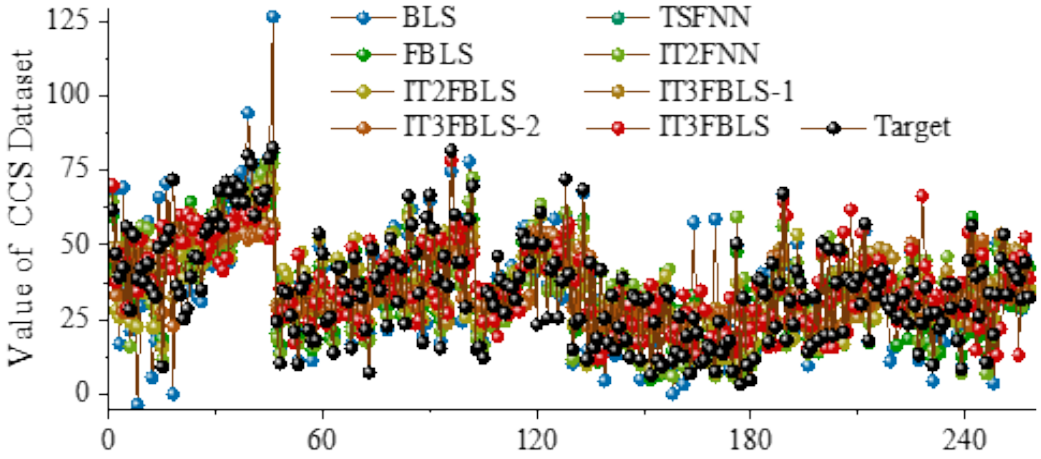
4.3 Results and Discussion

4.3.1 Benchmark dataset

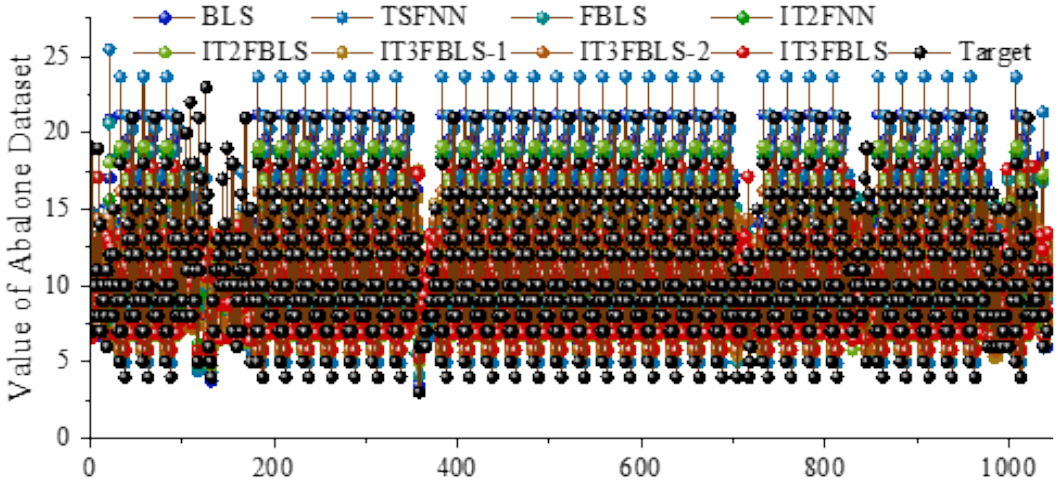
The prediction results and comparison results of the testing set are shown in Fig. 2 and Table 3. In Table 3, the bolded values indicate the two best-performing methods for each dataset across different evaluation metrics.



(a) Housing dataset



(b) CCS dataset



(c) Abalone dataset

Figure 2: Prediction results of the testing set for the benchmark dataset

Table 3: Performance comparison results of the testing set data on the benchmark dataset

Datasets	Methods	RMSE			MAE			R2		
		Mean	Var	Best	Mean	Var	Best	Mean	Var	Best
Housing	BLS	1.8988E+01	1.7532E+01	4.9911E+00	5.0214E+00	1.7552E+00	3.3301E+00	-6.5651E+00	1.2706E+01	7.1119E-01
	FNN	6.1635E+00	9.4208E-01	5.4973E+00	4.0787E+00	6.4398E-01	3.6233E+00	5.5443E-01	1.3463E-01	6.4964E-01
	FBLS	5.9531E+00	1.1426E+00	4.7884E+00	3.9587E+00	3.9816E-01	3.3284E+00	5.7475E-01	1.6724E-01	7.8698E-01
	IT2FNN	**4.1984E+00**	**4.1219E-01**	**3.5039E+00**	**3.0080E+00**	**1.9918E-01**	**2.6240E+00**	**7.9378E-01**	**4.1478E-02**	**8.5766E-01**
	IT2FBLS	6.3649E+00	7.3477E-01	4.9374E+00	4.5117E+00	5.0624E-01	3.5953E+00	5.2438E-01	1.0990E-01	7.1737E-01
	IT3FBLS-1	6.2516E+00	7.3677E-01	5.5300E+00	4.6600E+00	4.8271E-01	4.1938E+00	5.4123E-01	1.1426E-01	6.4546E-01
	IT3FBLS-2	6.1518E+00	7.0482E-01	4.9612E+00	4.1655E+00	5.0065E-01	3.8651E+00	5.8664E-01	1.0812E-01	6.8520E-01
	IT3FBLS	**5.8991E+00**	**3.6990E-01**	**4.2865E+00**	**3.8528E+00**	**2.2710E-01**	**3.2322E+00**	**6.0470E-01**	**4.7700E-02**	**7.9050E-01**
CCS	BLS	3.5018E+01	3.8531E+01	1.0920E+01	9.9500E+00	4.3984E+00	6.3229E+00	-7.8145E+00	2.3569E+01	6.0136E-01
	FNN	1.2610E+01	6.2004E-01	1.1921E+01	8.8290E+00	9.0309E-01	8.0701E+00	4.6742E-01	5.3011E-02	5.2493E-01
	FBLS	9.9873E+00	1.0259E+00	8.1331E+00	7.5818E+00	8.2169E-01	6.0590E+00	**6.6320E-01**	6.8675E-02	**7.4887E-01**
	IT2FNN	**8.3817E+00**	3.6140E-01	**7.4174E+00**	**6.3209E+00**	2.7262E-01	**5.5798E+00**	5.2650E-01	**1.9941E-02**	7.1607E-01
	IT2FBLS	1.2468E+01	1.3018E+00	9.3524E+00	9.6935E+00	1.0329E+00	7.3395E+00	4.7491E-01	1.0449E-01	7.0760E-01
	IT3FBLS-1	1.3538E+01	**2.8278E-01**	1.3235E+01	1.1004E+01	**1.9383E-01**	1.0824E+01	3.8710E-01	**2.5980E-02**	4.1439E-01
	IT3FBLS-2	1.3860E+01	7.2619E-01	1.3084E+01	1.1275E+01	5.5857E-01	1.0557E+01	3.5637E-01	6.7368E-02	4.2767E-01
	IT3FBLS	**2.4234E+00**	**1.5360E-01**	**2.1406E+00**	**1.8176E+00**	**1.2040E-01**	**1.6300E+00**	**7.6473E-01**	5.9300E-02	**7.3200E-01**
Abalone	BLS	8.2271E+00	1.1380E+01	2.6158E+00	2.6303E+00	6.5151E-01	1.9711E+00	-1.1315E+01	4.3493E+01	5.5819E-01
	FNN	2.9319E+00	**3.5560E-02**	2.9068E+00	2.2705E+00	**5.0842E-02**	2.2346E+00	4.4492E-01	**1.3464E-02**	4.5444E-01
	FBLS	2.7202E+00	2.7033E-01	2.3118E+00	1.9126E+00	1.3661E-01	1.7431E+00	5.1777E-01	9.9518E-02	6.5492E-01
	IT2FNN	2.4935E+00	1.7485E-01	2.3083E+00	1.8871E+00	1.2127E-01	1.7381E+00	5.9677E-01	5.8821E-02	6.5598E-01
	IT2FBLS	**2.3091E+00**	1.3151E-01	**2.1305E+00**	**1.7435E+00**	1.1133E-01	**1.5805E+00**	**6.5468E-01**	3.9673E-02	**7.0694E-01**
	IT3FBLS-1	2.6090E+00	1.2095E-01	2.4669E+00	1.9028E+00	1.1005E-01	1.7313E+00	5.5973E-01	4.0807E-02	6.0707E-01
	IT3FBLS-2	**2.3936E+00**	**6.6976E-02**	2.3062E+00	1.7696E+00	**4.6055E-02**	1.7170E+00	**6.2984E-01**	**2.0760E-02**	**6.5660E-01**
	IT3FBLS	2.3975E+00	1.0250E-01	**2.2039E+00**	**1.6459E+00**	1.0730E-01	**1.4590E+00**	5.3750E-01	3.9600E-02	6.0990E-01

Figure 2 and Table 3 shows that:

1) Comparison with the BLS: On the Housing dataset, IT3FBLS reduces RMSE by approximately 23.6% compared to the original BLS. The MAE decreases, while the R2 improves. These results demonstrate the advantages of IT3FBLS in capturing the complex nonlinear relationships and uncertainties present in housing price data.

2) Comparison with FNN and IT2FNN: IT3FBLS consistently outperforms both FNN and IT2FNN. On the Housing dataset, IT3FBLS achieves a lower RMSE than both FNN and IT2FNN, with corresponding improvements in MAE and R2. This suggests that leveraging a higher-order Type-3 fuzzy logic allows for more accurate representation of uncertainty, leading to enhanced predictive performance. Similarly, on the CCS and Abalone datasets, IT3FBLS maintains a lower RMSE and MAE while achieving higher R2. Specifically, performance improvements over FNN and IT2FNN range from 8%–20% in RMSE and MAE, while R2 increases by 2–6%, further demonstrating the effectiveness of Type-3 fuzzy structures in handling complex data.

3) Compared to FBLS and IT2FBLS: IT3FBLS ranks among the best-performing methods across all evaluation metrics. On the Abalone dataset, while FBLS and IT2FBLS achieve strong results, IT3FBLS further reduces RMSE and improves R2. IT3FBLS emerges as one of the top two methods in both RMSE and R2, indicating that the entropy-adaptive regularization and gradient-based secondary optimization mechanisms significantly enhance predictive accuracy and generalization ability.

4) Ablation study Analysis: IT3FBLS-1, which omits the proposed parameter learning algorithm, exhibits a decline in R2. IT3FBLS-2, which retains entropy-weighted regularization but removes GD-based secondary optimization, shows a similar performance degradation trend. These findings highlight the importance of each hyperparameter module working collaboratively within IT3FBLS. The complete model effectively captures data uncertainty and complexity, leading to consistently lower errors and higher goodness-of-fit across multiple datasets.

4.3.2 Uncertain Function dataset

For uncertain function modeling, the system dynamics are based on the Mackey-Glass equation and the Rössler attractor equation. The Mackey-Glass equation is formulated as:

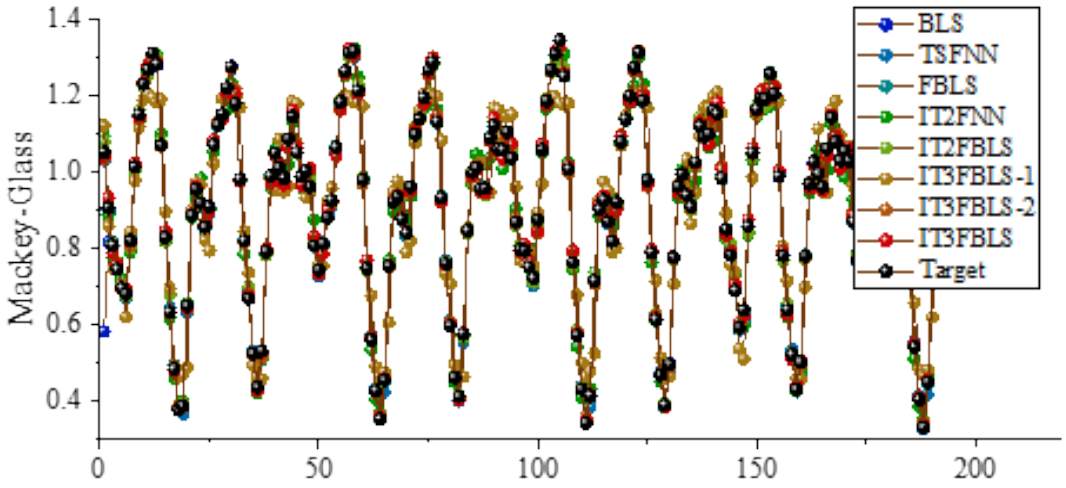
$$\dot{x}(t) = \frac{0.2 \cdot x(t-\tau)}{0.2 \cdot x^{10}(t-\tau)} - 0.1 \cdot x(t) \tag{49}$$

where $x(t)$is the system state, $x(0) = 1.1$, and $\tau = 20$ represents system delay time. The output variable is $y(t) = x(t)$, while the input variables consist of delayed states from multiple past time steps, including $x(t-1)$, $x(t-2)$, $x(t-6)$, and $x(t-12)$. This equation characterizes the complex nonlinear relationship between the current state $x(t)$ and historical states $x(t-\tau)$, where higher-order nonlinear terms induce various uncertain dynamic behaviors such as periodic solutions and chaotic solutions.

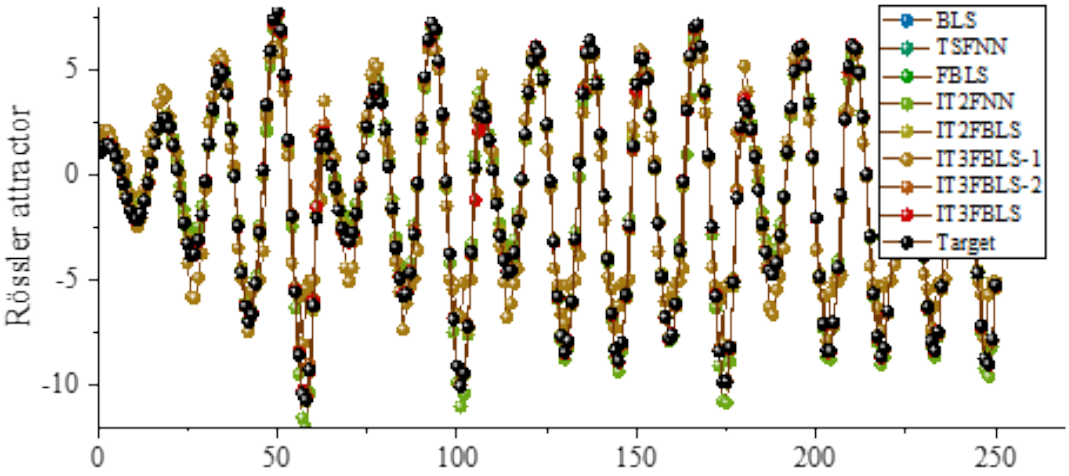
The Rossler system is defined by the following system of equations:

$$\begin{cases} \dot{x}_1(t) = -x_2(t) - x_3(t) \\ \dot{x}_2(t) = x_1(t) + a \cdot x_2(t) \\ \dot{x}_3(t) = b - c \cdot x_3(t) + x_1(t) \cdot x_2(t) \end{cases} \tag{50}$$

where $a = 0.1, b = 0.2, c = 5.7$, and the initial condition are $x_1(0), x_2(0), x_3(0) = 1,1,1$. The output variable is $y(t) = x_1(t), x_2(t), x_3(t)$. This model exhibits characteristic chaotic dynamics, where system states are highly dependent on historical states and exhibit strong nonlinear interactions. Due to the presence of nonlinear terms and coupling effects, even minor perturbations in initial conditions can lead to significant trajectory divergence, introducing uncertainty into data-driven modeling. The prediction results and comparison results of the testing set are shown in Fig. 3 and Table 4.



(a) Mackey-Glass



(b) Rossler system

Figure 3: Prediction results of the testing set of uncertain function dataset

Table 4: Performance comparison results of the testing set in the uncertain function dataset

Datasets	Methods	RMSE			MAE			R2		
		Mean	Var	Best	Mean	Var	Best	Mean	Var	Best
Mackey – Glass	BLS	2.9764E-01	3.9052E-01	3.3935E-02	2.2310E-02	2.7710E-02	3.7310E-03	2.5341E+01	9.2931E-01	9.8257E-01
	FNN	2.3121E-02	6.6905E-03	1.2152E-02	1.8167E-02	4.8734E-03	9.7597E-03	9.9130E-01	4.7095E-03	9.9776E-01
	FBLS	**2.4116E-03**	**1.5833E-04**	**2.2007E-03**	**1.9114E-03**	**1.1725E-04**	**1.7442E-03**	**9.9861E-01**	**1.2254E-05**	**9.9913E-01**
	IT2FNN	4.3339E-02	8.2000E-03	3.0138E-02	3.2803E-02	5.5523E-03	2.2987E-02	9.7060E-01	1.1591E-02	9.8625E-01
	IT2FBLS	2.3594E-02	1.0772E-02	1.1484E-02	1.8609E-02	8.4352E-03	9.9888E-03	9.8991E-01	9.7716E-03	9.9800E-01
	IT3FBLS-1	8.7532E-02	4.6528E-03	7.9216E-02	7.0162E-02	4.2087E-03	6.2543E-02	8.8373E-01	1.2219E-02	9.0502E-01
	IT3FBLS-2	1.4820E-02	7.8595E-04	1.3528E-02	1.1452E-02	6.0698E-04	1.0386E-02	9.9667E-01	3.5497E-04	9.9723E-01

	IT3FBLS	**1.3573E-02**	**3.8224E-04**	**1.2991E-02**	**1.0227E-02**	**2.7178E-04**	**9.8179E-03**	**9.9721E-01**	**1.5743E-04**	**9.9845E-01**
Rossler system	BLS	5.1297E-01	**3.2984E-03**	4.1499E-01	6.4590E-02	**2.0603E-04**	**3.6554E-04**	9.4233E+01	**1.7868E-04**	9.7453E+00
	FNN	1.4850E-01	2.8354E-01	1.8088E-01	5.9146E-02	8.9973E-02	5.1575E-03	9.9580E-01	1.2641E-02	9.9129E-01
	FBLS	**1.7019E-03**	**3.1604E-03**	**1.5816E-06**	**7.4489E-04**	2.3595E-03	**1.0278E-06**	**9.9911E-01**	**1.5678E-04**	**9.9929E-01**
	IT2FNN	1.1028E+00	2.1280E-01	5.1417E-01	7.4501E-01	1.2732E-01	3.7827E-01	9.4394E-01	2.0560E-02	9.8823E-01
	IT2FBLS	5.7999E-01	6.0014E-01	1.6707E-01	3.4652E-01	3.9376E-01	8.3696E-02	9.6979E-01	7.5461E-02	9.9876E-01
	IT3FBLS-1	1.7675E+00	2.2979E-01	1.5419E+00	1.4358E+00	2.5266E-01	1.1590E+00	8.5878E-01	3.7874E-02	8.9415E-01
	IT3FBLS-2	2.0261E-01	2.2748E-02	1.7888E-01	7.7464E-02	**2.2375E-03**	6.3005E-02	**9.9815E-01**	4.3197E-04	9.9858E-01
	IT3FBLS	**2.0795E-01**	5.9879E-02	**1.2618E-01**	**5. 3515E-02**	1.0391E-02	4.4466E-02	9.9793E-01	1.2413E-03	**9.9929E-01**

Figure 3 and Table 4 show that:

1) Comparison with BLS: The BLS, due to its lack of a fuzzy mechanism, has limited nonlinear modeling capability, resulting in higher prediction errors. In contrast, the IT3F-BLS, which integrates a T3FS, demonstrates significant performance improvements in uncertain function modeling tasks. In the Mackey-Glass chaotic sequence modeling task, the average RMSE of IT3FBLS is reduced by approximately 95% compared to BLS, with the MAE decreased by 54%. Additionally, R^2 significantly increased from 0.25 for BLS to over 0.997 for IT3FBLS. In the Rössler system modeling task, the average RMSE and MAE of IT3FBLS are reduced by 59.46% and 17.14%, respectively, with R^2 improving by 0.78. This improvement mainly stems from the effective representation of input-output uncertainties by the third-order fuzzy sets. By deeply integrating the fuzzy feature layer with the BLS broad learning architecture, IT3FBLS achieves more precise mathematical modeling of complex nonlinear relationships.

2) Comparison with FNN: IT3FBLS exhibits significant advantages on the Mackey-Glass dataset: the RMSE and MAE are reduced by 41% and 44%, respectively, while R^2 increases from 0.9913 to 0.9972. Compared to IT2FNN, the reduction in RMSE and MAE reaches 69%. Notably, in Rössler system modeling, even though FNN has achieved high precision ($R^2 > 0.99$), IT3FBLS still demonstrates stronger adaptability. Compared to IT2FNN, its RMSE and MAE are reduced by 81% and 92.8%, respectively. Overall, IT3F-BLS outperforms FNN in most cases, especially in the Mackey-Glass task, showcasing the superiority of higher-order fuzzy modeling in handling complex dynamic systems. This indicates that the T3FS, when integrated with BLS, can fit complex mapping relationships more accurately than T1FS and T2FS represented by FNN and IT2FNN. The introduction of third-order fuzziness enhances the model's ability to represent uncertainty, leading to significantly better performance on these datasets than neural networks using only T2FS.

3) Comparison with IT2FNN: IT3FBLS combines the efficient training mechanism of broad learning with the strong uncertainty representation capability of the T3FS. This not only overcomes the limitations of T1FS in modeling capability but also addresses the shortcomings of T2FS in handling higher-level uncertainty. The third-order fuzziness, by introducing a degree of membership to the membership degree, effectively describes the credibility range of fuzzy rules, allowing the model to maintain high precision when facing complex noise or unmodeled dynamics. The introduction of BLS provides the ability to quickly expand feature spaces and efficiently solve linear equations, significantly improving the training efficiency while maintaining high accuracy.

4) Comparison to FBLS and IT2FBLS: In the Mackey-Glass system, the average RMSE of IT3FBLS is slightly higher than that of FBLS, but its optimal RMSE is close to FBLS, and the optimal R^2 value is slightly better than FBLS. While FBLS outperforms IT3FBLS in terms of average performance, their R^2 variances are similar, indicating that both models exhibit comparable stability. However, IT3FBLS demonstrates superior generalization ability in more complex scenarios. In the Rössler system, IT3FBLS reduces

the average RMSE by 64.15% and the MAE by 84.55% compared to IT2FBLS, with the optimal R^2 value approaching the theoretical limit. Given that IT3FBLS introduces a more complex third-order fuzzy uncertainty modeling approach, its advantages are expected to be even more pronounced in environments with noise or higher uncertainty. Therefore, IT3FBLS can be considered to have achieved performance levels comparable to the most advanced methods, while offering enhanced generalization ability. By more effectively modeling uncertainty, IT3FBLS retains the advantages of ensemble learning and significantly improves predictive accuracy. It combines the ensemble learning characteristics of FBLS with the powerful representational capabilities of T3FS, thus outperforming earlier IT2FBLS models. Overall, the key improvement of IT3FBLS over existing ensemble models lies in its handling of third-order fuzzy uncertainty. FBLS, which uses first-order fuzzy sets, achieves high accuracy but does not explicitly account for the uncertainty in membership degrees. IT2FBLS introduces uncertainty modeling for membership degree intervals but may increase model complexity and optimization difficulty. In contrast, IT3FBLS further refines the credibility of the membership function itself through third-order fuzzy sets, offering a more granular uncertainty representation. As a result, IT3FBLS has a potential advantage in tasks where generalization ability and noise robustness are critical.

4) Ablation experiment analysis: In IT3FBLS-1, its performance significantly declines. In the Mackey-Glass system, the average RMSE of IT3FBLS-1 increases approximately 6.5 times compared to the full model, with R^2 decreasing by about 11%. In the Rössler system, the impact is even more pronounced, with RMSE increasing by more than 750% and R^2 dropping nearly 14%. These results clearly indicate that removing the entropy-based parameter learning algorithm leads to a dramatic increase in error and a severe degradation in the model's ability to fit complex relationships. In IT3FBLS-2, Experimental results show that in the Mackey-Glass system, the RMSE of IT3FBLS-2 is slightly higher than the full model, while in the Rössler system, the average MAE of IT3FBLS-2 increases by 44.7%. This demonstrates that the gradient-based second-order optimization algorithm plays a significant role in suppressing parameter oscillations and improving model stability.

4.3.3 Complex Industrial Process Dataset

In this article, the modeling of two complex industrial processes, the MSWI process and the TE process, is explored. The description of the MSWI and TE chemical processes and their flow chart are given out as follows.

The MSWI process consists of six stages: solid waste storage and transportation, solid waste combustion, waste heat recovery, steam generation, flue gas treatment, and flue gas emission. Among these, the stability and efficiency of the solid waste combustion stage are directly linked to the overall stable operation of the plant and pollutant emissions [38]. Its process structure is shown in the Figure 4.

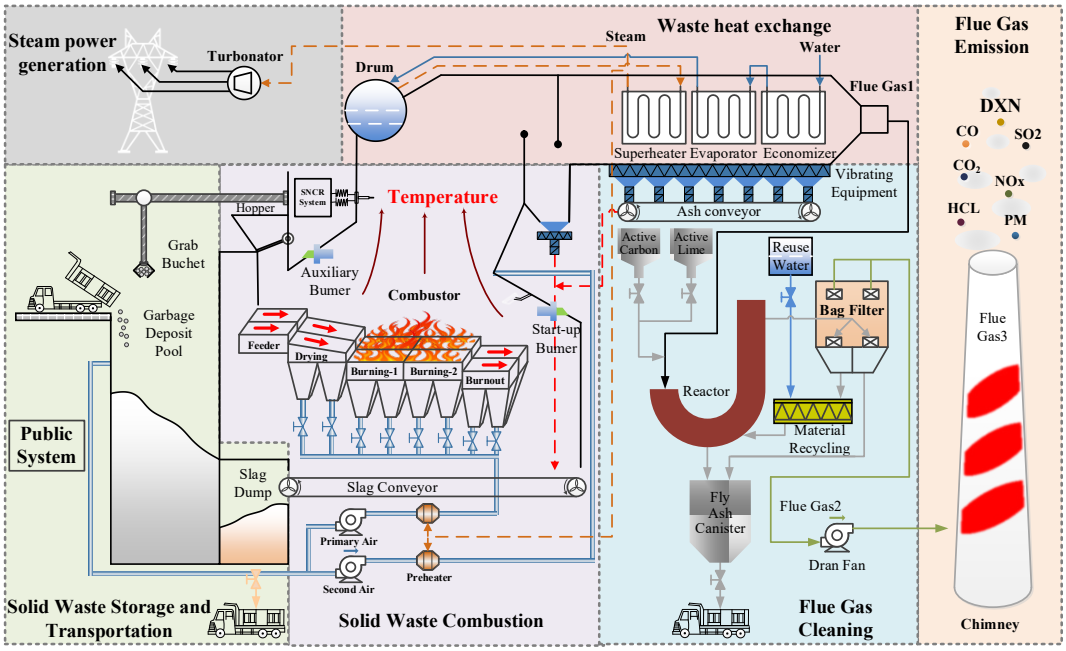


Figure 4: MSWI process flow diagram

The FT is crucial for the combustion efficiency and stability of this stage, necessitating precise control. There are significant differences in the characteristics of MSW between developing countries, such as China, and developed countries, such as those in Europe and America. These differences often lead to the use of manual operation modes in many developing countries. In the manual mode, domain experts typically rely on their human brain models to predict the trends of key controlled variables, and then determine the output values of manipulated variables

Tennessee Eastman (TE) process is a complex chemical process with strong variable coupling and nonlinearity [ ]. The TE process was developed by Eastman Chemical Company in the United States to provide a realistic industrial process for evaluating fault detection and variable prediction methods. Its process structure is shown in the Figure 5.

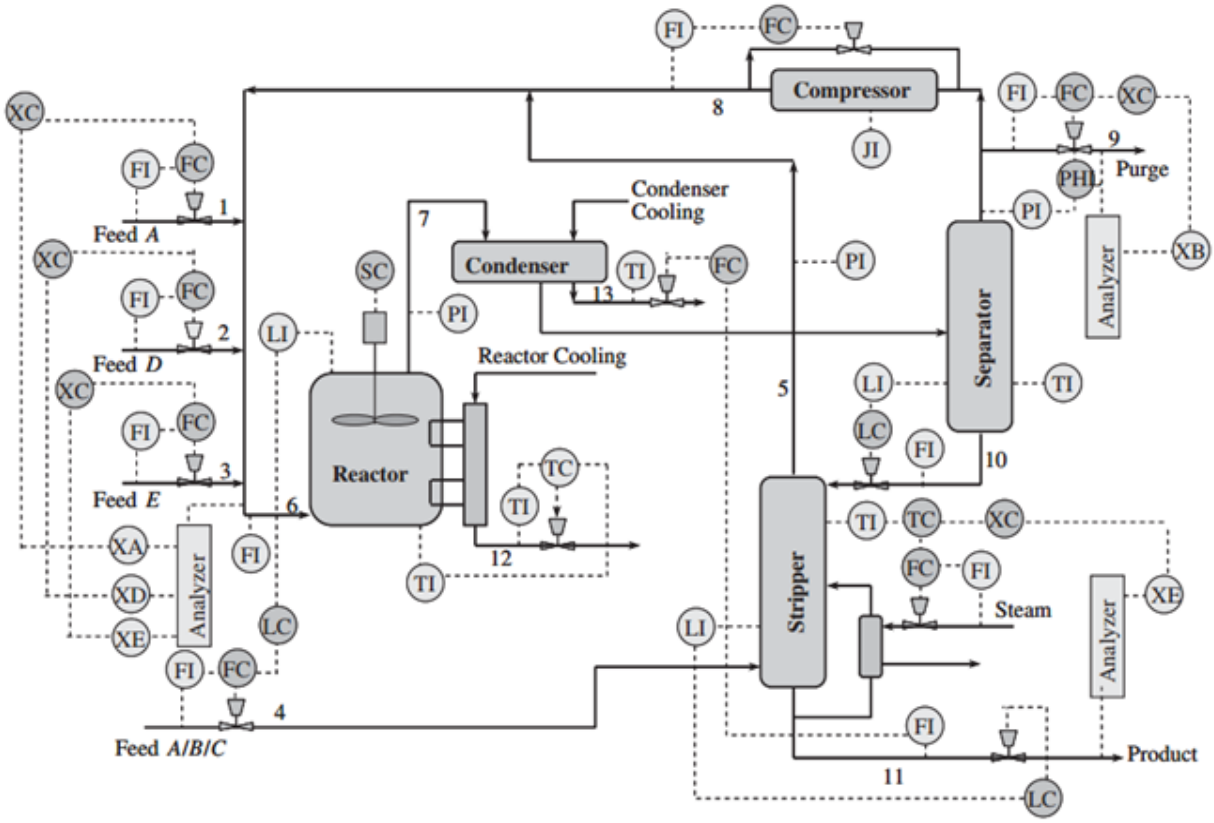


Figure 5: Structure diagram of Tennessee Eastman process

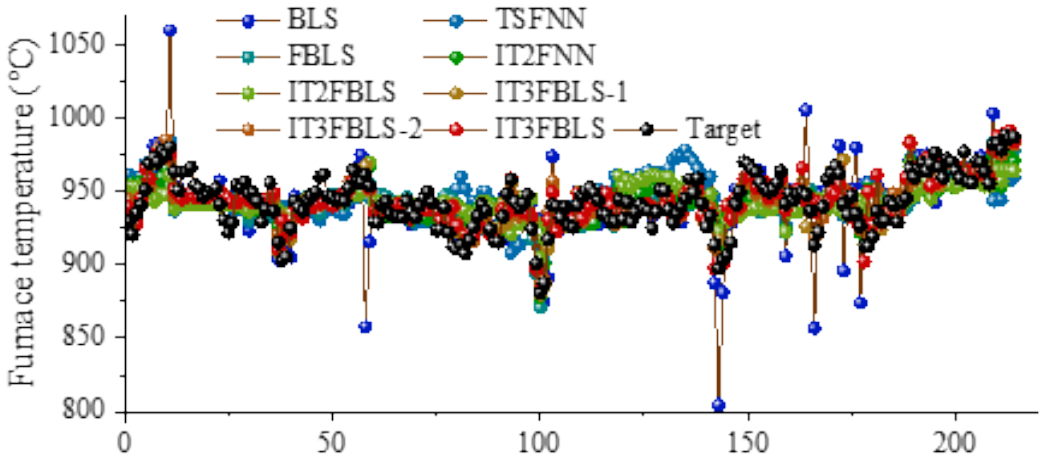
As can be seen in Fig. G1, the TE process contains eight constituent variables: A, B, C, D, E, F, G, and H. Of these, A, C, D, and E are reactants, B is an inert, G and H are the main products generated by the reactor, and F is a by-product. The dataset selected for this article has a sampling interval of 3mins and contains a total of seven feature variables, namely: A feed (stream 1), D feed (stream 2), E feed (stream 3), total feed (stream 4), recycle flow (stream 8) and E component (stream 11) [40].

In the MSWI process, the furnace temperature is a critical variable influencing combustion efficiency, pollutant emissions, and system stability. However, due to the complex composition of waste, uneven combustion characteristics, and external environmental disturbances, the furnace temperature exhibits significant nonlinearity, time-varying behavior, and uncertainty. Therefore, accurately modeling furnace temperature not only aids in optimizing incineration control and improving energy recovery efficiency but also contributes to the effective reduction of pollutants such as dioxins (DXN) and nitrogen oxides (NOx).

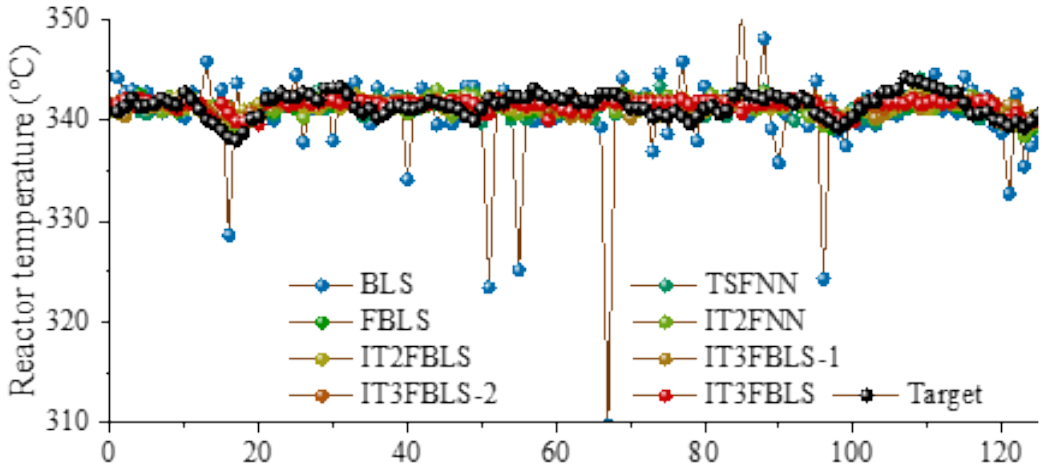
In the TE process, a typical industrial chemical process, involves a reactor where temperature is a key variable determining the reaction rate and product quality. As the system includes complex heat exchange and reaction dynamics, its dynamic behavior also presents significant nonlinearity and uncertainty. The sources of uncertainty, sampling intervals, and input/output variables for both the MSWI and TE processes are summarized in Table 5.

Table 5: Uncertainty sources, sampling intervals, and process variable descriptions for the MSWI process and the TE process

	Source of uncertainty	Sampling interval (min)	Input variable	Output variable
MSWI process	Fuel composition fluctuations, air supply, combustion dynamics, strong nonlinear combustion, turbulent heat transfer, operating condition fluctuations	1	Primary air volume, secondary air volume, average feeder speed, average drying grate speed, ammonia injection volume	Furnace temperature
TE process	Reaction kinetics, heat exchange, material transport, multi-step chemical reactions, thermodynamics, fluid transport, transport failures and abnormal conditions	3	D feed rate (stream 2), E feed rate (stream 3), A feed rate (stream 1), total feed rate (stream 4), compressor recirculation valve, discharge valve (stream 9), separator tank liquid flow (stream 10), stripper liquid product flow (stream 11), stripper water flow valve, reactor cooling water flow, condenser cooling water flow	Reactor temperature



(a)MSWI Process



(b) TE process

Figure 6: Prediction results of the testing set of complex industrial process data sets

Table 6: Performance comparison results of the testing set in complex industrial process data sets

Datasets	Methods	RMSE			MAE			R2		
		Mean	Var	Best	Mean	Var	Best	Mean	Var	Best
MSWI	BLS	4.2869E+01	1.9537E+01	1.8329E+01	1.5747E+01	3.7545E+00	1.1383E+01	5.0442E+00	5.3205E+00	7.7202E-02
	FNN	1.8251E+01	**1.2451E-01**	1.8123E+01	1.4488E+01	1.4933E-01	1.4272E+01	8.4970E-02	1.2505E-02	9.7762E-02
	FBLS	1.3577E+01	5.2206E-01	1.2591E+01	1.0832E+01	4.4640E-01	1.0033E+01	4.9293E-01	3.8997E-02	5.6452E-01
	IT2FNN	1.4292E+01	3.6553E-01	1.3677E+01	1.1703E+01	3.7454E-01	1.0961E+01	4.3853E-01	2.8698E-02	4.8619E-01
	IT2FBLS	1.6738E+01	4.4930E-01	1.6093E+01	1.3856E+01	2.8086E-01	1.3419E+01	2.2992E-01	4.1974E-02	2.8854E-01
	IT3FBLS-1	1.0180E+01	1.6538E-01	9.9034E+00	7.7655E+00	8.6324E-02	7.5849E+00	7.1528E-01	**9.2605E-03**	7.3059E-01
	IT3FBLS-2	**9.9189E+00**	**1.5445E-01**	**9.7150E+00**	**7.5013E+00**	**1.3604E-01**	**7.2075E+00**	**7.2968E-01**	**8.4652E-03**	**7.4074E-01**
	IT3FBLS	**9.1893E+00**	1.7770E-01	**8.6750E+00**	**7.2749E+00**	**1.8865E-01**	**7.0219E+00**	**7.4554E-01**	2.0913E-02	**7.6339E-01**
TE process	BLS	5.9468E+01	1.2131E+02	4.7453E+00	9.2506E+00	1.1582E+01	2.5239E+00	-1.1923E+04	4.4483E+04	-1.4329E+01
	FNN	1.0766E+00	6.7992E-02	9.9546E-01	**8.5259E-01**	5.8838E-02	7.9559E-01	**2.0840E-01**	1.0168E-01	3.2542E-01
	FBLS	1.1011E+00	4.5375E-02	1.0186E+00	8.7610E-01	**2.7352E-02**	8.2791E-01	1.7336E-01	6.7758E-02	2.9376E-01
	IT2FNN	**1.0602E+00**	6.8572E-02	9.7473E-01	**8.3604E-01**	5.4041E-02	**7.6247E-01**	**2.3175E-01**	1.0123E-01	**3.5323E-01**
	IT2FBLS	1.1221E+00	6.5875E-02	**9.8712E-01**	9.1175E-01	5.2059E-02	8.1825E-01	1.4005E-01	9.9557E-02	3.3667E-01
	IT3FBLS-1	1.1244E+00	**4.1835E-02**	1.0725E+00	9.2102E-01	**3.6988E-02**	8.7056E-01	1.3816E-01	**6.4728E-02**	2.1698E-01
	IT3FBLS-2	**1.0650E+00**	**4.2266E-02**	1.0090E+00	8.7492E-01	3.8107E-02	8.1693E-01	1.8214E-01	**6.2344E-02**	3.0688E-01
	IT3FBLS	1.1102E+00	8.1063E-02	**9.4379E-01**	8.8026E-01	5.8782E-02	**7.4087E-01**	1.5670E-01	1.2226E-01	**3.9363E-01**

Figure 6 and Table 6 show that:

1) Comparison with the BLS: In the MSWI process, the standard BLS performs poorly on the MSWI furnace temperature modeling. Its mean RMSE far higher than IT3F-BLS. This represents a 78.6% reduction in RMSE with IT3F-BLS. Likewise, MAE drops about 53.8% lower. BLS's R^2 is essentially zero, indicating almost no predictive power, whereas IT3FBLS boosts R^2 to 0.7455 on average. In fact, the best run of IT3FBLS reaches an R^2 of 0.7634, whereas BLS never exceeds 0.08. This huge gain highlights that IT3FBLS can capture the furnace's complex dynamics and uncertainties that BLS misses. In more challenging TE process modeling, BLS proves ineffective in tracking system dynamics, while IT3FBLS demonstrates a clear advantage. Unlike BLS, IT3FBLS maintains lower RMSE and a positive R^2, showcasing superior modeling capabilities. Quantitative results reveal that IT3FBLS reduces RMSE by over 98% compared to BLS, successfully transforming a previously unusable model into a feasible model with moderate accuracy. Although the R^2 value of 0.1567 indicates that unexplained variability still exists in the TE reactor temperature, IT3FBLS at least provides a usable model, whereas BLS is entirely unsuitable for this task. This result highlights that the nonlinear modeling capability of IT3F-BLS in the TE process makes it more robust and applicable in complex industrial scenarios.

2) Comparison with FNN and the IT2FNN: Both the FNN

and the IT2FNN perform better than BLS on the MSWI problem, but they are outperformed by IT3FBLS. IT3FBLS yields roughly 50% lower RMSE than FNN and 36% lower RMSE than IT2FNN. In terms of R^2, IT3FBLS boosts the explained variance by a large margin (about 30% higher than IT2FNN). These results indicate that IT3FBLS's advanced fuzzy modeling captures the furnace temperature dynamics much more accurately than the simpler type-1 or type-2 fuzzy systems. The type-3 fuzzy logic in IT3FBLS provides a more expressive uncertainty representation, leading to better generalization in this complex process. On the TE process, the fuzzy neural approaches FNN and IT2FNN achieve decent accuracy. IT3FBLS's mean RMSE is slightly higher than IT2FNN by about 4.7%, and its R^2 is a bit lower than IT2FNN's. However, IT3FBLS achieved the highest best-case R^2 on TE. This suggests that although run-to-run variability affected its average, IT3FBLS has the capacity to learn a superior model of the reactor temperature under the right conditions. The slight shortfall in mean performance could be due to the TE process's characteristics. Nonetheless, IT3FBLS's ability to match the fuzzy neural networks on accuracy while providing a richer uncertainty handling framework is an encouraging sign for its robustness. Therefore, compared to classical fuzzy systems, IT3FBLS benefits from the BLS which can integrate features in parallel and solve weights efficiently, combined with type-3 fuzzy logic. The type-1 FNN lacks uncertainty modeling, and the IT2FNN handles only interval uncertainty in membership functions. IT3FBLS extends this to a higher-order fuzzy representation, capturing more complex and dynamic uncertainty.

3) Comparison to FBLS and IT2FBLS: FBLS and IT2FBLS are advanced ensemble models that combine fuzzy logic with BLS. While FBLS uses Type-1 fuzzy logic, IT2FBLS employs Type-2 fuzzy logic. In MSWI process, the FBLS and IT2FBLS models show improved accuracy over basic BLS and fuzzy networks alone. However, IT3FBLS outperforms both, with RMSE is 32% lower than FBLS and 45% lower than IT2FBLS. Correspondingly, IT3FBLS's R^2 higher than FBLS about 51% and more than triple IT2FBLS. Even the best-case RMSE of IT3FBLS is better than the best FBLS and IT2FBLS. These gains demonstrate that moving from type-1 and type-2 fuzzy logic to type-3 fuzzy logic within the broad learning ensemble yields a significant accuracy boost. The MSWI process likely involves uncertainties that type-3 fuzzy BLS handles more effectively, explaining the superior performance. In the TE process, the advanced ensemble models all perform similarly, with small differences. However, IT3FBLS stands out with best RMSE and best R^2 – considerably higher than FBLS's best or IT2FBLS's. This indicates that while IT3FBLS did not consistently dominate in every run, it has the most potential when conditions are right, likely due to its more expressive model. In practical terms, all ensemble fuzzy BLS models handle the TE reactor problem far better than plain BLS, but IT3FBLS provides the strongest overall capability, especially under varying conditions or if further optimized. In the TE process, when nonlinear factors dominate over uncertainty, the BLS framework is already well-equipped to handle the core complexities, leading to similar performance among all fuzzy logic-based BLS variants. Nevertheless, IT3FBLS does not compromise prediction accuracy and performs at least at the level of the best-case scenario. This suggests that IT3FBLS retains all the advantages of IT2FBLS while offering greater potential for improvement in handling uncertainty factors.

4) Ablation Experiment Analysis: The full IT3FBLS clearly outperforms its ablations on the MSWI task. The RMSE of IT3FBLS-1 and -2 are higher than full IT3FBLS. In terms of R^2, full IT3FBLS is about 2–4% higher in absolute terms. IT3FBLS reduces MAE by 3.2% compared to IT3FBLS-2 and 6.3% compared to IT3FBLS-1, and its best-case RMSE is lower than the best of -1 or -2. These differences, though not enormous, are consistent: IT3FBLS is better on every metric than either ablation. This indicates that each component removed in the ablations does play a role in boosting performance. The full model's superiority on the uncertain MSWI problem suggests that the combination of broad learning and type-3 fuzzy logic is synergistic – removing parts leads to a noticeable drop in accuracy. On the TE task, the ablation results are more mixed. IT3FBLS-2 had a slightly lower mean RMSE than full IT3FBLS and a higher R^2 on average. IT3FBLS-1 was a bit worse. However, as noted, IT3FBLS had the best optimal performance of the three. This suggests that the full IT3FBLS model has higher capacity or flexibility, which sometimes leads to better solutions, but might also introduce more variability. The higher variance in IT3FBLS's results on TE supports this interpretation – the full model is more complex and can occasionally overfit or get stuck in suboptimal solutions on this dataset. Nonetheless, the fact that IT3FBLS -1 and -2 don't consistently outperform the full model confirms that the removed components are generally beneficial. In a more uncertainty-dominated setting, those components likely prove crucial. For TE process, it appears the full IT3FBLS is at least as good as the best ablation on average, and it retains a higher ceiling for

performance. With further tuning or ensemble averaging, the full model could likely surpass the ablations outright even in TE.

Overall, the type-3 fuzzy component and the ensemble broad feature learning are complementary – the former improves model flexibility and uncertainty handling, while the latter ensures strong function approximation and generalization. The ablated models, lacking one of these, cannot consistently match the full IT3FBLS, especially in the face of complex, uncertain industrial data like MSWI process.

4.4 Hyperparameter Analysis

This section presents a sensitivity analysis of IT3FBLS, using the MSWI process dataset as an example. Detailed information about the hyperparameter settings is provided in Table 2.

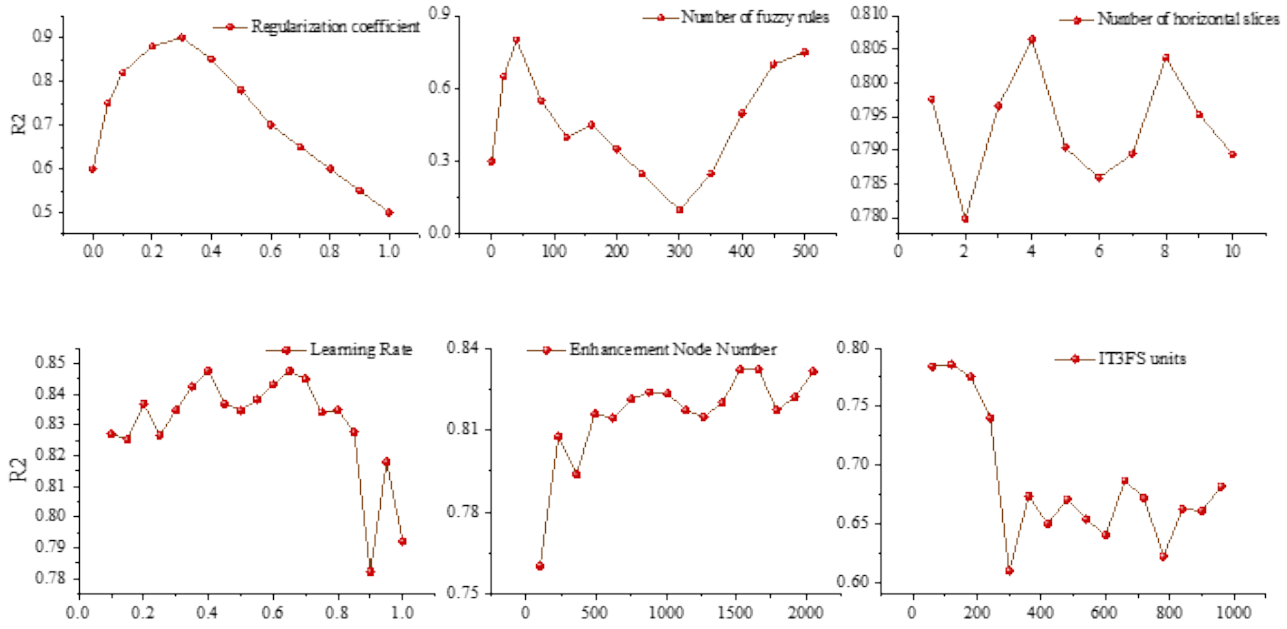


Figure 7: Parameter sensitivity of IT3FBLS

Fig. 7 shows the results as follows:

(1) The regularization coefficient λ significantly affects IT3FBLS performance. When λ within the range of [0, 0.3], R^2 shows a notable improvement, indicating that moderate regularization effectively mitigates overfitting. However, once λ exceeds 0.3, R^2 drops rapidly, suggesting that excessive regularization leads to underfitting.

(2) The number of fuzzy rules J significantly enhances R^2 within the [5, 20] range, but when the number exceeds 20, R^2 stabilizes, indicating diminishing returns in performance with an increasing number of rules. Furthermore, when J exceeds 80, it negatively impacts model performance;

(3) The number of horizontal slices K leads to the most significant improvement in R^2 within the [2, 4] range, but beyond 4, R^2 decreases slightly due to feature redundancy. The number of enhancement nodes L generally exhibits a positive correlation with model performance; however, when L exceeds 500, further improvements are marginal.

(4) The learning rate η shows a positive correlation with R^2 in the ranges [0.2, 0.4] and [0.45, 0.65], but when it exceeds 0.7, large step sizes in parameter updates may cause the model to get stuck in local optima, resulting in performance degradation.

(5) The value of number of IT3FS units M maintains high model performance only within the [10, 200] range. When exceeding 200, both model performance and computational time significantly degrade.

Therefore, λ and J are highly sensitive hyperparameters with the most significant impact on model performance. These should be prioritized for optimization, preferably through grid search or heuristic optimization. L and M also significantly influence model performance and are related to the computational complexity of the model. Therefore, it is recommended to set reasonable ranges based on domain knowledge to avoid unnecessary expansion. K and η to the model is lower sensitive, and their values can be dynamically adjusted through heuristic

strategies to balance performance and reduce computational costs.

5. Conclusion

A novel BLS model for uncertain data regression, termed IT3FBLS, is proposed in this article. To overcome the limitations of traditional fuzzy systems in high-order uncertainty representation and real-time learning efficiency, IT3FS is integrated with the BLS framework. This integration facilitates a refined characterization of multi-level uncertainties while simplifying the complexity of model parameter learning. Additionally, to address overfitting and enhance generalization capabilities in noisy or non-stationary environments, an entropy-based weighted adaptive regularization and gradient-based quadratic optimization parameter learning strategy is introduced in IT3FBLS. This approach dynamically quantifies node distribution and feature contribution, thereby improving robustness to noise and uncertainty. A systematic comparison of IT3FBLS with several conventional methods in the context of uncertainty regression modeling is conducted. Experiments are performed on datasets of varying scales and dimensions, uncertain functions, and real-world industrial process data. The results demonstrate that IT3FBLS achieves superior model accuracy and stability with a limited number of training samples. Moreover, it exhibits lower dependence on hyperparameters when handling multi-source heterogeneous data and high-dimensional noise, showcasing its inherent robustness and adaptability. Notably, IT3FBLS maintains strong performance across diverse scenarios, remaining insensitive to changes in data dimension and scale. These findings underscore the method's potential for high-dynamic industrial processes and other uncertainty-driven applications.

Future research focuses on further reducing the computational overhead of IT3FBLS while enhancing its efficiency and stability in practical industrial applications, such as real-time control and online modeling. Additionally, incorporating advanced ensemble learning mechanisms, parameter learning, and incremental learning methods into the IT3FBLS framework is to be essential for addressing challenges such as distribution drift and multi-task collaboration in dynamic environments.

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 article.

Data Availability

The dataset is available upon request from the corresponding

Credit Authorship Contribution Statement

Hao Tian: Methodology, Software, Validation, Data Curation, Writing - Original Draft.

References

[1] Y. Chen, C. Wang, Y. Zhou, Y. Zuo, Z. Yang, H. Li, and J. Yang. Research on multi-source heterogeneous big data fusion method based on feature level. International Journal of Pattern Recognition and Artificial Intelligence, vol. 38, no. 02 (2024): 2455001.

[2] W. Shao, C. Xiao, J. Wang, D. Zhao, and Z. Song. Real-time estimation of quality-related variable for dynamic and non-Gaussian process based on semisupervised Bayesian HMM. Journal of Process Control, vol. 111 (2022): 59-74.

[3] W. Yu, C. Y. Wong, R. Chavez, and M. A. Jacobs. Integrating big data analytics into supply chain finance: The roles of information processing and data-driven culture. International journal of production economics, vol. 236 (2021): 108135.

[4] A. M. Khedr, I. Arif, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. Intelligent Systems in Accounting, Finance and Management, vol. 28, no. 1 (2021): 3-34.

[5] I. H. Sarkr. AI-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems. SN computer science, vol. 3, no. 2 (2022): 158.

[6] C. Li, Y. Chen, and Y. Shang. A review of industrial big data for decision making in intelligent manufacturing. Engineering Science and Technology, an International Journal, vol. 29 (2022): 101021.

[7] J. M. Mendel. Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE, vol. 83, no. 3 (1995): 345-377.

[8] J. Fei, Z. Wang, X. Liang, Z. Feng, and Y. Xue. Fractional sliding-mode control for microgyroscope based on multilayer recurrent fuzzy neural network. IEEE transactions on fuzzy systems, vol. 30, no. 6 (2021): 1712-1721.

[9] H. Ding, J. Qiao, W. Huang, and T. Yu. Cooperative event-triggered fuzzy-neural multivariable control with multitask learning for municipal solid waste incineration process. IEEE Transactions on Industrial Informatics, vol. 20, no. 1 (2023): 765-774.

[10] J. Y. Ho, J. Ooi, Y. K. Wan, and V. Andiappan. Synthesis of wastewater treatment process (WWTP) and supplier selection via Fuzzy Analytic Hierarchy Process (FAHP). Journal of Cleaner Production, vol. 314 (2021): 128104.

[11] N. N. Karnik, J. M. Mendel, and Q. Liang. Type-2 fuzzy logic systems. IEEE transactions on Fuzzy Systems, vol. 7, no. 6 (1999): 643-658.

[12] Q. Liang, and J. M. Mendel. Interval type-2 fuzzy logic systems: theory and design. IEEE Transactions on Fuzzy systems, vol. 8, no. 5, (2000): 535-550.

[13] H. Tian, J. Tang, H. Xia, W. Yu, and J. Qiao. Bayesian optimization-based interval type-2 fuzzy neural network for furnace temperature control. IEEE Transactions on Industrial Informatics, vol. 21, no. 1, (2025): 505-514.

[14] O. Castillo, Towards finding the optimal in designing type-n fuzzy systems for particular classes of problems: A review, Appl. Comput. Math., vol. 17, no. 1, (2018): 3–9.

[15] Y. Chen and D. Wang, Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted Nie–Tan algorithms, Soft Comput., vol. 22, no. 22, (2018): 7659–7678.

[16] J. R. Castro, M. A. Sanchez, C. I. Gonzalez, P. Melin, and O. Castillo, A new method for parameterization of general type-2 fuzzy sets, Fuzzy Inf. Eng., vol. 10, no. 1, (2018): 31–57.

[17] P. Melin and D. Sánchez, Optimization of type-1, interval type-2 and general type-2 fuzzy inference systems using a hierarchical genetic algorithm for modular granular neural networks, Granular Comput., vol. 4, (2019): 211–236.

[18] A. Mohammadzadeh, M. H. Sabzalian, and W. Zhang. An interval type-3 fuzzy system and a new online fractional-order learning algorithm: theory and practice. IEEE Transactions on Fuzzy Systems, vol. 28, no. 9 (2019): 1940-1950.

[19] Z. Liu , A. Mohammadzadeh, H. Turabieh , M. Mafarja, S. S. Band , A. Mosavi. A new online learned interval type-3 fuzzy control system for solar energy management systems. IEEE Access, vol. 9, (2021): 10498-10508.

[20] L. Amador-Angulo, O J. R. Castro , P. Melin. A new approach for interval type-3 fuzzy control of nonlinear plants. International Journal of Fuzzy Systems, vo. 25, no. 4, (2023): 1624-1642.

[21] P. Ochoa, O. Castillo, P. Melin, J. R. Castro. Interval type-3 fuzzy differential evolution for parameterization of fuzzy controllers. International Journal of Fuzzy Systems, vol. 25m no. 4, (2023): 1360-1376.

[22] D. J. Singh, N. K. Verma, and A. Ki Ghosh, Appasaheb Malagaudanavar. An approach towards the design of interval type-3 T–S fuzzy system. IEEE Transactions on Fuzzy Systems, vol. 30, no. 9, (2021): 3880-3893.

[23] C. L. P. hen , and Z. Liu. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. IEEE transactions on neural

networks and learning systems, vol. 29, no .1, (2017): 10-24.
[24] X. Gong, T. Zhang, C. L. P Chen, and Z. Liu. Research review for broad learning system: Algorithms, theory, and applications. IEEE Transactions on Cybernetics, vol. 52, no. 9, (2021): 8922-8950.
[25] S. Feng, and C. L. P. Chen. Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification. IEEE transactions on cybernetics, vol. 50, no. 2, (2018): 414-424.
[26] Y. Zhou, Q. Zhang, H. Wang, P. Zhou, and T. Chai, EKF-based enhanced performance controller design for nonlinear stochastic systems, IEEE Trans. Autom. Control, vol. 63, no. 4, (2028): 1155–1162.
[27] L. Yin and L. Guo, Fault isolation for multivariate nonlinear nonGaussian systems using generalized entropy optimization principle, Automatica, vol. 45, no. 11, (2009): 2612–2619.
[28] H. Tang, P. Dong, and Y. Shi, A construction of robust representations for small data sets using broad learning system, IEEE Trans. Syst., Man, Cybern. Syst., vol. 51, no. 10, (2019): 6074–6084.
[29] C. L. Blake and C. J. Merz. (1998). UCI Repository of Machine Learning Databases. Dept. Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html
[30] G. Leon, and M. Mackey. Mackey-glass equation. Scholarpedia, vol. 5, no. 3, (2010): 6908.
[31] R. Marat, and J. M. Balthazar. On an optimal control design for Rössler system. Physics Letters A, vol. 333, no. 3-4, (2004): 241-245.
[32] H. Tian , J. Tang, and T. Wang. Furnace temperature model predictive control based on particle swarm rolling optimization for municipal solid waste incineration. Sustainability, vol. 16, no. 17, (2024): 7670.
[33] B. Andreas, N. L. Ricker, and M. Jelali. Revision of the Tennessee Eastman process model. IFAC-PapersOnLine, vol. 48, no. 8, (2015): 309-314.
[34] D. C. Souza, P. Vitor. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. Applied soft computing, vol. 92, (2020): 106275.
[35] S. Feng , C. L. P. Chen, L. Xu, and Z. Liu. On the accuracy–complexity tradeoff of fuzzy broad learning system. IEEE Transactions on Fuzzy Systems, vol. 29, no. 10, (2020): 2963-2974.
[36] Z. Sun, X. Meng, and J. Qiao. An IT2FNN with a Novel Hierarchical Learning Algorithm for Nonlinear System Modeling. In 2023 China Automation Congress (CAC), pp. 2848-2853. IEEE, 2023.
[37] H. Han, H Liu, J Qiao, and C. L. P. Chen. Type-2 fuzzy broad learning system. IEEE Transactions on Cybernetics, vol. 52, no. 10, (2021): 10352-10363.
[38] J. Tang, H. Xia, W. Yu, and J. Qiao, Research status and prospects of intelligent optimization control for municipal solid waste incineration process, Acta Anat. Sin., vol. 49, no. 10, (2023): 2019-2059.
[39] X. Gao, and J. Hou, An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process, Neurocomputing, vol. 174, part. B, (2026): 906-911.
[40] Y. W. Zhang, Y. D. Teng, and Y. Zhang, Complex process quality prediction using modified kernel partial least squares, Chem. Eng. Sci., vol. 65, no. 6, (2010): 2153-2158.