IT2-ENFIS: Interval Type-2 Exclusionary Neuro-Fuzzy Inference System, an Attempt Toward Trustworthy Regression Learning

Chuan Xue, Jianli Gao, and Zhou Gu, *Senior Member, IEEE*

***Abstract*—As machine learning technologies progress and are increasingly applied to critical and sensitive fields, the reliability issues of earlier technologies are becoming more evident. For the new generation of machine learning solutions, trustworthiness frequently takes precedence over performance when evaluating their applicability for specific applications. This manuscript introduces the IT2-ENFIS neuro-fuzzy model, a robust and trustworthy single-network solution specifically designed for data regression tasks affected by substantial label noise and outliers. The primary architecture applies interval type-2 fuzzy logic and the Sugeno inference engine. A meta-heuristic gradient-based optimizer (GBO), the Huber loss function, and the Cauchy M-estimator are employed for robust learning. IT2-ENFIS demonstrates superior performance on noise-contaminated datasets and excels in real-world scenarios, with excellent generalization capability and interpretability.**

***Impact Statement*—Current machine learning algorithms often struggle to handle the complexities of real-world situations, raising concerns about the trustworthiness of their outputs. This study presents the IT2-ENFIS model, designed for predictive modeling tasks, which establishes trustworthiness through four key perspectives: robustness, generalization, interpretability, and fairness. Building on the excellent generalization capability and interpretability of interval type-2 fuzzy neural networks, it endows the model with a degree of autonomy in discerning erroneous information, thereby elevating its trustworthiness to a new level. This research has the potential to broaden the application scenarios of machine learning methodologies, enhancing adaptability in high-noise environments and situations with poor data quality, while also contributing valuable insights to the field.**

***Index Terms*—Data-driven modeling, interval type-2 inference systems, neuro-fuzzy systems, trustworthy machine learning.**

Received 2 March 2025; revised 12 April 2025 and 12 May 2025; accepted 25 May 2025. Date of publication 27 May 2025; date of current version 31 December 2025. This work was supported by the National Natural Science Foundation of China under Grant 62273183. This article was recommended for publication by Associate Editor Teena Sharma upon evaluation of the reviewers' comments. *(Corresponding author: Zhou Gu.)*

Chuan Xue is with the School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213164, China (e-mail: cxue12@cczu.edu.cn).

Jianli Gao is with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: jianli.gao18@imperial.ac.uk).

Zhou Gu is with the School of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China (e-mail: gzh1808@163.com).

Digital Object Identifier 10.1109/TAI.2025.3574299

I. Introduction

The expeditious development of artificial intelligence (AI) is driving a profound technological revolution, impacting our daily lives fundamentally. McKinsey, a renowned management consulting firm, estimated in 2023 that AI could contribute economic growth equivalent to the UK's GDP to the global economy annually and automate 60% to 70% of current jobs in the foreseeable future [1]. However, as AI adoption becomes widespread, the limitations of existing technologies become evident, particularly their inability to adapt to complex data conditions. Conventional machine learning algorithms trained under the empirical risk minimization (ERM) principle [2] are designed for idealized data environments, which often clash with the chaotic nature of real-world scenarios. Data uncertainties, such as unobserved confounders [3], data bias [4], and spurious features [5], are pervasive even in meticulously selected high-quality datasets. Statistical correlations between misleading features and labels can easily misguide these algorithms, leading to a range of adverse outcomes, from performance degradation to erroneous model outputs. Even the most advanced large language models are not immune to this plague. For instance, within three months of ChatGPT-4's public release, its capabilities in mathematics, problem shooting, coding, and visual reasoning notably declined, with the success rate in identifying prime numbers plummeting from 97.6% to 2.4% [6]. Although the reasons for the decline in model performance may be multifaceted, it is clear that uncertainties in the data environment play a significant role and present formidable challenges to machine learning applications.

In recent years, the reliability of machine learning has surfaced as a critical concern within the discipline. Concerted efforts have been made to address these challenges across various applications and research domains, resulting in a diverse array of methods designated to tackle each unique issue. All these methods fall under the framework of trustworthy machine learning [7], where "trustworthiness" generally encompasses four key metrics: the model's ability to generalize, its robustness (security), interpretability, and fairness [8]. Specifically, generalization refers to the capability to extract knowledge from limited training data and extrapolate it to unseen data [9]. Robustness denotes the resistance to errors, erroneous inputs, and even adversarial attacks [10], making it essential for machine learning systems operating in empirical environments.

Insufficient robustness can lead to unintended or harmful behavior, thereby jeopardizing the safety and integrity of the model [11]. Interpretability consists of two parts: explainability [12], which focuses on understanding how a model derives its results, and transparency [13], which seeks to provide insight into the model's entire lifecycle. Fairness, as its name suggests, refers to the system's capacity to mitigate biases present in the task, including data bias, model bias, and procedural bias [14]. Note that the trustworthiness of a model is flexible and contingent on specific contexts and should not be confined to a rigid, one-size-fits-all definition. Before delving deeper into this topic, it is essential to clarify which aspects of the model and under what circumstances should users expect it to be trustworthy.

Inspired by profound insights from prior research, this article proposes the innovative Interval Type-2 Exclusionary Neuro-Fuzzy Inference System (IT2-ENFIS) to enhance model trustworthiness in targeted tasks, where the term "exclusionary" denotes the removal of interference caused by data anomalies during training. This work has four main contributions in terms of methodology.

1) It presents the first trustworthy machine learning system designed based on interval type-2 fuzzy logic, utilizing its excellent interpretability and generalization capability to achieve the goal of trustworthy learning.
2) A metaheuristic optimizer, known as the gradient-based optimizer (GBO) [15], is adopted. GBO exhibits reduced sensitivity to fluctuations, mitigating issues like failure to converge or convergence to local optima caused by noisy labels. Also, it does not require closed-form first-order derivatives, which simplifies the optimization process.
3) Given that noise-contaminated labels cannot correctly reflect training errors, the Huber loss function [16] is introduced to suppress label noise during the iterative training process.
4) The Cauchy influence function [17] is utilized to assess the reliability of training samples. It can identify outliers that exert a significant impact on the model and neutralize their effect by assigning lower weights to deviating data points and higher weights to samples without notable anomalies.

Interval type-2 fuzzy systems offer both generalization and interpretability, while the aforementioned contributions further enable the model to achieve noise robustness in data regression tasks. As a result, the proposed model can meet the first three key requirements of trustworthy learning. The fourth requirement of trustworthiness, i.e., model fairness, is closely linked to the ethical implications of deploying AI models in sensitive domains. Fairness is also an intriguing topic due to its relatively subjective nature. Model fairness in regression tasks primarily focuses on whether prediction errors across different groups show systematic differences or whether certain sensitive variables unduly influence the results. For instance, when assessing a candidate's merit, we expect the algorithm to avoid unjustly relying on variables such as gender or race, and similarly, when using the model for critical decisions, such as allocating medical resources, we expect it not to overlook the interests of minority groups, whose relatively small representation may be downplayed by inductive statistical learning architectures. However, factors such as sex, race, and age are crucial for estimating a patient's risk of a specific disease, and designs aimed at ensuring identity-based equality no longer apply. In such cases, the algorithm should prioritize maximizing human well-being and lifesaving over achieving socio-economic equalization. To achieve fairness, different fields may have varying, or even opposing demands for algorithm design, making it highly challenging to balance through algorithm design alone. The most effective way to ensure fairness is through model interpretability, which helps identify the causes of unfair predictions and implement targeted interventions. The proposed model is interpretable at every stage to ensure fairness in practice.

The rest of this article is organized as follows: Section II reviews related work to provide a stronger contextual foundation for readers. Section III introduces the methodology of the proposed method, covering the main network structure, and the robust loss function. Section IV illustrates the implementation of noise-robust learning using a gradient-based optimizer and the Cauchy M-estimator. Section V presents and discusses the experiments and their outcomes. Finally, Section VI provides the conclusion and discusses future work.

II. Related Work

A. Fuzzy Sets and Theory

In classical set theory, an element either belongs to a set or does not belong, i.e., its membership is binary. However, binary logic has limitations when describing real-world problems. For example, in meteorology, days with average temperatures above 30°C are often considered "hot". According to classical set theory, if the average temperature on a particular day is 29.9°C, that day would not be categorized as "hot". Nevertheless, there is no substantial difference between 29.9°C and 30°C, and classical set theory is unable to delineate such nuances. In 1965, Zadeh introduced fuzzy set theory [18] to better represent similar cases in the real world. In fuzzy logic, the relationship between elements and sets is quantified by degrees of membership, represented as values between 0 and 1. The collection of these values forms the membership function. For a universe of discourse X, a fuzzy set $\tilde{V}$ is defined by a set of elements $x \in X$, expressed as

$$\tilde{V} = \{(x, \mu_V(x)) \mid x \in X\} \tag{1}$$

where $\mu_V(x) \in [0, 1]$ represents the degree of membership of the element x in the fuzzy set $\tilde{V}$. Specifically, $\mu_V(x) = 0$ indicates that x does not belong to $\tilde{V}$ at all, $\mu_V(x) = 1$ means that x fully belongs to $\tilde{V}$, and $0 < \mu_V(x) < 1$ implies that x partially belongs to $\tilde{V}$.

Fuzzy set theory provides a mathematical framework for dealing with uncertainty. Functionally, it shares some similarities with probability theory. However, fuzzy logic addresses "vagueness", while probability describes "randomness" [19]. The two represent uncertainty from different angles and should not be conflated. Interestingly, there is a fairly widespread misunderstanding in computer science that probability is superior to fuzzy logic in modeling uncertainty. Those who hold this view provide two main arguments: first, they perceive fuzzy logic as "rough" rather than "precise"; second, they point out that fuzzy logic emerged late and its theoretical framework is not as developed as that of probability. The first argument is clearly misguided, as fuzzy logic provides a method for quantifying imprecision, but this does not necessitate its mathematical nature being imprecise. Fuzzy mathematics is as rigorous and precise as any other branch of mathematics.

The second argument misinterprets the relationship between fuzzy logic and probability, treating them as competitive rather than complementary. Probability and fuzzy logic are distinct methods for handling uncertainty, each with its own advantages in different scenarios, making it illogical to consider either as superior. The existence of a research field known as the fuzzy-probabilistic model [20] strongly supports this perspective. Also, the truth that fuzzy theory emerged later and its framework is less developed compared with its probability counterpart does not logically imply inferiority. Even the seemingly more established probability theory has its shortcomings, most notably the inconsistency between the frequentist and Bayesian interpretations of probability, exemplified by Lindley's paradox [21]. As George E.P. Box famously said, "All models are wrong, but some are useful." [22], a sentiment that applies equally to mathematical theories.

Gödel's incompleteness theorem demonstrates that mathematics cannot be reduced to a finite set of axioms and inference rules, nor can every mathematical conclusion be derived from a limited set of axioms [23]. It unveils the essence of mathematical truth that transcends any formal system. This conclusion also implies that any computer system based on finite rules will inevitably have blind spots and will be unable to solve every problem. Fuzzy theory provides a unique approach to handling uncertainty and may address issues that other theories cannot, highlighting the profound significance of its research. It draws inspiration from the imprecise nature of human thinking and closely mirrors the logic of human language, making it, to some extent, a mathematical extension of human intuition. This also gives fuzzy systems exceptional interpretability. In an era dominated by black-box machine learning methods, transparent and interpretable fuzzy systems also contribute to unique and valuable research perspectives.

B. Interval Type-2 (IT2) Fuzzy Sets and Systems

Zadeh introduced the idea of type-2 fuzzy logic in 1975 [24] by adding a dimension of uncertainty to type-1 fuzzy logic, i.e., "the union of the primary memberships," explicated by Mendel [25]. A type-2 fuzzy membership function typically consists of an upper membership function (UMF) and a lower membership function (LMF), the gap between them is referred to as the footprint of uncertainty (FOU) [26]. To clarify this concept, we continue with the previous example. It was mentioned that a daily average temperature above 30°C is typically considered "hot." However, the definition of "hot" weather varies across countries, some setting the threshold at 27°C, while others may define it at 35°C. Therefore, at a daily average temperature of 29.9°C, its membership degree for "hot weather" becomes a range, i.e., the FOU, rather than a fixed value. The relationship of 29.9°C to the 27°C and 35°C thresholds corresponds to the UMF and LMF, respectively.

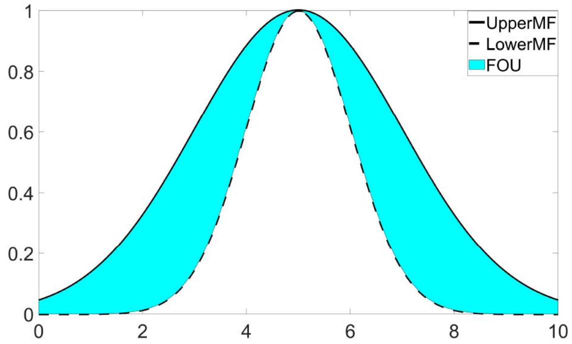

Fig. 1. IT2 Gaussian membership function and its FOU.

This key feature allows type-2 logic to model both intra-individual and inter-individual uncertainty. However, the three-dimensional nature of FOU in general type-2 logic [27] usually results in exceedingly complex computations. Therefore, the simplified version, interval type-2 fuzzy logic [28], is often preferred as it fixes the distance between the UMF and LMF in the third dimension to 1, which reduces the FOU to two dimensions and significantly simplifies the calculations. Assume X is a set containing a collection of elements x within the universe of discourse W, with $\mu_A(x)$ representing the corresponding membership value. The interval type-2 fuzzy set $\tilde{A}$ can be defined as follows:

$$\tilde{A} = \{[(x,u),1] \mid \forall x \in W, \forall \mu \in J_x \subseteq [0,1]\} \tag{2}$$

where J_x denotes the primary membership. For a continues ξ, the FOU of interval type-2 fuzzy sets can be represented as follows:

$$FOU(\xi) = \bigcup_{x \in W} J_x = \bigcup_{x \in W} \left[\underline{\mu_\xi}, \overline{\mu_\xi}\right] \tag{3}$$

where $\underline{\mu_\xi}$ and $\overline{\mu_\xi}$ are the lower membership function and the upper membership function, respectively. Fig. 1 provides a visualization of an interval type-2 Gaussian membership function and its FOU.

Similarly to type-1 fuzzy logic, interval type-2 logic also has two different methods for organizing fuzzy inference systems: the Mamdani method [29] and the Sugeno method [30], [31]. The Mamdani inference system is known for its excellent semantic interpretability and is widely used in fuzzy systems where rules are based on expert knowledge. In contrast, the Sugeno inference system is more formally concise and aligns closely with rigorous mathematics, leading to higher accuracy compared with the Mamdani approach [32]. This characteristic makes Sugeno systems suitable for data-driven adaptive neuro-fuzzy systems, which is also the case for the proposed IT2-ENFIS in this article.

For a fuzzy inference system with p inputs and one output, the antecedent of the i th fuzzy rule can be represented as follows:

$$\begin{aligned}&\text{Rule } i\text{: IF } l_1 \text{ is } A_{i,1}\left(x_1\right) \text{ and } l_2 \text{ is } A_{i,2}\left(x_2\right)\ldots\\ &\quad \text{and } l_p \text{ is } A_{i,p}\, x_p\right).\end{aligned}$$

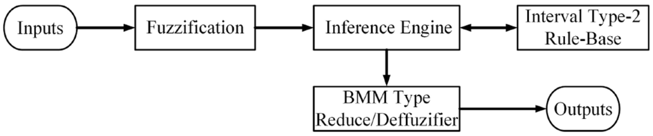

Fig. 2. Interval type-2 inference system with the BMM type reducer.

Following the defuzzification strategy of the Sugeno method, the output δ of this rule is given by

$$\delta_i = a_{i,0} + \sum_{j=1}^{p} a_{i,j} x_j, i = 1, 2, \ldots, q, j = 1, 2, \ldots, p \tag{4}$$

where $a_{i,j}$ denotes the corresponding consequent parameter of $A_{i,j}(x_j)$. For interval type-2 inference systems, the fuzzy inference results must undergo type reduction to achieve the final model output. This article employs the popular Begian–MelekMendel (BMM) [33] type reduction method, described as follows:

$$y = u \frac{\sum_{i=1}^{q} \underline{s_i} \delta_i}{\sum_{i=1}^{q} \underline{s_i}} + v \frac{\sum_{i=1}^{q} \overline{s_i} \delta_i}{\sum_{i=1}^{q} \overline{s_i}} \tag{5}$$

where u and v are adjustable coefficients, $\underline{s_i}$ and $\overline{s_i}$ indicate the firing strengths of the i th LMF and UMF, respectively. Fig. 2. shows a schematic diagram of a typical interval type-2 fuzzy inference system with the BMM type reducer. Note that the BMM method performs both type reduction and Sugeno defuzzification simultaneously, allowing for direct crisp outputs.

Type-2 fuzzy logic can model both intra-individual and inter-individual uncertainty within the same rule, demonstrating robust adaptability and generalization capabilities. As a simplified version of general type-2 fuzzy logic, interval type-2 fuzzy logic retains most of its properties with considerably reduced difficulty in computation [34]. Compared with traditional type-1 fuzzy logic, interval type-2 fuzzy logic can enclose significantly more uncertainty, thereby reducing the number of rules needed in reasoning [35]. It provides greater design flexibility for inference systems while avoiding the complexity of general type-2 fuzzy systems, making it ideal for constructing the trustworthy learning model discussed in this article.

C. Efforts in Noise-Robust Machine Learning

In many cases, measurement errors, human errors, sampling biases, and even natural outliers can become potential sources of noise. Researchers in the field of machine learning have long recognized the detrimental effects of noise and have been investigating countermeasures to mitigate its impact. Traditionally, data pre-processing techniques, such as manual screening, signal processing (e.g., filtering and smoothing), clustering analysis, and some statistical methods, are available to remove noise. Another intuitive approach to improve the noise resistance of a model is replacing the cost function with a robust alternative, such as the Huber loss [16]. However, these methods have limitations and are insufficient to tackle more complex noise environments.

Further robustness can be achieved by implementing structured noise adaptation mechanisms in models, such as label-noise representation learning (LNRL) [36]. Goldberger and Benreuven [37] enhanced model resistance to outliers by integrating a noise adaptation layer into deep networks. Patrini et al. [38] introduced loss correction methods in deep learning models to bolster robustness. Wang et al. [39] and Ren et al. [40] employed data reweighting techniques to enhance network tolerance to noise. Jiang et al. [41] and Han et al. [42] proposed and refined a small-loss technique, effectively eliminating cumulative errors and suppressing label noise. Despite advancement in robustness through LNRL methods, they also increase training complexity. Later, Wang et al. [43] employed a meta-learning approach to estimate the noise transition matrix, compensating for the shortcomings of previous LNRL methods and enhancing the efficiency of the network. Carmon et al. [44] proposed a semisupervised learning strategy that allows networks to enhance robustness by simply adding more unlabeled data samples, for which the approach is theoretically proved. However, both semisupervised learning and meta-learning require clean validation sets for pretraining, which may not be available in some scenarios. Deng et al. [45] trained specialized generative models to simulate the distribution of class labels, aiming to enhance the robustness of the discriminator against label noise. Na et al. [46] proposed a noise adaptation mechanism based on a transition-aware weight estimator and a time-dependent noisy-label classifier to enhance the robustness of diffusion models against label noises.

Despite many exploratory attempts, current noise-resistant trustworthy machine learning methods still suffer from several drawbacks.

1) Complex to implement and mainly suited for deep networks or large models instead of smaller-scale architectures.
2) Research mainly focuses on improving the label noise robustness of classification and discrimination models, with relatively little attention given to noise-resistant mechanisms for regression tasks.
3) Improvements in noise resistance often come at the expense of reduced accuracy and generalization capability.
4) The noise-countering mechanism significantly increases the complexity of model training.
5) Poor interpretability and transparency, as they were initially developed for black-box models, which contradicts the general requirement for interpretability in trustworthy systems.

To address the above issues, Xue et al. proposed an exclusionary neural complex fuzzy inference system (ENCFIS) [47] based on complex fuzzy sets and logic [48], [49], [50]. As the first de-facto trustworthy solution in the field of fuzzy systems customized for high-noise regression learning environments, this model builds upon the rapid adaptive complex fuzzy inference system (RACFIS) [51] by incorporating a robust loss function [16] as well as the Welsch M-estimator [52] to mitigate pernicious effects of label noise in the data, achieving outstanding results. However, problems remain in three aspects for ENCFIS. First, the complex fuzzy theory employed is still

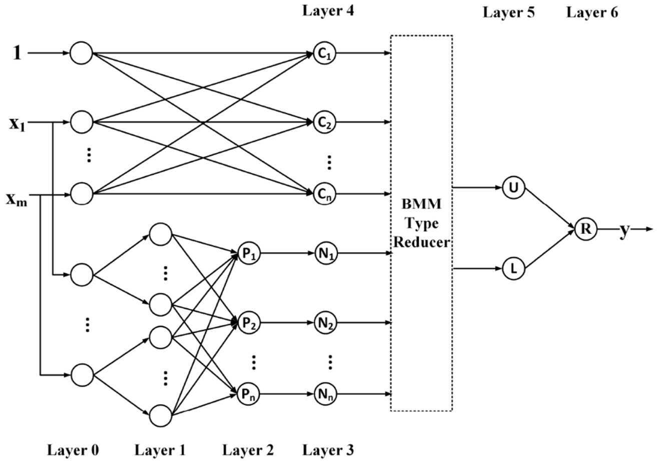

Fig. 3. Main network structure of IT2-ENFIS. (X_i denotes the input variable, P_i represents the firing strength of each rule, N_i is the normalization layer, U and L are the outcomes of the Sugeno defuzzification and the BMM type-reducer for the upper and lower membership functions, respectively. The sum of the previous layer, R, is identical to the network's output, denoted as y).

theoretically underdeveloped, as its interpretability remains inaccessible. Second, complex fuzzy sets cannot simultaneously model intraindividual and interindividual uncertainty, thus limiting generalization capability in many scenarios. Lastly, ENCFIS adopts a first-order derivative optimization method, which is susceptible to fluctuation and interference in the data, making convergence difficult and prone to local optima. By introducing a more effective methodology, this article aims to resolve the aforementioned shortcomings.

III. Network Structure

This section outlines the forward structure of the proposed neuro-fuzzy inference system, detailing how interval type-2 fuzzy logic is employed to derive the network output and how a robust loss function evaluates the quality of this output.

A. Primary Network of IT2-ENFIS

IT2-ENFIS utilizes a six-layer network architecture, as depicted in Fig. 3. It utilizes an interval type-2 Gaussian membership function, of which the upper and lower functions share the same centroid but have different spread values [53]. The functioning of each layer with the p-dimensional input vector $x(k) = [x_1(k), x_2(k), \ldots, x_p(k)]^T$ at time k is described as follows:

Layer 1: In this layer, fuzzification occurs, mapping the FOU into a two-dimensional domain using the following Gaussian membership function:

$$O^1_{i,j}(k) = \exp\left(-\frac{(x(k) - \mu_{i,j})^2}{2\left(\sigma_{i,LMF}{}^2, \sigma_{i,UMF}{}^2\right)}\right) \tag{6}$$

where $O^1_{i,j}(k)$ represents the IT2 membership function of the i th rule for the j th input variable, and $\{\mu_{i,j}, \sigma_{i,LMF}, \sigma_{i,UMF}\}$ denotes the set of antecedent parameters for each node.

Layer 2: This layer calculates the firing strength of each rule by applying t-norm intersection using the product method

$$O^2_i(k) = \prod_{j=1}^{p} O^1_{i,j}(k) =: [\alpha_i(k), \beta_i(k)] \tag{7}$$

$$\begin{cases} \alpha_i(k) = \prod_{j=1}^{p} \exp\left(-\frac{(x(k)-\mu_{i,j})^2}{2\sigma_{i,LMF}{}^2}\right) \\ \beta_i(k) = \prod_{j=1}^{p} \exp\left(-\frac{(x(k)-\mu_{i,j})^2}{2\sigma_{i,UMF}{}^2}\right) \end{cases}, \tag{8}$$

where $O^2_i(k)$ denotes the output of the i th firing and $\alpha_i(k), \beta_i(k)$ represent the firing strengths of LMFs and UMFs, respectively.

Layer 3: The firing strengths of LMFs and UMFs are normalized separately in this layer to determine the inference result of the antecedent network

$$O^3_i(k) = \frac{O^2_i(k)}{\sum_{r=1}^{4} O^2_r(k)} = \begin{cases} \frac{\alpha_i(k)}{\sum_{r=1}^{4} \alpha_r(k)} \\ \frac{\beta_i(k)}{\sum_{r=1}^{4} \beta_r(k)} \end{cases} \tag{9}$$

where $O^3_i(k)$ denotes the normalized output of i th node in this layer.

Layer 4: This layer connects directly to the input vector to compute the consequent coefficients for a type-1 Sugeno. defuzzifier

$$O^4_i(k) = e_{i,0} + \sum_{j=1}^{p} e_{i,j} x_j(k) \tag{10}$$

where $O^4_i(k)$ is the output of this layer, to be used in the subsequent layer as the consequent coefficient to generate the crisp output, and $\{e_{i,0}, e_{i,1}, e_{i,2}, \ldots, e_{i,p}\}$ are linear parameters.

Layer 5: In this layer, the BMM-type reducer and Sugeno defuzzifier are applied to convert the fuzzy inference outcomes into crisp values

$$\begin{cases} O^5_{LMF}(k) = u \cdot \frac{\sum_{i=1}^{q} a_i(k) O^4_i(k)}{\sum_{i=1}^{q} a_i(k)} \\ O^5_{UMF}(k) = v \cdot \frac{\sum_{i=1}^{q} \beta_i(k) O^4_i(k)}{\sum_{i=1}^{U} \beta_i(k)} \end{cases} \tag{11}$$

where u and v are BMM coefficients, and $O^5_{LMF}(k)$ and $O^5_{UMF}(k)$ represent the crisp outputs of the consequent network for LMFs and UMFs, respectively. Note that in IT2 neuro-fuzzy systems, the adjustable coefficient of the BMM type reducer is usually fixed at 0.5 to simplify computation. This research adopts the same approach, setting $u = v = 0.5$ for the IT2-ENFIS network.

Layer 6: The overall network output is determined by simply summing the outputs from layer 5

$$O^6(k) = O^5_{LMF}(k) + O^5_{UMF}(k) \tag{12}$$

where $O^6(k)$ is the final crisp output.

The proposed architecture adopts an incomplete rule-based design, where the number of rules corresponds to the number of membership functions, establishing a linear relationship between complexity and the number of dimensions. The complexity of the antecedent network can be expressed in Big O notation as $O(kdt)$, while the complexity of the consequent network is $O(k(d+1))$, where k represents the number of rules, d is the dimensionality of the input features, and t is the number of parameters of each membership function. Thus, the complexity of the primary network is $O(kdt + k(d+1))$. Traditional complete rule-based fuzzy inference systems compute the firing strength by searching all possible combinations of membership functions, leading to an exponential growth of the rule base as dimensionality increases. Given the high dimensionality of modern data scenarios, the above approach often causes an explosion in the rule base. Moreover, neuro-fuzzy systems with self-learning capabilities can accomplish tasks with few rules, making a complete rule-based design unnecessary. Therefore, adopting an incomplete rule base in this context is a reasonable choice.

B. Robust Loss Functions

Traditionally, machine learning regression tasks employ the L2 loss function, i.e., the mean square error (MSE). This function is favored because it ensures stable performance even with minute residuals, leading to high accuracy of the solver. However, L2 loss is susceptible to surges in residuals and is ill-suited for data environments with significant noise and outliers. In contrast, the L1 loss function, also known as the mean absolute error (MAE), imparts high robustness and reduces sensitivity to fluctuations. However, the function's lack of smoothness near zero makes it challenging to achieve precise solutions, leading to its less frequent use.

Huber combined the characteristics of both loss functions to create the Huber robust loss function [16], which preserves their strengths while addressing their limitations. The general form of the Huber function is as follows:

$$\hat{h}(t) = \begin{cases} \frac{1}{2}(y(t) - g(t))^2, & |y(t) - g(t)| \leq \delta \\ \delta|y(t) - g(t)| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \tag{13}$$

where $\hat{h}(t)$ denotes the Huber loss, $|y(t) - g(t)|$ represents the residual at the t th input, δ is a tuning coefficient that serves as the threshold for switching between L1 and L2 losses. By adjusting δ, the Huber loss function applies L1 loss to large residuals and L2 loss to smaller residuals, thereby enhancing the robustness of the traditional MSE loss function while retaining its smoothness around zero. Since noise can cause MSE to deviate from the optimization target, introducing Huber loss helps to counteract the misleading effects of erroneous labels and smooth the training process. Although other robust loss functions, such as quantile loss [54] and log-cosh loss [55], are available, they have not demonstrated better performance than the Huber loss in experiments with the proposed architecture. As a result, the original Huber loss remains the optimal choice for IT2-ENFIS.

C. Interpretability Analysis

The interpretability of neural networks lacks a unified evaluation standard. The definition and requirements for interpretability differ across various scenarios, and their forms vary significantly depending on the model architecture. The interpretability of IT2-ENFIS is analyzed from the perspective of fuzzy systems and primarily arises from the following aspects.

1) The network is essentially a Sugeno fuzzy inference system, where intricate input-output mapping relations are represented by a set of IF-THEN rules. This structure closely mirrors human thought processes, and, unlike conventional black-box models, it can unravel the reasoning process behind the results.
2) Through membership functions, input variables are transformed into interval type-2 fuzzy sets, defining the degree of membership in various semantic contexts and assigning them physical meaning. For example, as mentioned earlier, hot weather is defined as a daily average temperature above 27°C in some countries, while in others, the threshold is 35°C. For a daily average temperature of 29.9°C, the membership value is 1 according to the 27°C standard (completely satisfying the condition), while according to the 35°C standard, the membership value may be a reasonable value such as 0.7. As a result, the path from each input variable to the output is visible, and in conjunction with IF-THEN rules, it becomes possible to analyze how the input variables affect the output.
3) It has a clear structure of functional modules, where the functionality and output of each module are independently interpretable. The fuzzification layer defines the semantics of the inputs, the inference layers show the contribution of different rules to the results, and the defuzzification layer explains the final process that leads to the outcome. Such a design ensures the transparency of the algorithm and the traceability of the results.
4) Fuzzy inference also provides domain interpretability, as its semantic explanation enables the evaluation of fuzzy rules and membership functions through expert knowledge. It is easier to gain acceptance and trust when the model aligns with expert understanding. Even when the generated model differs from expert knowledge, it remains valuable for uncovering potential new patterns and insights.

In conclusion, IT2-ENFIS exhibits model transparency, global interpretability, local interpretability for each module, and semantic interpretability. Although the definition of interpretability varies across domains, it is reasonable to assert that the interpretability of IT2-ENFIS satisfies the established criteria for trustworthy learning.

IV. Noise-Robust Learning for IT2-ENFIS

This section examines the Gradient-based Optimizer (GBO) and the Cauchy M-estimator and elucidates how they facilitate the noise-robust learning of the proposed network. Fig. 4 provides a flow chart representation of the IT2-ENFIS noise-resistant learning architecture.

A. Gradient-Based Optimizer (GBO)

The gradient-based optimizer [15] combines the strengths of second-order Newton's method with metaheuristic techniques, enabling it to effectively address complex optimization problems. It provides faster convergence and higher search accuracy against traditional meta-heuristic methods such as particle

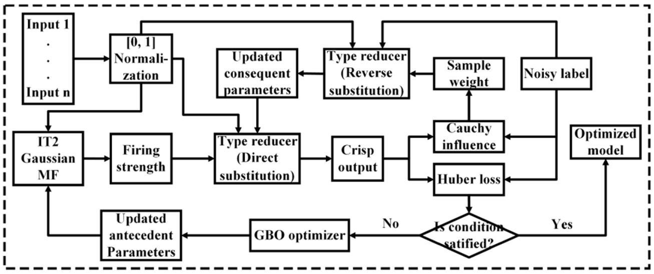

Fig. 4. Flowchart of the IT2-ENFIS noise-robust learning architecture.

swarm optimization (PSO) [56]. This optimizer features two innovative operators: the gradient search rule (GSR) and the local escaping operator (LEO).

1) Gradient Search Rule (GSR): The gradient search rule (GSR) directs vector movement to enhance solution search within the feasible domain. It helps manage random behavior during optimization, facilitating exploration and avoiding local optima. For a vector z_n to be optimized, its GSR operator is defined as follows:

$$GSR = \text{randn} \cdot \frac{2 - \Delta z \cdot z_n}{\left(z_{\text{nocot}} - z_{\text{best}} + \varepsilon\right)} \cdot \rho_1 \tag{14}$$

where random value randn $\in (0, 1]$ follows a standard normal distribution, ε can be an arbitrary number between 0 and 0.1, z_{worst} and z_{best} represent the worst position and best position among the entire population, respectively. ρ_1 signifies the GSR coefficient, which can be calculated as follows:

$$\begin{cases} \rho_1 = (2 \cdot \text{ rand } - 1) \cdot \alpha \\ \alpha = \left|\beta \sin\left(\frac{3\pi}{2} + \sin\left(\frac{3\pi}{2}\beta\right)\right)\right| \\ \beta = \beta_{\min} + (\beta_{\max} - \beta_{\min})\left(1 - \left(\frac{m}{M}\right)^3\right)^2 \end{cases} \tag{15}$$

where rand $\in (0, 1]$ is a randomly generated number, $\beta_{\min}$ and $\beta_{\max}$ are 0.2 and 1.2 , respectively. m refers to the ordinal of current iteration, while M denotes the total number of iterations. Given that the complexity of calculating derivatives and the fact that closed-form derivatives may not be available in many optimization scenarios, Δz serves as a numerical replacement for the genuine gradient in (14), which can be obtained as follows:

$$\begin{cases} \Delta z = \text{rand}(N) \cdot \frac{\left|\left(z_{\text{best}} - z_{r_1}\right) + \eta\right|}{2} \\ \eta = 2 \cdot \text{rand} \cdot \left(\left|\frac{z_{r_1} + z_{r_2} + z_{r_3} + z_{r_4}}{4} - z_n\right|\right) \end{cases} \tag{16}$$

where $z_{r_1}, z_{r_2}, z_{r_3}$, and z_{r_4} $(r_1 \neq r_2 \neq r_3 \neq r_4 \neq n)$ are seeds randomly selected from the population, $\text{rand}(N) \in (0, 1]$ is a randomly generated N dimensional vector with N refers to the population size.

To better exploit the solution space near z_n, the direction of movement (DM) factor is used in conjunction with the GSR, which can be determined as follows:

$$DM = \text{rand} \cdot \left(z_{r_1} - z_{r_2}\right) \cdot \rho_2 \ , \tag{17}$$

where ρ_2 can also be obtained from (15), with the condition that ρ_1 is distinct from ρ_2. Therefore, a new position of the current solution vector z_n can be updated as follows:

$$Z_n^1 = z_n - GSR + DM. \tag{18}$$

By substituting (14) and (17) into (18), the complete form is acquired as follows:

$$\begin{aligned} Z_n^1 = z_n - \text{randn} \cdot & \frac{2 \cdot \Delta z \cdot z_n}{\left(z_{\text{worst}} - z_{\text{best}} + \varepsilon\right)} \cdot \rho_1 \\ & + \text{rand} \cdot \left(z_{r_1} - z_{r_2}\right) \cdot \rho_2. \end{aligned} \tag{19}$$

However, two additional position vectors are needed to finalize the GSR process. By replacing vectors z_{worst} and z_{best} with vectors p_n and q_n in (19), and substituting z_n with z_{best}, the second position vector Z_n^2 can be generated as follows:

$$\begin{aligned} Z_n^2 = z_{\text{best}} - \text{randn} \cdot & \frac{2 \cdot \Delta z \cdot z_n}{\left(p_n - q_n + \varepsilon\right)} \cdot \rho_1 \\ & + \text{ rand} \cdot \left(z_{r_1} - z_{r_2}\right) \cdot \rho_2 \end{aligned} \tag{20}$$

where p_n and q_n are determined as follows:

$$\begin{cases} p_n = \text{rand} \cdot \left(\frac{[g_{n+1} + z_n]}{2} + \text{ rand } \cdot \Delta z\right) \\ q_n = \text{rand} \cdot \left(\frac{[g_{n+1} + z_n]}{2} - \text{rand} \cdot \Delta z\right) \\ g_{n+1} = z_n - \text{randn} \cdot \frac{2 \cdot \Delta z \cdot z_n}{\left(z_{\text{wost}} - z_{\text{best}} + \varepsilon\right)} \end{cases} . \tag{21}$$

Subsequently, the third position vector Z_n^3 can be obtained as follows:

$$Z_n^3 = z_n - \rho_1 \cdot \left(Z_n^2 - Z_n^1\right). \tag{22}$$

Hence, one can obtain the solution vector for the GSR

$$z_n^{GSR} = r_a \cdot \left(r_b \cdot Z_n^1 + (1 - r_b) \cdot Z_n^2\right) + (1 - r_a) \cdot Z_n^3 \tag{23}$$

where $r_a, r_b \in (0, 1]$ are unequal random numbers.

2) Local Escaping Operator (LEO): The LEO is a crucial component of the GBO optimizer, designated to facilitate the search accuracy and prevent premature convergence to local optima. By leveraging LEO, GBO is better equipped to tackle complex optimization problems effectively. It utilizes position vectors obtained from the GSR, i.e., Z_n^1, Z_n^2 and z_{best}, to generate a better solution Z_{LEO}, as illustrated as follows:

$$\begin{aligned} Z_{LEO} = & \ z_n^{LEO} + f_1 \cdot \left(l_1 z_{\text{best}} - l_2 z_k\right) \\ & + 0.5 f_2 \rho_1 \left[l_3 \cdot \left(Z_n^2 - Z_n^1\right) + l_2 \cdot \left(z_{r_1} - z_{r_2}\right)\right] \end{aligned} \tag{24}$$

where $f_1 \in [-1, 1]$ is a uniformly distributed random value, $f_2 \in (0, 1]$ accords with the standard normal distribution. z_n^{LEO} denotes the initial position in this process. If rand < 0.5, then $z_n^{LEO} = z_n^{GSR}$; otherwise, $z_n^{LEO} = z_{\text{best}}$. u_1, u_2 and u_3 are also random numbers which can be determined using the following method:

$$\begin{cases} l_1 = 2 \cdot S \cdot \text{ rand } + (1 - S) \\ l_2, l_3 = S \cdot \text{ rand } + (1 - S) \end{cases} \tag{25}$$

where S is a binary value defined as $S = 1$ if $l_1 < 0.5$; otherwise, $S = 0$. Subsequently, the position vector z_k in (24) can be obtained as follows:

$$z_k = \begin{cases} Z_{\min} + \text{ rand } \cdot \left(Z_{\max} - Z_{\min}\right), \mu_2 < 0.5 \\ z_p, \ \text{otherwise} \end{cases} \tag{26}$$

where rand $\in (0, 1]$ is a random number, and z_p refers to a random seed from the population. Therefore, the solution vector z_n^{new} at the end of this iteration is as follows:

$$z_n^{\text{new}} = Z_{LEO}. \tag{27}$$

B. Cauchy M-Estimator

In linear parameter estimation, classical statistical methods are effective as long as the parametric model is accurate and free from significant outliers. However, real-world situations often deviate from idealized situation and contain noise or distortions. "Robust estimators" [57] were developed to tolerate imprecise models and abnormal observations. M-estimation, a prominent method in this genre, introduces an influence function to evaluate the importance of data samples, making it sensitive to data near the mean in distribution while remaining robust against outliers. Thus, it can provide reliable estimates even without prior knowledge of the data. There are two types of M-estimators: the ρ-estimator and ψ estimator.

For ρ-type estimation, consider an n-dimensional measure space $\Omega \in \mathbb{R}^n$, with $\zeta \in \Omega$ as the parameter vector of the model. The representation of this M estimator $\Xi(T)$, according to the mapping $\rho : \chi \times \Omega \to \mathbb{R}^n$, is given as follows:

$$\Xi(T) := \text{argmin}_{\zeta \in \Omega} \int_{\chi} \rho(x, \zeta) dT(x) \tag{28}$$

where $\rho(x, \zeta)$ represents the influence function, T denotes the distribution of observed values, and χ refers to the distribution of estimates. The M-estimator can be reduced to the ordinary maximum likelihood estimator if $\rho(x, \zeta) = -\ln(\partial T(x, \zeta)/\partial x)$.

For an influence function ρ that is differentiable and continuous, the M-estimator has a more convenient form, known as the ψ-type. In this form, the estimator $\lambda(T) = (\partial \rho(x, \zeta)/\partial \zeta)$ is defined as follows:

$$\int_{\chi} \psi(x, \lambda(T)) dT(x) = 0 \tag{29}$$

and the estimate of ζ can be determined using the equation below

$$\int_{\chi} \frac{\partial \rho(x, \zeta)}{\partial \zeta} dT(x) = 0. \tag{30}$$

The robustness of an M-estimator depends closely on the chosen influence function for a given problem. Among the various influence functions proposed, five are widely used: Huber, Hampel, Welsch, Bisquare, and Cauchy functions [17].

This article applies a ψ-type M-estimator [17] based on Cauchy influence function to robustly estimate the network's consequent parameters. The Cauchy influence is defined as follows:

$$f_C(\varepsilon) = \frac{\varepsilon}{1 + \left(\frac{\varepsilon}{c}\right)^2}, |\varepsilon| \leq \infty. \tag{31}$$

TABLE I
EFFICIENCY LEVEL UNDER DIFFERENT c FOR CAUCHY M-ESTIMATOR

c	1.7249	2.3850	3.3962	4.2904
Efficiency level	90%	95%	98%	99%

For a discrete system model

$$y_C = S_C \theta + \omega_C \tag{32}$$

where S_C denotes the sample space, θ represents the estimate, y_C is the observation vector, and ω_C refers to the independent noise. Results of the estimator are obtained by minimizing the following objective function:

$$\sum_{i=1}^{n} f_C \left(\frac{y_i - s_i \hat{\theta}}{\hat{\sigma}} \right) = \sum_{i=1}^{n} f_C (\varepsilon_i) \tag{33}$$

where $s_i \in S_C$, $\hat{\sigma}$ represents the scale factor, ε_i is i th the residual, f_C refers to Cauchy influence. For a continuously differentiable function f_C, let $\psi = (\partial f_C/\partial \theta)$. The minimization step simplifies to solving the following equation:

$$\sum_{i=1}^{n} \psi \left(\frac{y_i - s_i \hat{\theta}}{\hat{\sigma}} \right) s_i = \sum_{i=1}^{n} \psi (\varepsilon_i) s_i = 0. \tag{34}$$

Define the weight function $w(\varepsilon)$ under the Cauchy influence as follows:

$$w(\varepsilon) = \frac{\psi(\varepsilon)}{\varepsilon} = \frac{1}{1 + \left(\frac{\varepsilon}{c}\right)^2} \tag{35}$$

and substitute it into (34), the function further reduces to

$$\sum_{i=1}^{n} w(\varepsilon_i) \varepsilon_i s_i = 0. \tag{36}$$

By substituting $\varepsilon_C = S_C \hat{\theta} - y_C$ into (36), the M-estimation becomes a weighted least squares problem as follows:

$$S_C^T W S_C \hat{\theta} - S_C^T W y_C = 0 \tag{37}$$

where W is the weight matrix defined by $w(\varepsilon)$. Thus, we can achieve the final estimate $\hat{\theta}$ as follows:

$$\hat{\theta} = \left(S_C^T W S_C \right)^{-1} S_C^T W y_C. \tag{38}$$

The tuning coefficient c in the weight function governs the degree to which the influence function suppresses outliers. Higher values reduce sensitivity to outliers, improving the M-estimator's performance but sacrificing robustness. Normally, a larger value of c is chosen when the noise level is low, whereas a smaller c is preferred when the noise level is high. Each value of c corresponds to a specific confidence level. For instance, at the 90% confidence level, $c = 1.7249$. Table I shows the mapping relation between different c settings and their corresponding confidence levels. Typically, the 95% confidence level is recommended [17] in most cases.

C. Parameters Updating

This section illustrates the proposed robust learning method, which involves the GBO, the Cauchy M-estimator, and the Huber loss. For the GBO method, the search space should be confined to a specific range, ideally between 0 and 1. Significant variations in the scale of variables can also adversely affect the performance of the M-estimator. Therefore, all inputs should be normalized to the interval $[0, 1]$ before being applied to the network. For an antecedent network with p rules and q input dimensions, its parameter vectors can be defined as follows:

$$\begin{cases} \vec{\mu} = [\mu_{1,1}, \mu_{1,2}, \ldots, \mu_{1,q}, \ldots, \mu_{p,1}, \mu_{p,2}, \ldots, \mu_{p,q}] \\ \overrightarrow{\sigma_{LMF}} = [\sigma_{1,LMF}, \sigma_{2,LMF}, \ldots, \sigma_{p,LMF}] \\ \overrightarrow{\sigma_{UMF}} = [\sigma_{1,UMF}, \sigma_{2,UMF}, \ldots, \sigma_{p,UMF}] \end{cases} . \tag{39}$$

Subsequently, the position vector z used for the GBO optimization, also referred to as the seed, can be constructed as follows:

$$z = [\vec{\mu}, \overrightarrow{\sigma_{LMF}}, \overrightarrow{\sigma_{UMF}}] . \tag{40}$$

Assuming N seeds, each with elements ranging between 0 and 1, are stochastically initialized for training, the entire population Λ can be represented as follows:

$$\Lambda = \left[z_1^T, z_2^T, \ldots, z_N^T\right] . \tag{41}$$

Next, parameters represented by each seed are fed into the network, and the Huber loss function is used to evaluate their performance. For a training dataset with n samples, the average Huber loss $\widehat{H}(z_\tau)$ across all samples can be computed as follows:

$$\widehat{H}(z_\tau) = \frac{1}{n} \sum_{k=1}^{n} \widehat{h}_{z_\tau}(k) \tag{42}$$

where z_τ denotes the τ th seed in the population, $\tau = 1, 2, \ldots, N$ and $\widehat{h_{z_l}}(k)$ can be obtained using formula (13). The worst and best performing seeds are then selected as z_{worst} and z_{best}, respectively, according to the following criteria:

$$\begin{cases} z_{\text{worst}} = \operatorname{argmax}_{z \in \Lambda} \left\{ \widehat{H}(z_1), \widehat{H}(z_2), \ldots, \widehat{H}(z_N) \right\} \\ z_{\text{best}} = \operatorname{argmin}_{z \in \Lambda} \left\{ \widehat{H}(z_1), \widehat{H}(z_2), \ldots, \widehat{H}(z_N) \right\} \end{cases} . \tag{43}$$

By applying z_{worst} and z_{best} to the GBO optimizer and training with the GSR and LEO operators, an updated z_{best} is obtained, serving as an optimized solution for the antecedent network in this iteration.

For the linear consequent parameters, the Cauchy M-estimator is used to robustly determine the optimum. Given a set of n data samples, the equation coefficient matrix S_i, diagonal weight matrix W_c, data label vector y, and parameter vector θ_i are defined as follows:

$$S_i = \begin{bmatrix} f_{i,0}(x_1) & f_{i,1}(x_1) & \cdots & f_{i,q}(x_1) \\ f_{i,0}(x_2) & f_{i,1}(x_2) & \cdots & f_{i,q}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_{i,0}(x_n) & f_{i,1}(x_n) & \cdots & f_{i,q}(x_n) \end{bmatrix},$$

$$W_c = \operatorname{diag}(w_1, w_2, \ldots, w_n),$$

$$y = [y_1, y_2, \ldots, y_n]^T, \vartheta_i = [e_{i,0}, e_{i,1}, \ldots, e_{i,q}]^T \tag{44}$$

where elements $w_1, w_2, \ldots, w_n$ of W_c are determined according to (35), $f_{i,0}, f_{i,1}, \ldots, f_{i,q}$ represent the reasoning results of the antecedent network, and $y_1, y_2, \ldots, y_n$ are the training labels. Thus, the estimated consequent parameter for the i th rule can be calculated as:

$$\widehat{\vartheta_l} = \left[S_i^T W_c S_i\right]^{-1} S_i^T W_c y. \tag{45}$$

Algorithm 1: Learning Process for IT2-ENFIS (Proposed).

```
Step 1. Normalize the dataset to the interval $[0, 1]$.
Step 2. Generate the initial population $\Lambda = \left[z_1^T, z_2^T, \ldots, z_N^T\right]$ and the initial consequent parameter matrix $\hat{\theta}$.
Step 3. Input each seed into the network and calculate its average Huber loss $\widehat{H}$ on the training set.
Step 4. Evaluate $\left\{\widehat{H}(z_1), \widehat{H}(z_2), \ldots, \widehat{H}(z_N)\right\}$ obtained from the previous step, then designate the seed with the highest loss as $z_{\text{worst}}$ and the seed with the lowest loss as $z_{\text{best}}$.
Step 5. Apply $z_{\text{worst}}$ and $z_{\text{best}}$ to the GBO optimizer to acquire an updated $z_{\text{best}}$.
Step 6. Compute the antecedent network output with the latest $z_{\text{best}}$ and estimate the consequent parameter matrix $\hat{\theta}$ using the Cauchy M-estimator.
Step 7. Calculate the output of the network using the updated $z_{\text{best}}$ and $\hat{\theta}$ to obtain the renewed average Huber loss $\widehat{H}_{\text{new}}$.
Step 8. Repeat Step 5 Step 7 until the maximum iteration is reached.
```

Similarly, by applying (45) to the remaining rules, the consequent parameter matrix of this iteration can be obtained as follows:

$$\hat{\theta} = \left[\widehat{\vartheta_1}, \widehat{\vartheta_2}, \ldots, \widehat{\vartheta_p}\right] . \tag{46}$$

The GBO optimizer and M-estimator approach should repeat several recursions until the maximum epoch is reached to realize trustworthy learning. This process is highly robust against label noise for the following reasons. First, the GBO optimizer integrates the powerful line search capability of Newton's method with the stochastic aspects of a population-based metaheuristic approach, which reduces sensitivity to sporadic errors and improves stability. Second, the Huber loss function replaces the ordinary loss function for outcome evaluation that mitigates noise interference. Third, the noise-insensitive M-estimator determines the consequent parameters, improving issues with noise labels and outliers. The simplified algorithm execution process is illustrated in Algorithm 1 as follows.

V. Experiments and Discussions

This section includes two experiments designed to test the performance of the proposed trustworthy learning architecture. Simulations were conducted in the MATLAB 2023b environment. The benchmark models were primarily programmed from scratch, with some from MATLAB's built-in toolbox. The performance indicator used is the root-mean-square error (RMSE), calculated using the following equation:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{47}$$

where n denotes the number of samples, y_i is the true value, $\hat{y}_i$ is the estimated value, and $y_i - \hat{y}_i$ represents the residual.

TABLE II
IT2-ENFIS HYPERPARAMETER SETTING FOR SUNSPOT TIME-SERIES TEST

GBO and Network		M-Estimator and Huber Loss	
Population size N	50	Tuning constant c	2.3850
Max-iteration M	50		
Number of rules	4	Huber coefficient δ	0.1

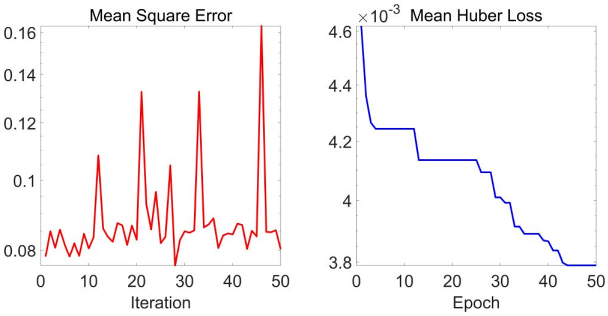

Fig. 5. Mean square error and mean Huber loss during training.

A. Noise-Contaminated Sunspot Time-Series Test

The Sunspot time-series dataset [58] was employed to benchmark ENCFIS in [47]. This study uses the same contaminated dataset to validate the performance of IT2-ENFIS, allowing for a better comparison with previous studies. For this purpose, 602 sunspot observations from 1976 to 1980 are utilized to construct the contaminated training set, whereas a separate set of 602 records from 1985 to 1989 is used to form the clean test set. Each test data sample is organized as $\{\gamma(\tau-2), \gamma(\tau-1); \gamma(\tau)\}$, with the two previous observations as the input vector and the current observation $\gamma(\tau)$ as the output label. In the training set, 150 out of 602 observations are evenly replaced with random values, compromising 25% of the time series. This introduces significant challenges for predictive models, as time series forecasting depends on prior observations to predict future values. Consequently, the training set constructed from this corrupted time series will result in 50% of the samples having incorrect inputs, 25% having misleading labels, and only 25% being intact, presenting both intense label and input noises.

The hyperparameter settings for IT2-ENFIS are presented in Table II, while Fig. 5 compares the MSE and mean Huber loss (MHL) during training. As shown in Fig. 5, the trends of MSE and MHL differ significantly due to the high data noise, which undermines the trustworthiness of the MSE values. In contrast, the MHL values generated by the Huber loss function more accurately reflect the actual learning progress of the model, thereby guiding the training process in the right direction. Fig. 6 visualizes the outputs of nonrobust models on both the contaminated training set and the clean test set, while Fig. 7 displays the corresponding experimental results of noise-robust models. Table III presents the RMSEs of all benchmark models on the clean test set.

The original Sunspot time series is not particularly challenging, as many models perform well with this dataset. The issues arise from strong noise and numerous outliers. Thus, robustness to disturbances is essential for achieving success with this data. From Fig. 6, models without dedicated noise-robust mechanisms, such as DBN [59], RBF [60], GRNN [61], BP [62], LSTM [63], and ANFIS [64], can hardly learn the correct information from contaminated data, which is also reflected in Table III that the RMSEs of these models on the clean test set are very high. However, interesting results emerge when evaluating the performance of models with enhanced robustness against noise. Based on their RMSE performances, the models are divided into three tiers. The lowest tier includes IT2-Sugeno [65] and SVR [66], which is expected as neither is designed for high-noise environments. IT2-Sugeno's interval type-2 fuzzy logic provides a constrained tolerance to noise in input values, whereas SVR applies a "soft margin" to manage limited label noise.

TABLE III
COMPARISON OF THE MODELS OVER THE CORRUPTED SYNTHETIC DATA TEST

	DBN	RBF	GRNN	BP	LSTM	ANFIS
RMSE	62.4277	37.5726	75.1519	101.4287	36.3564	67.1891
Epoch	90	50	-	100	200	100
	IT2-Sugeno	SVR	T1-PSO	ENCFIS	IT2-PSO	IT2-GBO
	23.9481	17.2526	11.5529	6.4720	8.0265	3.8761
	50	30	50	30	50	50

The top-tier models include ENCFIS [47] and the IT2-GBO-ENFIS proposed in this article. Both models are meticulously engineered for strong noise tolerance, allowing effective learning in high-noise scenarios while minimizing noise interference. The application of interval type-2 logic and GBO further enhances the performance of the latter, making IT2-ENFIS a superior solution under similar conditions compared with the previously proposed ENCFIS. The Kruskal–Wallis test [67] is also employed to ensure this conclusion is statistically robust. To perform the significance test, IT2-ENFIS and ENCFIS are each executed 20 times, producing a set of RMSE values for both models to serve as performance indicators. The null hypothesis is defined as the statement that "there is no statistical difference between the two sets of RMSE values". A p-value less than 0.05 would suggest a statistically significant difference between the two. Finally, the p-value returned from the Kruskal-Wallis test is 0.0001, leading to the rejection of the null hypothesis. This indicates that the performance of IT2-ENFIS on this dataset is statistically better.

The midlevel models include T1-PSO-ENFIS and IT2-PSO-ENFIS. These models are essentially ablation test cases for IT2-ENFIS, aimed at determining the extent to which interval type-2 logic, GBO, and Cauchy influence contribute to the model's noise-robustness. The T1-PSO-ENFIS model replaces interval type-2 logic and GBO with type-1 logic and PSO. In contrast, the IT2-PSO-ENFIS model retains interval type-2 logic but substitutes GBO with PSO. The results show that interval type-2 logic improves noise tolerance, while GBO optimization provides even more benefits, revealing the efficacy of the proposed learning approach in noisy situations. In addition, an analysis of the hyperparameter sensitivity of the model was also

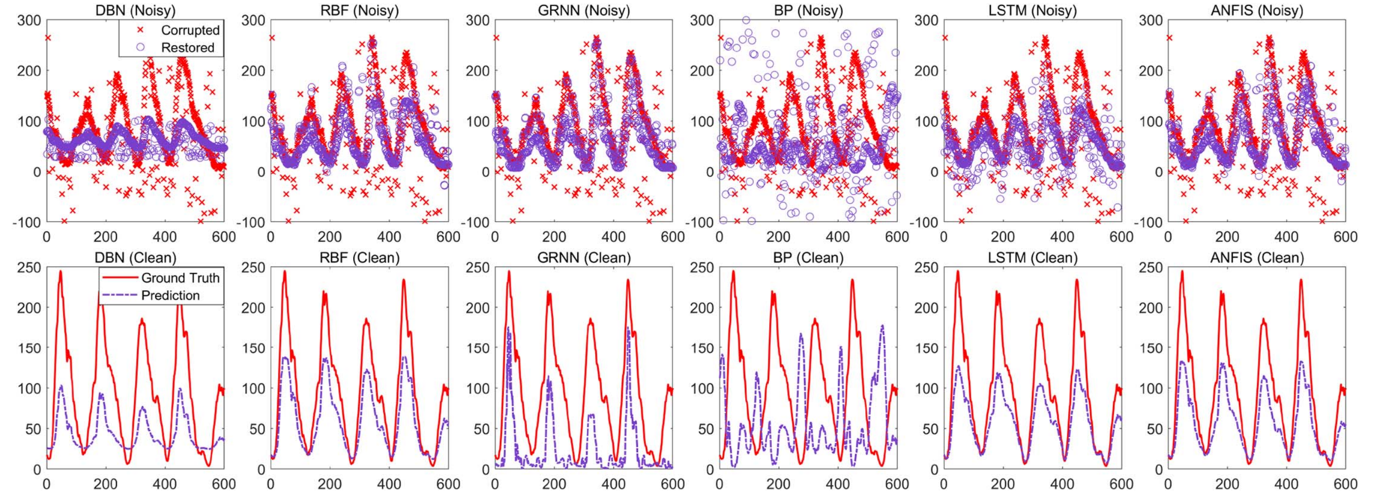

Fig. 6. Visualization of results on the contaminated sunspot time series for nonrobust models. (The results in the first row show the performance of the trained model in the training dataset (noisy), where the red crosses represent the labels of the training set, and the blue circles indicate model predictions. The second row shows the performance of the trained model on a clean, noise-free test set. The red solid line represents the actual time series, while the blue dashed line indicates the predicted time series. From this figure, the impact of noise on each model can be observed. For example, the predictions of BP exhibit a divergent trend, indicating that the model has overfitted the noise and has barely captured the features of the time series, whereas for other models, the impact of noise mainly leads to underfitting).

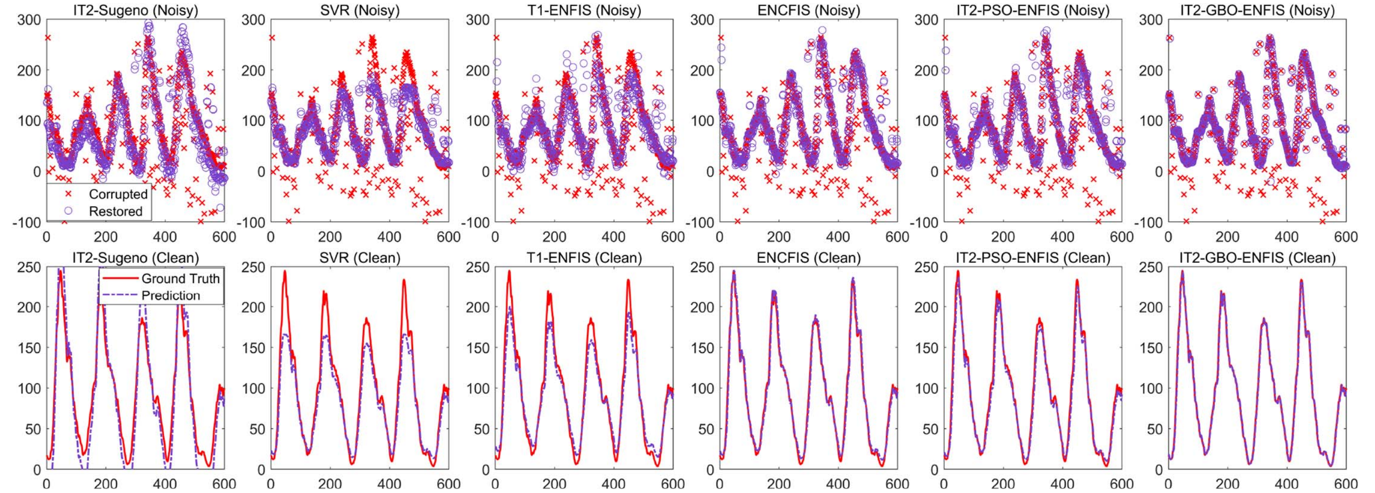

Fig. 7. Visualization of results on the contaminated sunspot time series for noise-robust models. For the meaning of the legends, please refer to Fig. 6. (This figure shows that the performance of noise-robust models is much better than that of structures without such mechanisms. Among them, IT2-Sugeno, SVR, and T1-ENFIS still suffer from slight underfitting, whereas ENCFIS, IT2-PSO-ENFIS, and IT2-GBO-ENFIS deliver solid performance).

conducted on this dataset, which indicates that even in high-noise environments, this method maintains low sensitivity to hyperparameters. Since extreme noise interference as simulated is uncommon in real-world scenarios, the proposed IT2-ENFIS model should effectively handle simpler situations.

B. Ultimate Tensile Strength of Steel Alloys Test

The ultimate tensile strength (UTS) represents the maximum engineering stress observed in a stress-strain curve, a critical property for metal materials [68]. In the metallurgical industry, ensuring that materials meet specific requirements is challenging due to the highly nonlinear and sparse relationship between the variables in the production process. In this experiment, 1500 data samples from industrial processes were selected, with 1000 for training and 500 for testing. The dataset includes 15 input dimensions: 13 numerical and 2 categorical variables. By removing variables with minimal impact on UTS, only the 10 most relevant variables remain, i.e., tempering temperature, cooling medium, sample size, test depth, category site, and the content of elements for C, Mn, Cr, Mo, and Ni. Since the data is sourced from industrial processes, it contains outliers due to measurement errors, incorrect entries, or other unknown reasons. Although limited, these outliers can affect model performance. This experiment simulates real-world scenarios to validate the trustworthiness of the model in practical applications.

The hyperparameter settings for the network are detailed in Table IV, the performance comparison of all benchmark models is presented in Table V, and the model output visualization is displayed in Fig. 8. As shown in Table V, RMSE values show

TABLE IV
IT2-ENFIS Hyperparameter Setting for UTS Test

GBO and Network		M-Estimator and Huber Loss	
Population size N	40	Tuning constant c	2.3850
Max-iteration M	50		
Number of rules	6	Huber coefficient δ	0.1

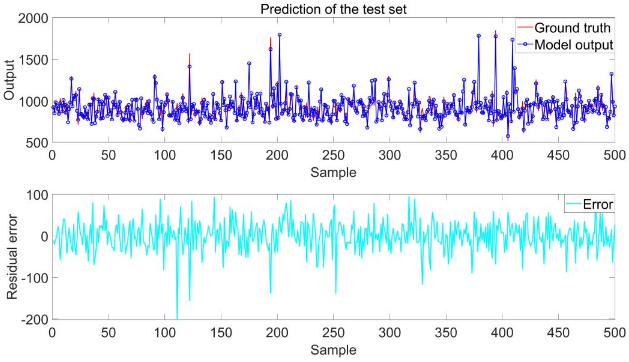

Fig. 8. Predictive output of IT2-ENFIS on UTS test set.

significant performance variation among models. Traditional regression models such as BP, RBF, GRNN, LSTM, and DBN suffer from poor performance for three main factors. First, the data distribution in industrial scenarios is often sparse, and these architectures may require denser feature representations to learn complex mapping relations from the data. Second, the dataset is small and insufficient to support deeper structures such as LSTM and DBN, making them prone to overfitting on the training set and performing poorly on the test set. Finally, unknown outliers in the data can lead to prediction shifts, a challenge that is difficult to address for models without specialized noise-resistant mechanisms. The classical type-1 ANFIS neuro-fuzzy model also performs poorly due to the limited generalization capability of type-1 fuzzy logic, which tends to overfit noise or randomness when the data is sparse. Among all the benchmark models, four achieved RMSE values below 40: SVR, IT2-Sugeno, ENCFIS, and IT2-ENFIS. SVR excels due to its vector representation, which effectively handles high dimensionality and sparsity, and its “soft margin” setting enhances its robustness to noise. IT2-Sugeno benefits from the improved generalization of interval type-2 logic, which makes it less sensitive to noise and data uncertainty. However, both methods fail to achieve better performance, as they are still affected by outliers in the data.

The ENCFIS and IT2-ENFIS models achieved the best outcomes in this test, with IT2-ENFIS performing better than the former. Both models are trustworthy learning designs, but the proposed IT2-ENFIS clearly stands out in RMSE values. The superior performance of the IT2 version of the network can be attributed to two factors: First, interval type-2 logic models both intra-individual and inter-individual uncertainties, offering superior generalization compared with type-1 logic and the complex fuzzy logic. Second, the hybrid optimization algorithm combining GBO, and the M-estimator outperforms the gradient method used by ENCFIS. This reduces noise sensitivity and brings results closer to the global optimum. Furthermore, interval type-2 logic provides significantly better interpretability than complex fuzzy logic, whose semantic explainability remains a topic of debate within the academic community. As a step toward more trustworthy machine learning, the IT2-ENFIS design proves to be highly effective. The above results suggest that IT2-ENFIS could benefit industrial scenarios with limited data and poor data quality, such as in portable or edge computing environments. Current research prefers large-scale models that require vast amounts of data, and algorithm development for smaller-scale scenarios has attracted few attentions. The proposed method can be seen as a valuable complement to mainstream research.

TABLE V
Comparison of the Models (UTS Data Test)

	BP	RBF	GRNN	LSTM	DBN
RMSE	44.4965	54.1319	56.5168	56.5765	47.7999
Epoch	200	100	-	200	50
Rule	-	-	-	-	-
	SVR	IT2Sugeno	ANFIS	ENCFIS	IT2-ENFIS
	39.5723	38.7600	45.4163	38.0137	36.1934
	30	100	50	20	50
	-	6	6	6	6

C. Spindle Thermal Compensation Dataset Test

For CNC machining, the spindle of the machine tool often experiences a temperature increase due to heat accumulation, leading to thermal expansion that affects machining precision [69]. Therefore, cooling water is required to lower the temperature, but controlling its flow rate is challenging. If the flow rate is too slow, the spindle cannot dissipate heat in time. Conversely, if the flow rate is too fast, the spindle cools too rapidly, which can cause contraction. Performing precise flow rate control is feasible, but such a complex control system would significantly increase the cost of the machine tool. An economical alternative is to use an adaptive algorithm to predict thermal error based on spindle temperature information, thus achieving effective compensation.

The spindle thermal compensation dataset contains 3000 sets of thermal displacement data at varying temperatures, with each example corresponding to readings from five different temperature sensors. During the experiment, we randomly selected 2000 samples for the training set, with the remaining 1000 used for testing. Compared with previous experiments, we further introduced extreme learning machine (ELM) [70] and Gaussian mixture model (GMM) [71] to enrich the benchmark testing and enhance the diversity of the experiments. We compared the performance of up to 12 different models in total, with the experiment divided into two parts. In the first part, the models were trained on a clean training set and tested on a clean test set to evaluate their performance in a noise-free environment. In the second part, the models were trained on a dataset containing 25% noise, but were still tested on the same clean test set to assess their learning capacity and robustness under noisy conditions. In industrial environments, spindle displacement sensors

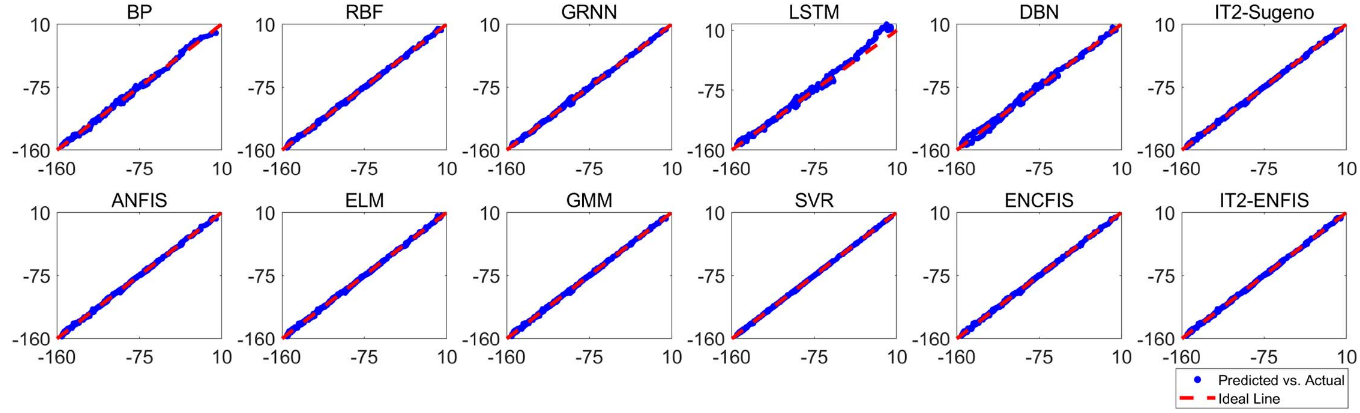

Fig. 9. Model performances in the spindle thermal compensation test (trained on clean dataset). The blue points are determined by both the predicted and actual values. The closer they are to the central ideal line, the more accurately the predictions align with reality, i.e., better model prediction.

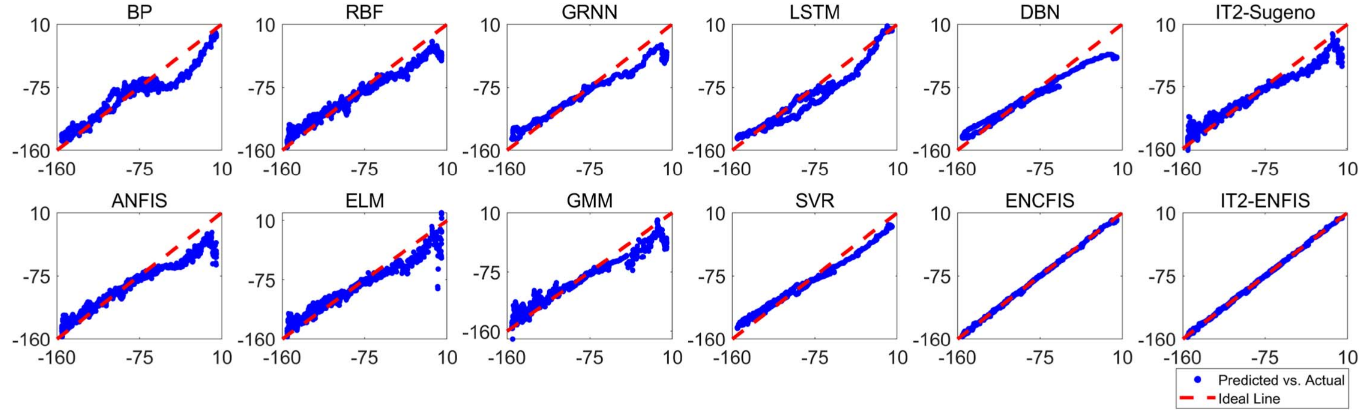

Fig. 10. Model performances in the spindle thermal compensation test (trained on noisy dataset).

TABLE VI
COMPARISON OF THE MODELS (SPINDLE THERMAL COMPENSATION TEST)

	BP	RBF	GRNN	LSTM	DBN	IT2Sugeno	ANFIS	ELM	GMM	SVR	ENCFIS	IT2-ENFIS
RMSE(clean)	2.3846	1.2005	1.5871	4.3193	3.2763	1.0113	1.1858	1.1501	1.0919	1.0590	1.0357	1.0100
RMSE(noisy)	12.9740	11.4440	11.3460	11.6839	12.1157	11.6783	13.4395	11.2369	11.9755	8.6758	1.2503	1.1007

are prone to abnormalities due to temperature changes, electromagnetic interference, vibrations, and power fluctuations. It is crucial to evaluate the performance of the model in noisy environments. The noisy set is created by replacing 25% of the data points in the original dataset with random values that fall within the same range as the original data to simulate interference that can occur in real-world scenarios.

The experimental results are visualized in Figs. 9 and 10, while the RMSE performance of each model on the clean test dataset, regardless of whether they were trained on the clean or noisy dataset, is presented in Table VI. The results show that under ideal noise-free training conditions, most models perform excellently in this dataset, with RMSE values slightly above 1, and only a few models exceed 2. However, the results tell a different story in the noisy scenario. Models, including SVR, show a significant deviation from their original performance, while only the noise-resistant architectures, ENCFIS and the proposed IT2-ENFIS, remain effective. Notably, IT2-ENFIS outperforms its predecessor ENCFIS, further validating the effectiveness of the proposed methodology and its robustness in real-world applications. In industrial applications, interference mitigation measures are typically in place, making it unlikely to encounter extreme noise scenarios like the one simulated in this experiment. Therefore, it is reasonable to infer that the IT2-ENFIS model can effectively handle most similar challenges, demonstrating high practical value.

VI. CONCLUSION AND FUTURE WORK

In this article, a trustworthy learning architecture for numerical regression tasks is proposed. Inspired by the ENCFIS model, it employs interval type-2 fuzzy logic and uses a novel metaheuristic optimization method, GBO, to optimize the fuzzy antecedent network, while the Cauchy M-estimator updates the consequent network. This approach enhances noise resistance, and yields results that are closer to the global optimum compared with ENCFIS. Meanwhile, the ability of interval type-2 logic to model both intra-individual and inter-individual uncertainties provides IT2-ENFIS with excellent data adaptability and generalization capabilities. Compared with the ENCFIS model, which uses complex fuzzy logic, the newly proposed model significantly outperforms in interpretability. The experimental results also confirm that the proposed IT2-ENFIS demonstrates remarkable performance, whether dealing with problems involving significant artificial noise or real-world datasets with moderate noise levels. These factors collectively make IT2-ENFIS a superior trustworthy learning solution compared with ENCFIS, as it excels in all four main indicators of trustworthiness of a learning model, i.e., generalization, interpretability, robustness, and fairness.

Future work may focus on addressing two critical questions: first, how can we enhance the performance of the primary network? Second, how can we handle more challenging noise types, such as dynamic noise and adversarial perturbations? Regarding the first question, introducing general type-2 logic into trustworthy learning frameworks could be beneficial, as it provides better generalization capabilities vis a vis interval type-2 logic. Another approach is to increase the network depth, which would enhance its nonlinear mapping capabilities and noise tolerance, ultimately leading to improved performance. The second question, however, is particularly challenging. To neutralize dynamic noise, the algorithm must support online learning and be capable of suppressing noise without prior knowledge of it. Achieving this would require a redesign of the system and the introduction of a fundamentally new noise-countering strategy. Additionally, adversarial noise, as a carefully designed perturbation to sabotage model training, presents another compelling area of study. This type of noise is often hard to detect but can be highly disruptive. Existing methods for mitigating adversarial noise rely mainly on intricate black-box mechanisms. In contrast, interpretable models provide a promising perspective, enabling more targeted and efficient strategies to combat such anomalies.

Finally, we will also apply this model to a broader range of real-world scenarios to assess its practical value. Potential application areas include medicine, industry, decision-making and so on. In the medical field, where human health and well-being are directly involved, conventional machine learning models often fail to pass ethical reviews due to insufficient trustworthiness. With its unique characteristics, IT2-ENFIS shows significant advantages in medical applications. In industrial scenarios, machine learning methods also struggle to compete with mechanism-driven approaches due to safety concerns arising from their unverifiable black-box design. The trustworthy learning solution holds significant potential in such cases. In the decision-making domain, its advantages are even more pronounced, as full interpretability ensures transparent and accountable data-driven decision-making, leading to a transformative impact on the field. Future research will investigate the application of the proposed model across each of the scenarios mentioned above.

References

[1] M. Chui et al., "The economic potential of generative AI: The next productivity frontier," 2023. Accessed: Jun. 25, 2024. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier#introduction

[2] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. 4th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991, pp. 831–838.

[3] T. J. VanderWeele and I. Shpitser, "On the definition of a confounder," *Ann. Stat.*, vol. 41, no. 1, pp. 196–220, Feb. 2013.

[4] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *Proc. Comput. Vis. (ECCV)*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 158–171.

[5] N. Joshi, X. Pan, and H. He, "Are all spurious features in natural language alike? an analysis through a causal lens," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Assoc. for Comput. Linguistics, Dec. 2022, pp. 9804–9817. Accessed: Aug. 11, 2024. [Online]. Available: https://aclanthology.org/2022.emnlp-main.666

[6] L. Chen, M. Zaharia, and J. Zou, "How is ChatGPT's behavior changing over time?" *Harvard Data Sci. Rev.*, vol. 6, no. 2, Mar. 12 2024. Accessed: Aug. 11, 2024. [Online]. Available: https://hdsr.mitpress.mit.edu/pub/y95zitmz

[7] H. Liu, M. Chaudhary, and H. Wang, "Towards trustworthy and aligned machine learning: A data-centric survey with causality perspectives," 2023, *arXiv:2307.16851*. [Online]. Available: https://api.semanticscholar.org/CorpusID:260333892

[8] B. Li et al., "Trustworthy AI: From principles to practices," *ACM Comput. Surv*, vol. 55, no. 9, pp. 1–46, Jan. 2023.

[9] I. Goodfellow, Y. Bengio, and A. Courville, "Machine learning basics," in *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016, pp. 98–164.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 86–94.

[11] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," 2016, *arXiv:1606.06565*.

[12] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Inf. Fusion*, vol. 76, pp. 89–106, Dec. 2021. Accessed: Sep. 6, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001093

[13] N. Antunes, L. Balby, F. Figueiredo, N. Lourenco, W. Meira, and W. Santos, "Fairness and transparency of machine learning for trustworthy cloud services," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, 2018, pp. 188–193.

[14] M. Du, F. Yang, N. Zou, and X. Hu, "Fairness in deep learning: A computational perspective," *IEEE Intell. Syst.*, vol. 36, no. 4, pp. 25–34, 2021.

[15] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, 2020. Accessed: Aug. 17, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520306241

[16] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in Statistics: Methodology and Distribution*, S. Kotz and N. L. Johnson, Eds., New York, NY: Springer New York, 1992, pp. 492–518, doi: 10.1007/978-1-4612-4380-9_35.

[17] D. de Menezes, D. Prata, A. Secchi, and J. Pinto, "A review on robust m-estimators for regression analysis," *Comput. Chem. Eng.*, vol. 147, 2021, Art. no. 107254. Accessed: Sep. 2, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135421000326

[18] L. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965. Accessed: Aug. 5, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001999586590241X

[19] B. Sanjaa and P. Tsoozol, "Fuzzy and probability," in *Proc. Int. Forum on Strategic Technol.*, 2007, pp. 141–143.

[20] M. Beer, "Fuzzy probability theory," in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, Ed. New York, NY: Springer New York, 2009, pp. 4047–4059. Accessed: Aug. 23, 2024. [Online]. Available: https://doi.org/10.1007/978-0-387-30440-3_237

[21] D. V. Lindley, "A statistical paradox," *Biometrika*, vol. 44, no. 1/2, pp. 187–192, 1957. Accessed: Aug. 21, 2024. [Online]. Available: http://www.jstor.org/stable/2333251

[22] G. E. P. Box, "Science and statistics," *J. Am. Stat. Assoc.*, vol. 71, no. 356, pp. 791–799, 1976. [Online]. Available: http://www.jstor.org/stable/2286841

[23] E. Nagel and J. R. Newman, “Concluding reflections,” in *Godel’s Proof*, Manhattan, NY, USA: NYU Press, 2001, pp. 109–113. [Online]. Available: http://www.jstor.org/stable/j.ctt9qg4j9.12

[24] L. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—i,” *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020025575900365

[25] J. Mendel and R. John, “Type-2 fuzzy sets made simple,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 117–127, Apr. 2002.

[26] J. M. Mendel, “Type-2 fuzzy sets including word models,” in *Explainable Uncertain Rule-Based Fuzzy Systems*, 3rd Edition, Cham, Switzerland: Springer Int. Publishing, 2024, pp. 237–280, doi: 10.1007/978-3-031-35378-9_6.

[27] N. Karnik, J. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 643–658, Dec. 1999.

[28] Q. Liang and J. Mendel, “Interval type-2 fuzzy logic systems: theory and design,” *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, Dec. 2000.

[29] E. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *Int. J. Man Mach. Stud.*, vol. 7, no. 1, pp. 1–13, 1975. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020737375800022

[30] M. Sugeno, “An introductory survey of fuzzy control,” *Inf. Sci.*, vol. 36, no. 1, pp. 59–83, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002002558590026X

[31] M. Sugeno and G. Kang, “Structure identification of fuzzy model,” *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15–33, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0165011488901133

[32] S. Galichet, R. Boukezzoula, and L. Foulloy, “Words or numbers, mamdani or sugeno fuzzy systems: A comparative study,” in *Uncertainty and Intelligent Information Systems, Chapter 21*, pp. 291–305, 2008. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9789812792358_0021

[33] M. B. Begian, W. W. Melek, and J. M. Mendel, “Stability analysis of type-2 fuzzy systems,” in *Proc. IEEE Int. Conf. Fuzzy Syst. (IEEE World Congr. Comput. Intell.),* 2008, pp. 947–953.

[34] J. M. Mendel, “Interval type-2 fuzzy systems,” in *Explainable Uncertain Rule-Based Fuzzy Systems.* Cham, Switzerland: Springer Int. Publishing, 2024, pp. 385–451, doi: 10.1007/978-3-031-35378-9_9.

[35] J. H. Aladi, C. Wagner, and J. M. Garibaldi, “Type-1 or interval type-2 fuzzy logic systems — on the relationship of the amount of uncertainty and FOU size,” in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, 2014, pp. 2360–2367.

[36] B. Han et al., “A survey of label-noise representation learning: Past, present and future,” 2020, *arXiv:2011.04406*.

[37] J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer,” in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–7. Accessed: Sep. 5, 2024. [Online]. Available: https://openreview.net/forum?id=H12GRgcxg

[38] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2233–2241.

[39] Y. Wang, A. Kucukelbir, and D. M. Blei, “Robust probabilistic modeling with bayesian data reweighting,” in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3646–3655.

[40] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *Proc. 35th Int. Conf. Mach. Learn., Proc. Mach. Learn. Res. (PMLR)*, J. Dy and A. Krause, Eds., vol. 80, Jul. 2018, pp. 4334–4343. Accessed: Aug. 27, 2024. [Online]. Available: https://proceedings.mlr.press/v80/ren18a.html

[41] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, “MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in *Proc. 35th Int. Conf. Mach. Learn. (PMLR)*, vol. 80, pp. 2304–2313, Jul. 2018.

[42] B. Han et al., “Co-teaching: robust training of deep neural networks with extremely noisy labels,” in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 8536–8546.

[43] Z. Wang, G. Hu, and Q. Hu, “Training noise-robust deep neural networks via meta-learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 4523–4532.

[44] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi, “Unlabeled data improves adversarial robustness,” in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 11192–11203.

[45] Z. Deng, D. Li, J. He, Y.-Z. Song, and T. Xiang, “Generative model based noise robust training for unsupervised domain adaptation,” *arXiv:2303.05734*, 2023. Accessed: Sep. 5, 2024. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr2303.html#abs-2303-05734

[46] B. Na et al., “Label-noise robust diffusion models,” in *Proc. 12th Int. Conf. Learn. Representations*, 2024. Accessed: Sep. 5, 2024. [Online]. Available: https://openreview.net/forum?id=HXWTXXtHNl

[47] C. Xue and M. Mahfouf, “ENCFIS: An exclusionary neural complex fuzzy inference system for robust regression learning,” *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 3, pp. 1539–1552, Mar. 2024.

[48] D. Ramot, R. Milo, M. Friedman, and A. Kandel, “Complex fuzzy sets,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 171–186, Apr. 2002.

[49] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, “Complex fuzzy logic,” *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 450–461, 2003.

[50] S. Dick, “Toward complex fuzzy logic,” *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 405–414, 2005.

[51] C. Xue and M. Mahfouf, “Racfis: A new rapid adaptive complex fuzzy inference system for regression modelling,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 1238–1252, 2024.

[52] J. E. D. Jr. R. E. Welsch, “Techniques for nonlinear least squares and robust regression,” *Commun. Statist. - Simul. Comput.*, vol. 7, no. 4, pp. 345–359, 1978, doi: 10.1080/03610917808812083.

[53] D. Wu, “Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2012, pp. 1–8.

[54] R. Koenker, “Fundamentals of quantile regression,” in *Quantile Regression, Ser. Econometric Society Monographs*, Cambridge, U.K.: Cambridge Univ. Press, 2005, pp. 26–67.

[55] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A comprehensive survey of loss functions in machine learning,” *Ann. Data Sci.*, vol. 9, no. 2, pp. 187–212, Apr. 2022.

[56] M. R. Bonyadi and Z. Michalewicz, “Particle swarm optimization for single objective continuous space problems: A review,” *Evol. Comput.*, vol. 25, no. 1, pp. 1–54, 2017.

[57] O. B. Sheynin, “Studies in the history of probability and statistics. xxv. on the history of some statistical laws of distribution,” *Biometrika*, vol. 58, no. 1, pp. 234–236, 1971.

[58] “Sunspot index and long-term solar observations (SILSO).” Accessed: July 5, 2024. [Online]. Available: https://www.sidc.be/silso/datafiles

[59] G. Hinton, “Deep belief nets,” in *Encyclopedia of Machine Learning*, 2010, pp. 267–269, doi: 10.1007/978-0-387-30164-8_208.

[60] J. Moody and C. J. Darken, “Fast learning in networks of locally-tuned processing units,” *Neural Comput.*, vol. 1, no. 2, pp. 281–294, 1989.

[61] D. Specht, “A general regression neural network,” *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[63] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[64] J.-S. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[65] D. S. Mai, T. H. Dang, and L. T. Ngo, “Optimization of interval type-2 fuzzy system using the PSO technique for predictive problems,” *J. Inf. Telecommun.*, vol. 5, no. 2, pp. 197–213, 2021.

[66] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” in *Proc. 9th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Cambridge, MA, USA: MIT Press, 1996, pp. 155–161.

[67] W. H. Kruskal and W. A. Wallis, “Use of ranks in one-criterion variance analysis,” *J. Am. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952. Accessed: Sep. 7, 2024. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441

[68] Huda, Z. “Mechanical testing and properties of materials,” in *Mechanical Behavior of Materials: Fundamentals, Analysis, and Calculations*, Cham, Switzerland: Springer Int. Publishing, 2022, pp. 39–61, doi: 10.1007/978-3-030-84927-6_3.

[69] Y. Li, W. Zhao, S. Lan, J. Ni, W. Wu, and B. Lu, “A review on spindle thermal error compensation in machine tools,” *Int. J. Mach. Tools Manuf*, vol. 95, pp. 20–38, 2015. Accessed: Feb. 18, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0890695515300298

[70] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, “A review on extreme learning machine,” *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 611–660, Dec. 2022, doi: 10.1007/s11042-021-11007-7.

[71] D. Reynolds, *Gaussian Mixture Models*, Boston, MA, USA: Springer US, 2009, pp. 659–663, doi: 10.1007/978-0-387-73003-5_196.