
Article:
Xue, C. orcid.org/0000-0003-0311-5850 and Mahfouf, M. orcid.org/0000-0002-7349-5396 (2023) RACFIS: A new rapid adaptive complex fuzzy inference system for regression modelling. IEEE Transactions on Emerging Topics in Computational Intelligence. ISSN 2471-285X

https://doi.org/10.1109/tetci.2023.3343988

RACFIS: A New Rapid Adaptive Complex Fuzzy Inference System for Regression Modelling

Chuan Xue and M. Mahfouf

***Abstract*—The theory of complex fuzzy sets has made great breakthroughs in recent times. Complex fuzzy theory (CFT) allows a fuzzy set to include more information with the help of its two-dimensional rule-base, which is of great potential for improving the related fuzzy system performance while managing the size of the associated rule-bases. In this paper, a new rapid adaptive complex fuzzy inference system (RACFIS) is proposed by redesigning the optimization policy of the earlier complex neuro-fuzzy system (CNFS) algorithm. Improvements include a new three-parameter Quasi-hyperbolic momentum (QHM) optimization method to replace the original particle swarm optimization (PSO), and unsupervised learning is introduced, for the first time, into the complex neuro-fuzzy model to pre-train the antecedent parameters for a better global optimum as well as a faster convergence. Experimental results show that RACFIS performs very well on all data sets, obtaining excellent prediction accuracies with on average 10 times lower epoch numbers (as compared with all benchmark models) and a reduction in the size of the rule-bases by nearly 20%∼30% (as compared with non-complex fuzzy models). RACFIS also possesses a strong generalization capability that outperforms all the benchmark algorithms employed in the simulation experiments. In addition, a mean impact value (MIV) algorithm based on radial basis function (RBF) network is also employed to select variables with higher relevance in order to mitigate the drawbacks caused by the higher dimensionality of the data.**

***Index Terms*—Complex fuzzy inference system (CFIS), Adaptive neuro-fuzzy systems, Online learning, Numerical prediction.**

I. Introduction

THE modelling of real-world data has always been a challenging problem. Unlike those popular data sets that are purposefully selected as examples to test the performance of specific algorithms, data sets which users are more likely to encounter in scientific research or industry are often not so ideally presented. Real-world data sets, more often than not, can lead to processing difficulties, including redundancy, imbalance and incompleteness of the data, unrelated variables, missing dimensions, human entry errors or measurement errors, etc., which all hinder the efficient implementation of machine learning algorithms. In such scenarios, it is sometimes impossible for users to know where the missing part and/or the wrong information exactly are for the purpose of manual modification. For this reason, the generalization capability of an algorithm is the crucial element to be relied upon to overcome such obstacles. Moreover, regression tasks, such as data analysis and prediction in various application areas, lead to some real-time requirements. A heavy and sophisticated algorithm which usually indicates a dramatic increase in hardware costs is less likely to be the ideal solution that can be widely accepted. Occasionally, researchers also hope that the model has a certain degree of interpretability so that they can adjust the errors in an easier way or even extract expert knowledge from the model. Fuzzy systems have overt advantages in this respect.

Nowadays, self-learning neuro-fuzzy systems have become mainstream due to the increasing complexity of real-world scenarios. Such models are not complicated from a classification viewpoint. According to the network structure, the neuro-fuzzy inference systems (NFIS) that have been hitherto proposed include static NFIS such as ANFIS [1], self-organized NFIS such as SONFIN [2] and, deep NFIS [3], where the word "static" refers to a network structure that is fixed and will not change with the training process, whereas the structure of the self-organizing architecture may adjust itself to adapt to the data. As for deep NFIS, it can be understood as the expansion of deep learning in the fuzzy systems area. From the perspective of fuzzy logic, the category of NFIS includes type-1 NFIS, general type-2 NFIS [4] and interval type-2 NFIS [5], and NFIS using other fuzzy logics. As far as defuzzification methods are concerned, Mamdani-NFIS are using Mamdani defuzzification [6] and TSK-NFIS uses Sugeno defuzzification [7], [8]. In recent years, fuzzy models based on quantum computation [9], [10] have also emerged.

It is widely accepted that apart from increasing the depth, the most promising directions of improving NFIS are of two fold, namely the fuzzy logic that is applied, and the optimization algorithm. In terms of the fuzzy logic aspect, type-1 NFIS models that are first to emerge have overt drawbacks because of the limited expressive ability of type-1 rules. To describe a complex problem, a relatively large rule-base is often required, which makes it extremely inefficient when dealing with real-world problems with many variables. Further, since type-1 logic can only represent a relatively superficial information, this often leads to overfitting and poor generalization capability of the model. Subsequently, type-2 fuzzy logic [11], [12], especially interval type-2 logic [13], has been introduced to build NFIS, which significantly improves accuracy, rule-base size, and the generalization capability of NFIS models. Nevertheless, in recent years, the related architectures of type-2 fuzzy logic are also close to optimum. If further improvements of the performance are required, then it is very necessary to

This paper was produced by Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD (UK)

Manuscript received ?; revised ?.

introduce a more powerful fuzzy logic.

Regarding the optimization policies of NFIS, there are largely two main categories [14]: backpropagation and hybrid optimization. Currently, the strategy of backpropagation training with a single method is considered suboptimal. A hybrid optimization usually means optimizing the antecedent and consequent parameters separately. Given that the antecedent part includes nonlinear fuzzy membership functions, whereas the consequent is only linear mappings, hybrid approaches can help facilitate faster training and obtain a better result. Methods for the consequent part training are relatively simple, including RLS [15], extended Kalman filter [16], ELM [17] or SVM (for classification purposes) [18]. For the nonlinear antecedent part, gradient-based methods were favored in the early years, but recently researchers have become more biased towards population methods, such as particle swarm [19], evolutionary algorithms [20] or artificial bee colonies [21]. Blindly opting for a derivative-free method may not be wise because the gradient is actually the prior knowledge of the optimization surface. Population-based optimization solutions often suffer from higher training complexity and lower speed due to the absence of this knowledge. Such a train of thought is unfavorable to real-world application scenarios that pursue concomitantly accuracy and efficiency.

In recent years, breakthroughs in multi-parameter gradient-momentum methods have improved the performance of gradient optimization [22]. In the meantime, the theory of complex fuzzy sets (CFT) [23]–[26] also brings new possibilities to improve NFIS. CFT is considered a breakthrough after the type-2 fuzzy theory. It extends the common domain of fuzzy membership to the entire unit circle in the complex plane, allowing a single rule to accommodate more information over previous fuzzy logic. Such a property is conducive to narrowing the system rule-base and improving the generalization performance. Complex fuzzy logic can describe problems that the previous fuzzy logic is able to, such as wave-particle duality in quantum mechanics, which helps to expand application scenarios of fuzzy systems. The frequency-domain nature of complex numbers also makes CFT more effective in dealing with periodicity. Also, the so-called "curse of dimensionality" [27] often leads to a sharp increase in computational complexity, the sparseness of information, and the weakening of non-linear connections between data points. Thus, a purposely designed mechanism capable of more accurately excluding insignificant variables is required. According to the above perspectives, we propose such an algorithm in this paper.

This paper mainly proposes a new rapid adaptive complex neural-fuzzy system (RACFIS) that learns from real-world data sets with a fast, accurate, and robust regression performance. Note that the word "adaptive" in this definition conveys the same idea as in adaptive neural-fuzzy inference system (ANFIS), meaning, self-learning. Such an algorithm utilizes complex fuzzy logic to improve the model performance against periodicity and stochasticity and also strengthen generalization capability while reducing the size of the rule-base. Regardless of the difficulty of understanding two-dimensional complex fuzzy logic due to its yet-to-decipher semantic, this network model is still as transparent as other neural-fuzzy architectures, which may be useful for some specific purposes, such as debugging and parameter tuning. Bisecting K-Means clustering (unsupervised learning) [28], Quasi-hyperbolic momentum [29], and RLS estimation together as a joint optimization strategy, assuring that the network can quickly converge to a better global minimum. In terms of excluding non-critical variables, the mean impact value (MIV) [30] algorithm based on the RBF network, which has proved to be an effective method for screening variables, is applied. Finally, the algorithm is tested on a synthetic dataset, a Sunspot time series dataset, and two metallurgical datasets which are highly dimensional, non-linear, and sparse. Experimental results show that the performance of RACFIS overtly exceeds other well-known benchmark algorithms.

The remainder of this article is organized as follows: in Section II, a review of related work on complex fuzzy sets theory and existing complex neural-fuzzy models is given. In Section III, the details of RACFIS including network structure, key variable selection algorithm and optimization method are outlined. In the fourth section the implementation of the proposed algorithm is reviewed and the results of experiments are presented, analysed and discussed. In Section V, conclusions are drawn and future vectors for research are outlined.

II. Related Work

A. Complex Fuzzy Theory

The prototype of the complex fuzzy theory (CFT) was proposed almost as early as the interval type-2 fuzzy logic. Kaufman [31], Moses [32] and Nguyen [33] successively mentioned similar concepts in their works during the period from 1985 to 2000. Despite this, what finally attracted researcher's attention around fuzzy systems and made this topic widely recognized as a new theory of interest was the article from Ramot *et al.* [34]. Inspired by the wave-particle duality of microscopic particles in quantum mechanics, Ramot extended the concept of fuzzy membership into the complex plane, defining the modulus of the complex number as the value of the complex membership, while designating the phase as the so-called "context". Hence, the complex fuzzy grade in Ramot's CFT is denoted as follows:

$$\phi_s(x) = \varphi_s(x) \cdot e^{j\omega_s(x)}, j = \sqrt{-1}, \tag{1}$$

where $\varphi_s(x)$ and $\omega_s(x)$ are respectively the modulus and the phase of the membership function. According to the definition, the degree of membership is defined in the unit circle of the complex plane, which delimits $\varphi_s(x) \in [0, 1]$, whereas $\omega_s(x)$ can be arbitrary value. This theory also indicates that the ordinary type-1 fuzzy set is a special case where the phase component of the complex fuzzy set is zero. As for the fuzzy inference logic of above complex fuzzy sets, Ramot *et al.* made a more detailed explanation in the subsequent article [35]. Among this, some of the most important operators for complex fuzzy reasoning are illustrated. Assume that there are two complex fuzzy sets A and B within the universe of discourse U, where set A is the aggregation of a series of sets $A_1, A_2, A_3, \cdots, A_n$. The operators for complement,

union, intersection, and aggregation are given respectively as equation (2) (3) (4) (5) as follows:

$$\phi_{\bar{A}}(x) = C[\varphi_A(x)] \cdot e^{(j\omega_{\bar{A}}(x))}, \tag{2}$$

$$\phi_{A \bigcup B}(x) = [\varphi_A(x) \bigoplus \varphi_B(x)] \cdot e^{(j\omega_{A \bigcup B}(x))}, \tag{3}$$

$$\phi_{A \bigcap B}(x) = [\varphi_A(x) \star \varphi_B(x)] \cdot e^{(j\omega_{A \bigcap B}(x))}, \tag{4}$$

$$\begin{aligned} \phi_A(x) &= aggregate[\phi_{A_1}(x), \phi_{A_2}(x), \ldots, \phi_{A_n}(x)] \\ &= \sum_{i=1}^{n} \omega_i \varphi_{A_i}(x), \end{aligned} \tag{5}$$

where $\bigoplus$ is said to be t-conorm, $\star$ is the representation of t-norm. In fuzzy mathematics, a t-norm denotes intersection in a lattice and conjunction in logic, whereas a t-conorm behaves as a disjunction logic or a union operator. $\omega_{A \bigcup B}(x)$ and $\omega_{A \bigcap B}(x)$ can be calculated in numerous ways, such as the sum of $\omega_A(x)$ and $\omega_B(x)$, the maximum or the minimum between these two, etc. The aggregation operator is in fact the inner product of the vector, i.e., Ramot's complex fuzzy membership degree of its essence is a vector.

Dick [36] introduced the notion of lattice [37] into the definition of the theory of complex fuzzy sets. According to Dick's interpretation, the complex fuzzy sets are divided into two categories, i.e., the ones with rotational invariance and the ones without rotational invariance. A complex fuzzy set is considered as rotationally invariant if and only if function $L : \Gamma \times \Gamma \to \Gamma, L(pe^{j\alpha} \cdot e^{jk}, qe^{j\beta} \cdot e^{jk}) = e^{jk} \cdot L(pe^{j\alpha}, qe^{j\beta})$ stands for all elements $pe^{j\alpha}, qe^{j\beta} \in \Gamma$, in which Γ is the lattice where this CFS belongs to. Dick also noticed that possible candidates for implication operators, including conjunction, disjunction, and negation, are conspicuously restricted if a CFS is rotationally invariant. For instance, the algebraic product should not be used as the conjunction operator for a CFS with the rotational invariance since the algebraic product does not satisfy the definition of rotational invariance. Instead, when it comes to a CFS without the rotational invariance, algebraic product may well apply for conjunction operations.

Following this, many researchers have since made futher valuable contributions in this area as well. Tamir [38], [39] proposed their own version of CFT which is called the pure complex fuzzy set. Alkouri *et al*. [40] developed the notion of complex Atanassov's intuitionistic fuzzy set (CAIFS). Yager [41], [42] created the idea of Pythagoras fuzzy set (PFS). Kumar *et al*. [43] put forward a complex intuitionistic fuzzy soft set (CIFSS) as an attempt to extend the original intuitionistic fuzzy soft set (IFSS) mentioned by Maji [44]. Thirunavukaras *et al*. [45] proposed the notion of a parameterized complex fuzzy soft set which is an upgrade of its real number counterpart. Li *et al*. [46], [47] investigated the property of sphere complex fuzzy sets, in which the truth-valuation domain is a high-dimensional hypersphere, enabling the modeling of multiple paralleling variables at a time. Ali *et al*. [48], [49] further expanded the concept of PFS by developing a 3-element membership function composed of membership, non- membership and indeterminacy, which leads to a neutron-sophic set for decision-making. Liu *et al*. [50] put forward the definition of a complex q-rung orthopair fuzzy set for multivariate and multiple feature group decision-making. Singh [51] has created a bipolar δ-equal complex fuzzy set to describe the inconsistency as well as completeness in real-world data. Greenfield [52] defined an interval valued complex fuzzy set by introducing the concept of type-2 fuzzy sets.

B. Complex Neural-Fuzzy Systems

Man *et al*. [53] proposed a six-layer neuro-fuzzy system which is considered to be the earliest complex fuzzy inference architecture, by replacing type-1 fuzzy logic with complex fuzzy logic in a single-input ANFIS, encouraged a debate over Adaptive Neuro Complex Fuzzy Inference System (ANCFIS). Chen *et al*. [54] further improved this concept, by structuring a relatively complete ANCFIS framework. Since the architecture is defined to solve the prediction of quasi-periodic problems, a sinusoidal fuzzy membership function is employed for this purpose, which is shown as follows:

$$r(\theta) = d\sin(a\theta + b) + c, \tag{6}$$

where the parameters a, b, c, d are said to be premise parameters. Note that the sinusoidal membership function can only be used to deal with time series data that has a certain periodic regularity. To match this function from the aspect of signal processing, three measurement operations are also recommended which are the Euclidean distance calculated through the L-2 norm, the convolution, and the Elliot function, respectively. Among the three alternatives, the Elliot function is considered to be the best option:

$$\frac{z\sum_{k=1}^{n} x_k \cdot [r_i(\theta_k)\cos(\theta_k) + jr_i(\theta_k)\sin(\theta_k)]}{1 + |\sum_{k=1}^{n} x_k \cdot [r_i(\theta_k)\cos(\theta_k) + jr_i(\theta_k)\sin(\theta_k)]|}, \tag{7}$$

where $\theta_k = k\frac{2\pi}{n}$,and n represents the number of elements inside the input vector. In the following years, Yazdanbakhsh *et al*. [55]–[59] studied this architecture further, including using ANCFIS for solar power prediction [55]; a tool for time-series prediction [56]; combining ANCFIS with extreme learning machine [57]; a modification of ANCFIS to allow it to be applied in multi-input-multi-output (MIMO) tasks [58]; and a fast adaptive version of ANCFIS which is named FANCFIS [59]. Yeganejou *et al*. successfully developed a classifier based on ANCFIS for condition monitoring [60]. For the most part, these works mainly focus on the application field, with limited improvements to the network structure itself. It is worth noting that, the characteristics of the sinusoidal membership function can only be used in the case of periodic data or problems with cyclical behaviour, such as time series forecasting, which greatly constrain the scale of application.

Li *et al*. [61], [62] created another strand of complex neural-fuzzy system based on a novel complex Gaussian fuzzy membership function, which is shown as follows:

$$\mu Gaussian(x, m, \sigma) = \exp\left[-0.5(\frac{x-m}{\sigma})^2\right]$$

$$- f(\frac{x-m}{\sigma^2}) \exp\left[-0.5(\frac{x-m}{\sigma})^2\right] j, \tag{8}$$

where x is said to be the base variable, m is the mean of a gaussian function while σ is the spread. Thus, the parameter set for this membership function is $\{m, \sigma\}$, which is also the premise parameter set for a single node. Compared with the sinusoidal membership function used in ANCFIS, the Gaussian complex membership function here can play the role of a 'universal approximator' for almost all purposes, rather than only for periodicity forecasting. For the objective CNFS, the complex fuzzy inference system is cast into a five-layered network if the input layer is not counted as one, which is the same as the classical ANFIS structure. With the help of the Particle Swarm Optimization (PSO)-Recursive Least Square Estimating (RLS) hybrid optimization algorithm that applies PSO algorithm to optimize the antecedent parameters and the RLS algorithm to optimize the subsequent parameters, the network can achieve excellent nonlinear approximation performance. It has been confirmed [61], [62] that the CNFS architecture is effective in its adaptive capability than its type-1 fuzzy logic counterpart. The dual-output attribute ensures the approximation of two functions concurrently in a CNFS network, which enhances its ability to solve complicated problems with multiple principles. Not the least of which, although only very few fuzzy rules are employed within the network, CNFS can still have a better performance than most of the similar architecture which only applies type-1 fuzzy logics. At present, this design has been successfully used in several application scenarios, such as image restoration [63], image noise canceling [64], knowledge discovery [65], time series forecasting [66], ARIMA forecasting [67] and multi-class prediction [68]. However, the deficiency of this algorithm is also notable. The derivative-free optimization method, such as PSO or artificial bee colony, is not efficient enough. Compared with gradient methods, these derivative-free algorithms usually require more computing resources, and randomness will inevitably exist in experimental results, which often leads to multiple experiments needed to obtain ideal results. This is unacceptable for applications with real-time requirements. In addition, such architecture is rather sensitive to the initial values of the antecedent parameters. If the outcome of random initialization happens to be inappropriate, the network may fail to converge or may converge to a local optimum. In other words, CNFS lacks a mechanism to effectively pre-determine antecedent parameters, which further increases the uncertainty of the results.

In fact, apart from the above two algorithms, other researchers have as well proposed neuro-fuzzy systems using complex-valued fuzzy membership functions, inclusive of ACNFIS [69] as proposed by Shoorangiz *et al.*, CNFIS [70] defined by Subramanian *et al.* and CVNF [71] investigated by Ryusuke *et al.* Note that despite the fact that there are "complex" and "fuzzy" concepts in the names of these architectures, they are not however essentially complex fuzzy neuro-fuzzy systems. The key aspect is that the complex fuzzy logic was never applied as operating rules to such models, and where only the form of the complex fuzzy membership is taken to reduce the size of the network to construct a more compact system. In other words, they are still variants of the type-1 algorithm, which cannot be considered to be complex neuro-fuzzy systems in the strict definition of the word "complex".

III. Methodology of the Proposed Algorithm

In this section, the architecture of the network, the unsupervised parameters that pre-train the method, the optimization strategy, and the variable screen algorithm are detailed. As will be explained, RACFIS that subjects to the complex fuzzy logic basically inherits the network structure of CNFS, and employs a similar complex Gaussian membership function. Assume the objective network has k inputs and one output, then the ith fuzzy rule can be represented as follows:

Rule i : IF l_1 is $A_1^i(x_1)$ and l_2 is $A_2^i(x_2), \ldots,$ and l_k is $A_k^i(x_k)$.

According to the defuzzification strategy of Sugeno method, the output z^i of this rule is as follows:

$$z^i = a_0^i + \sum_{j=1}^{k} a_j^i x_j, i = 1, 2, \ldots, n, \tag{9}$$

where$A_j^i(x_j)$ is the jth antecedent of the ith fuzzy rule,a_j^i is the corresponding consequent parameter. However, the non-derivative optimization method from the original CNFS is ignored, and the recently proposed three-parameter gradient momentum method, i.e., Quasi-hyperbolic momentum, is used instead to optimize the antecedent parameters. Regarding the consequent part, the recursive least square is taken as the optimal policy. Moreover, Bisecting K-Means as an unsupervised learning method is also applied to pre-train the antecedent parameters, while the RBFN-based MIV algorithm is utilized to select the variables. A general schematic diagram of the RACFIS is given in Fig. 1.

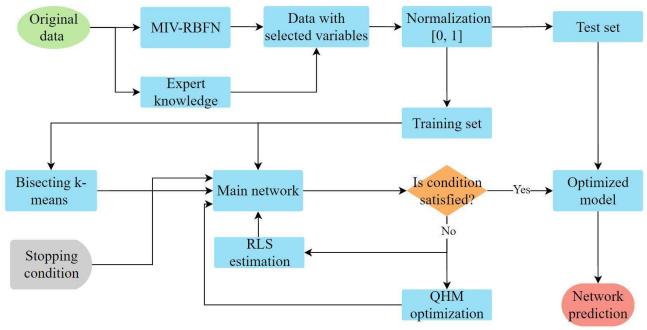

Fig. 1. RACFIS algorithm flow.

A. Network Structure of RACFIS

RACFIS is closely related to the CNFS model, which has already been mentioned above. The network is a five-layer structure as shown in Fig. 2. The operation of each layer of the network with the input vector $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$ and output $y(t)$ at time t is presented as follows:

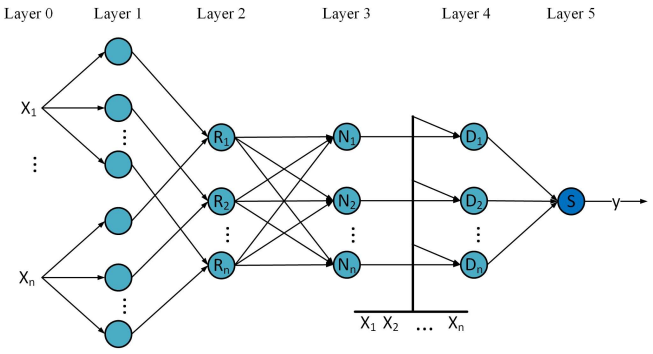

Fig. 2. The illustration of the main structure of RACFIS. X_i in the figure represents input variable, R_i denotes a complex fuzzy rule, N_i is the normalization of the product of each rule, D_i is the outcome of Sugeno defuzzification, and S denotes the sum of all results which is identical to the output of the network, i.e., y.

Layer 1: This is the fuzzification layer in which the real inputs are transferred into fuzzy membership grades with the simplified complex Gaussian membership function as follow:

$$O^i_{1,j}(t) = \exp\left(-\frac{(x(t)-\mu^i_j)^2}{2b^i_j}\right) - j\exp\left(-\frac{(x(t)-\mu^i_j)^2}{2b^i_j}\right)\frac{x(t)-\mu^i_j}{b^i_j}\delta^i_j, \quad (10)$$

where $O^i_{1,j}(t)$ denotes the membership of the ith rule of the jth input, and $\{\mu^i_j, b^i_j, \delta^i_j\}$ is the antecedent parameter set for each rule.

Layer 2: This layer is for calculating the firing strength of the inference system, in which a complex multiplication is applied as follows:

$$O^i_2(t) = \prod_{j=1}^{n} O^i_{1,j}(t) =: \alpha^i_j(t) + j\beta^i_j(t), \quad (11)$$

$$\alpha^i_j(t) = \gamma^i(t)\left[1 - \prod_{j=1}^{n}\frac{(x_j(t)-\mu^i_j)}{b^i_j}\delta^i_j\right], \quad (12)$$

$$\beta^i_j(t) = -\gamma^i(t)\sum_{j=1}^{n}\frac{x_j(t)-\mu^i_j}{b^i_j}\delta^i_j, \quad (13)$$

$$\gamma^i(t) = \exp\left(-\sum_{j=1}^{n}\frac{(x_j(t)-\mu^i_j)^2}{2b^i_j}\right), \quad (14)$$

where $O^i_2(t)$ is the strength of the ith firing and $\alpha^i_j(t), \beta^i_j(t)$ represent the value of real part and imaginary part of the fuzzy membership grade, respectively.

Layer 3: The normalization layer for relevant firing strengths. Here, complex division is used to perform normalization, so that the two dimensions of information can be fully interacted. For simplicity, formulas (12) and (13) are used here to simplify the expression:

$$O^i_3(t) = \frac{O^i_2(t)}{\sum_{r=1}^{K} O^r_2(t)} = \frac{\alpha^i(t) + j\beta^i(t)}{\sum_{r=1}^{K}\alpha^r(t) + j\sum_{r=1}^{K}\beta^r(t)}, \quad (15)$$

where $O^i_3(t)$ is the normalized value of ith node in this layer.

Layer 4: Each node in layer 4 is an adaptive node, in which the Sugeno method is applied to calculate the output of each fuzzy reasoning with the consequent parameters:

$$O^i_4(t) = O^i_3(t)\cdot(p^i_0 + \sum_{j=1}^{n} p^i_j x_j(t)), \quad (16)$$

where $O^i_4(t)$ is the output of each node and $\{p^i_0, p^i_1, p^i_2, \ldots, p^i_n\}$ is the set of consequent parameters.

Layer 5: Only one single node is in this layer, the function is to obtain the overall output of the network. In this case, only the real part of the complex output is utilized.

$$O_5(t) = \mathrm{Re}\sum_{i=1}^{K} O^i_4(t), \quad (17)$$

where "Re" means the real part of the complex number, $O_5(t)$ is the overall output of the neuro-fuzzy system and K is the number of fuzzy rules for each input. To perform a real value regression task, it would be incorrect if the output result is a complex number. This operation of converting a complex membership degree defined in the complex plane into a real value is essentially a step of CFS defuzzification. So far, some attempts have been made to achieve this purpose including the dot product (ANCFIS), the projection on the real axis (CNFS), or the modulo operation. For a complex fuzzy membership function defined in a polynomial form, its projection on the real axis equals the real component, which makes it the best strategy to directly use the real part of the final inference result as the actual defuzzified output. Additionally, directly employing the real part as network result does not change the mapping relation between the fuzzy antecedents and the output, making it possible to use the unsupervised convex clustering method to pre-train the antecedent parameters.

Similarly to many other neural networks based on Gaussian functions, the mean square error (MSE) function is used as the cost function for network optimization. There are three reasons for this choice. First, the use of the MSE function is convenient since we can easily find its derivative. Second, assuming that the error between the predicted output and the real output complies with a Gaussian distribution, the minimum MSE error is essentially consistent with their maximum likelihood estimation [72], which makes the MSE the ideal candidate for measuring the loss of a regression process. Last but not least, the MSE cost function generally leads to a faster convergence. Assume that there are N input-output pairs $\{x_1(t), x_2(t), \ldots, x_n(t), y(t)\}_{t=1}^{N}$, the corresponding MSE function is as follows:

$$\mathrm{MSE} = \frac{1}{2N}\sum_{t=1}^{N}(O_5(t) - y(t))^2, \quad (18)$$

B. Bisecting K-Means

The primeval k-means algorithm was first proposed respectively by Lloyd [73] and Kanungo *et al.* [74]. To divide a dataset into k clusters, the basic k-means first initializes k centroids $\xi_1, \xi_2, \ldots, \xi_k$, then computes the distance between

each point x_i and the centroid ξ_j. Every point is further assigned to the nearest cluster according to the point-to-center distance. Therefore, the updated centroids are obtained as follows:

$$\xi_j = \frac{1}{|c_j|} \sum_{x \in c_j} x \tag{19}$$

where c_j denotes the jth cluster and $j = 1, 2, \ldots, k$. The optimized clusters can be achieved by repeating the above process when stopping condition is satisfied. Traditional k-means algorithm has a manifest flaw that manual determination of the cluster centers is required, but manually selected centroid is not precise enough, which largely affects the accuracy of the algorithm. For tackling this, the k-means++ algorithm [75] has been developed with the ability to automatically select the centers. This method randomly selects cluster centers, then refines the centroids through multiple iterations to obtain the optimal clustering scheme. However, this solution causes randomness as well as local optima.

Steinbach *et al.* [28] introduced the idea of hierarchical clustering into the modification of basic k-means, leading to a new top-down clustering algorithm which is named as bisecting k-means. Similarly to k-means++, this algorithm also stochastically chooses clustering centers, but it effectively avoids local optima with the help of a binary decision tree. During the execution of the algorithm, all data is first regarded as a cluster, and then the cluster is divided into two, using ordinary k-means. According to the maximum SSE (sum of the squared errors) principle, the cluster which has the largest SSE will be split again. The above operations are repeated multiple times until the data is classified into k different clusters. Although the result obtained still has a certain degree of randomness, it can assure the outcome to be closer to the true global optimum. Bisecting k-means is a fast and efficient method. The computational speed is faster than traditional k-means or k-means++ especially when k is large. The detailed steps of bisecting k-means are shown as follows:

Algorithm 1 Bisecting K-Means.

Step 1. Manually decide on the number of clusters k. (In this paper, k must be identical to the number of fuzzy rules.)
Step 2. Initialize the objective data set, generalize a cluster which contains all data points.
Step 3. Bisect the cluster using basic k-means $(k = 2)$ by randomly opting two centroids.
Step 4. Estimate the SSE of two clusters, pick up the one with the larger SSE as the next target to split.
Step 5. Go back to Step 2, carry out following steps repeatedly until the overall cluster number is equal to k.

C. Quasi-Hyperbolic Momentum (QHM) Optimization

Currently, excluding the stochastic gradient methods such as SGD [76] and Adam [77], the gradient-momentum method such as heavy-ball [78] and Nesterov [79] are also widely applied. Such methods usually have two hyperparameters to determine, one is the learning rate, and the other is related to the momentum. Hu *et al.* [80] compared the first-order optimization method with the classic control theory, confirming that the gradient optimization process is essentially a PID controller. A PID controller has three linear independent parameters, namely proportional (P), integral (I), and derivative (D) terms. The essence of the momentum is the influence of historical gradients on the current process, which is analogous to the function of a lead-lag compensator in a PID control problem.

Motivated by this, the multi-parameter gradient-momentum optimization methods are receiving attention since they may be better adapted to the optimization surface, especially when it is non-convex, thereby improving the performance of the algorithm. The Quasi-hyperbolic momentum [30] algorithm is one of them, where the learning rate α, the immediate discount factor v and the momentum discount factor β constitute the hyperparameter space of the algorithm. Note that if one considers β as a free variable, the mathematical expression of QHM can be written in the form of a PID. The idea of quasi-hyperbolic momentum is initially revealed by an informal and conjectural variance reduction analysis, for which a plain explication is the average of the momentum and regular gradient drop that weighted by v. The ordinary definition of the heavy-ball momentum firmly relates the exponential discount factor β to the contribution of the immediate gradient. But the role played by the QHM directly originates from decoupling the momentum from the impact of current gradient (with deterministic setting), as well as decoupling the gradient square mean from the current gradient square (with stochastic setting) when updating the weights. The update rule for the QHM optimization using deterministic setting is written as follows:

$$g(k+1) = \beta g(k) + (1-\beta)\nabla f(x(k)) \tag{20}$$

$$x(k+1) = x(k) - \alpha[(1-\nu)\nabla f(x(k)) + \nu g(k+1)], \tag{21}$$

where $\alpha, \beta, v \in R, \nabla f(x(k))$ represent the first-order derivative of the optimization object,$(1-\nu)\nabla f(x(k)) + \nu g(k+1)$ denotes the modulo initialization bias which is a gradient estimator in short, and $g(k)$ can be viewed as the momentum buffer. Under normal circumstances, the value of ν is 0.7 and the value of β is 0.999, which means that in most cases such algorithm can be applied like a single parameter method by only adjusting the learning rate.

D. Recursive Least Square (RLS) Estimation

Traditionally, the consequent part of a Sugeno fuzzy system is updated utilizing the offline least square (LS) method, which is less effective than the online method. Kalman filtering can also be used to update the parameters, but in the absence of any additional knowledge of the target, this only increases the computational complexity without leading to any significant benefits. Therefore, the RLS which satisfies the requirement for online learning as well as computational efficiency, is employed here as the updating method for the consequent part of the RACFIS network. Assume the function of the model is written as follows:

$$Y_L = \phi_L \hat{\theta} + \omega_L, \tag{22}$$

where ϕ_L is the system matrix, $\hat{\theta}$ is the vector for the input variables, Y_L is the output vector, and ω_L is the vector of system error or noise. To estimate $\hat{\theta}$ at the kth recursion, according to the principle of the least square estimation, the following equation set is obtained:

$$\begin{aligned}\hat{\theta}(k) &= (\phi_k^T\phi_k)^{-1}\phi_k^T Y_k \\ \phi_k &= [\phi_{k-1}^T, \varphi(k)]^T \\ Y_k &= [Y_{k-1}^T, y(k)]^T,\end{aligned} \tag{23}$$

where $\varphi(k)$ and $y(k)$ are the system vector and output vector for the kth input-output pair, respectively. Here, it is specified that the equation starts from $\varphi(0)$, i.e., $k = 0, 1, , n-1$. Now, define $P(k) = (\phi_k^T\phi_k)^{-1}$, we can easily obtain:

$$P(k) = [P^{-1}(k-1) + \varphi(k)\varphi^T(k)]^{-1}, \tag{24}$$

Substitute equations (22) and (23) into the representation of the matrix inversion $(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$, where in this case $A = P^{-1}(k-1), C = I, B = \varphi(k)$ and $D = \varphi^T(k)$. Then, the following equation can be achieved:

$$P(k) = \left[I - \frac{P(k-1)\varphi(k)\varphi^T(k)}{1+\varphi^T(k)P(k-1)\varphi(k)}\right]P(k-1), \tag{25}$$

Consider that $\hat{\theta}(k) = (\phi_k^T\phi_k)^{-1}\phi_k^T Y_k$ and $P^{-1}(k)\hat{\theta}(k) = \phi_k^T Y_k$, therefore the ultimate form of $\hat{\theta}$ at the kth recursion is:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)], \tag{26}$$

For the implementation of RLS estimation, equations (25) and (26) constitute the update rule. Regarding the initialization of the algorithm, $\varphi(0)$ is set to be zero vector while the start of P is given as $P(0) = \alpha I$, where α usually takes a large positive number, I is the identity matrix.

E. Bisecting K-Means, QHM and RLS joint optimization for RACFIS

In this paper, a joint optimization method utilizing bisecting k-means, QHM and RLS is proposed. A key factor for this method to work is to ensure that all data is normalized in the interval $[0, 1]$ before application to the model. After the normalization process, bisecting k-means clustering is used to pre-train the antecedent part of the network for the purpose of having a faster convergence as well as a better result which is closer to the true global optimum. $\{\mu^i, b^i, \delta^i\}, i = 1, 2, \ldots, N$ is the set of the antecedent parameters, where μ^i is the center and b^i is the width of each complex Gaussian membership function. Applying bisecting k-means to operate on the entire training set, a set of cluster centroids $C = \{c_1, c_2, \ldots, c_i, \ldots, c_N\}$ is generated and at the same time the sum of the distances from all points in each cluster to the centroid is achieved, which is denoted as $D = \{d_1, d_2, \ldots, d_i, \ldots, d_N\}$, where $d_i = \sum_{k=1}^{n}(x_i^k - c_i)$. We determine the value of $\{\mu^i, b^i, \delta^i\}$ as follows:

$$\mu^i = c_i, b^i = \rho\sum_{k=1}^{n}(x_i^k - c_i), \delta^i = 1, \tag{27}$$

where x_i^k is the coordinate of each individual data point in the ith cluster, ρ denotes the expansion coefficient, which is valued between 0.8 and 0.95, and n is the number of points in this cluster.

As a result of clustering, the network is initialized at a position closer to the global minimum than that of a random initialization, but a gap with the optimal solution is still there. To implement QHM, combine the equations (20) and (21), the following expression of the update rule is achieved as follows:

$$x(k+1) = x(k) - \alpha[(1-\nu\beta)\nabla f(x(k)) + \nu\beta g(k)], \tag{28}$$

where α is the initial learning rate, β is the exponential discount factor, v is the immediate discount factor, $g(k) = \beta g(k-1) + (1-\beta)\nabla f(x(k-1))$ is the momentum buffer of each parameter, and $e(k) = \text{Output}(k) - y(k)$ is the error of the kth iteration. As mentioned above, each rule has three different antecedent parameters$\{\mu_j^i, b_j^i, \delta_j^i\}$. By substituting them into formula (28), the iterative results of each parameter in each step can be achieved. The expression for the parameter set at iteration $k+1$ is as follows:

$$\begin{aligned}\mu^i(k+1) &= \mu^i(k) - \alpha[(1-v\beta)e(k)\left.\frac{\partial e(k)}{\partial \mu^i}\right|_{\mu^i=\mu^i(k)} + v\beta g_\mu^i(k)] \\ b^i(k+1) &= b^i(k) - \alpha[(1-v\beta)e(k)\left.\frac{\partial e(k)}{\partial b^i}\right|_{b^i=b^i(k)} + v\beta g_b^i(k)] \\ \delta^i(k+1) &= \delta^i(k) - \alpha[(1-v\beta)e(k)\left.\frac{\partial e(k)}{\partial \delta^i}\right|_{\delta^i=\delta^i(k)} + v\beta g_\delta^i(k)].\end{aligned} \tag{29}$$

After updating the non-linear antecedent parameters by QHM, the linear antecedent part is updated using RLS. For $i = 1, 2, \ldots, q$, to update the consequent part at kth iteration, the following two variables are derived:

$$\begin{aligned}\hat{\theta} &= [(\theta^1)^T, (\theta^2)^T, \ldots, (\theta^q)^T], \\ \varphi(k) &= [(\varphi^1(k))^T, (\varphi^2(k))^T, \ldots, (\varphi^q(k))^T],\end{aligned} \tag{30}$$

where $\theta^i = [a_0^i, a_1^i, \ldots, a_q^i]^T$ and

$$\varphi^i(k) = [\lambda_R^i, \chi_1(k)\lambda_R^i, \ldots, \chi_{q-1}(k)\lambda_R^i]^T,$$

λ_R^i is the real part output of the corresponding normalization node. By substituting $\hat{\theta}, \varphi(k)$ into equations (25) and (26), the updated parameters can be determined, which can be directly applied to the next step. To optimize the objective function (18) of the network, the clustering process only needs to be implemented once, whereas the QHM-RLS refining process is supposed to be repeated iterations until the model satisfies the error requirement or reaches the maximum epoch setting. The whole parameter adapting operation is shown as Algorithm 2.

Algorithm 2 Bisecting K-Means, QHM and RLS joint optimization (proposed).

Step 1. Split the data set into training set and testing set.
Step 2. Normalize the data set to the interval [0, 1]. Note that training set and testing set should be normalized separately.
Step. 3 Initialize the antecedent part of the parameter set $\{\mu^i, b^i, \delta^i\}$ using the information obtained by clustering.
Step. 4 Calculate the output of the architecture with the original settings, derive the error by $e(k) = \text{Output}(k) - y(k)$.
Step. 5 Refine the antecedent part (non-linear) of the network by QHM optimization method.
Step. 6 Estimate the consequent part (linear) with RLS estimation.
Step. 7 Calculate the output of the architecture, derive the error by$e(k) = \text{Output}(k) - y(k)$.
Step. 8 Repeat Step. 5 Step. 7 until the stopping criterion is satisfied.

Algorithm 3 MIV-RBFN Algorithm.

Step 1. Initialize a RBF network, train the network with original input set $\chi^T = \{X_1, X_2, \ldots, X_q\}, q \in N^+$, where $X_k^T = \{x_1^k, x_2^k, \ldots, x_i^k, \ldots, x_n^k\}, k = 1, 2, \ldots, q, i = 1, 2, \ldots, n; n$ is the dimension of an input vector.
Step 2. Gradually increase the number of RBF kernels until the network accuracy reaches the setting value, denote the optimized network structure as function $G_{op}(X, a)$, where x is the input set and a is the parameter vector.
Step 3. Generate two new input sets, where $\chi^T_{\text{increase}} = \chi^T \cdot (1+\eta), \chi^T_{\text{decrease}} = \chi^T \cdot (1-\eta)$ and η is the adjustment rate.
Step 4. Run the network $G_{op}(X, a)$ with inputs χ^T_{increase} and χ^T_{decrease} respectively, obtain the output vectors $G_{op}(\chi^T_{\text{increase}}, a)$ and $G_{op}(\chi^T_{\text{decrease}}, a)$.
Step 5. Calculate the MIV of each variable: $\text{MIV}(x_i) = abs[G_{op}(\chi^T_{\text{increase}}, a) - G_{op}(\chi^T_{\text{decrease}}, a))]/q$.

F. MIV-RBFN Algorithm for Variable Analysis

Normally, principal component analysis (PCA) [81]–[83] is applied to select variables. However, as a linear method it does not perform well on highly non-linear data sets. Kernel PCA [84] is an extension of the conventional PCA, which can linearize non-linear features by mapping them into the Hilbert space via the kernel functions. Although this method is more effective with non-linear data, difficulties still exist to choose a suitable parameter configuration in the absence of prior knowledge of the data set, due to the need to manually adjust the settings.

For RACFIS, an MIV [30] algorithm integrated with the RBF network is employed to evaluate the variables for RACFIS when it comes to the real-world data. The RBF network has an intuitive structure, easy training, and fast convergence. It is capable of approximating arbitrary non-linear functions and shares inherent similarity with neuro-fuzzy systems, which makes it a suitable candidate as a referee network for the variable selection algorithm of the RACFIS network. It is worth noting that the MIV algorithm here should not a dimensionality reduction algorithm. It generally relies on a pretrained neural network as a supervisor. It judges the importance of variables by comparing the difference between two special network output vectors, where one is generated by an input vector that is a certain percentage larger than the original counterpart and another one is obtained by an input vector which is the same percentage smaller. When implementing the algorithm, only the referee network needs to be trained without the need of prior knowledge of the data. The mean impact value is considered as one of the most effective indicators for evaluating variable importance. The entire implementation process is given by Algorithm 3.

IV. Experimental Result

This section includes four experiments to validate the proposed RACFIS architecture, and all simulations were based on MATLAB 2022b. Some benchmark models run for the tests were from MATLAB's built-in toolbox, while the rest were programmed from scratch. Note that some results directly came from the literature.

A. Performance Index

To more comprehensively evaluate the performance of the proposed architecture, MSE, STD [85], RMSE [86], MAE [87] and SMAPE [88] are utilized to measure the performance of the algorithm. The associated mathematical expressions are as follows:

$$STD = (\frac{1}{N-1}\sum_{i=1}^{N}(y_i - u)^2)^{1/2} \tag{31}$$

$$RMSE = (\frac{1}{N}\sum_{i=1}^{N}(y_i - w_i)^2)^{1/2} \tag{32}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - w_i)^2 \tag{33}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - w_i| \tag{34}$$

$$SMAPE = \frac{100\%}{N}\sum_{i=1}^{N}\frac{2 * |y_i - w_i|}{|y_i| + |w_i|}, \tag{35}$$

where N is the number of samples,u is the overall average of the samples, w_i represents the actual value of each sample, and y_i denotes the observed value.

B. Synthetic Data Test

The synthetic data is produced by the "tooth" function [61],

$$f(x) = 0.08 \times \{1.2 \times [(x-1) \times (\cos(3x))] + [(x - (x-1) \times (\cos(3x))) \times \sin(x))]\}, 3 \le x \le 7, \tag{36}$$

in which 200 training data points sampled from [3,7] with intervals of 0.02 while 50 test points sampled with intervals 0.08 within the same scale. More details of the test are

given in TABLE I and the visualized outcome is shown in Fig. 3. Apparently, both models have achieved good results. However, RACFIS is superior to CNFS in nearly all aspects under the premise of the same number of rules. RACFIS performs better in areas with steep changes, and the testing error and the number of epochs required are both a magnitude smaller than CNFS. In addition, given that the optimization methods of RACFIS and CNFS both have a certain degree of randomness, apart from selecting the most ideal experimental data, we also conducted 20 random tests and recorded the MSE values. The Kruskal-Wallis [89] test was employed to compare the obtained MSEs of those two models from the statistical aspect. In this test, we assume that two objects have no statistical difference, thus a p-value less than 0.05 suggests a statistically significant difference between the two models. In fact, a p-value of 0.0138 was obtained, which is enough to reject the null hypothesis, i.e., it can be concluded that the performance of RACFIS is better than that of CNFS statistically. In addition, TABLE I also gives the training complexity of RACFIS and CNFS on the data dimension m and the number of samples n. It can be seen that RACFIS is only sensitive to the input dimensionality, while CNFS is sensitive to both dimensionality and data volumes, meaning that RACFIS is much more competitive when processing a large amount of data.

TABLE I
COMPARISON OF CNFS AND RACFIS OVER SYNTHETIC TEST.

	MSE	Rules	Settings	Iterations	Complexity
CNFS	1.8691×10^{-4}	9	650 (Population)	200	$O(mk^m+n!)$
RACFIS	1.3445×10^{-5}	9	10^{-6} (Step size)	10	$O(mk^m)$

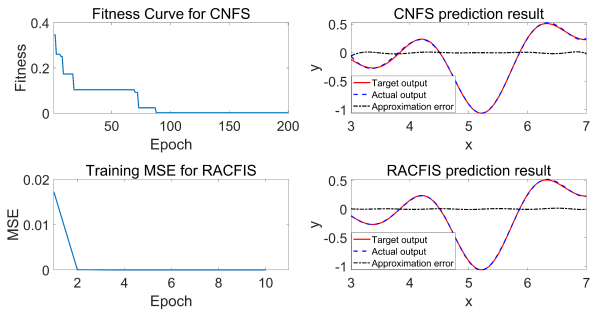

Fig. 3. Comparison between CNFS and RACFIS on "Tooth" function test for $\in [3, 7]$.

C. Sunspot Time Series Test

Sunspot data deployed in this test includes sunspot activity recorded between 1976 to 1992, which can be downloaded from World Data Center for the Sunspot Index (SIDC) [90]. The dataset comprises a total of 2000 single value samples. Naturally, to implement this experiment, each input-output data pair is constructed as $\{y(\tau-1), y(\tau); y(\tau+1)\}, \tau = 2, 3, \ldots, 2000$, in which the next sunspot state is predicted by the value of the current moment as well as the previous moment. Generated data points are firstly normalized between 0 and 1, then equally divided to get training set and test set. The results of the experiment on RACFIS and benchmark models are shown in TABLE II. The visualized prediction outcome is given in Fig. 4.

TABLE II
COMPARISON BETWEEN ALGORITHMS OVER SUNSPOT TEST.

	Testing MSE	Rules	Max epoch
SARIMA [91]	6.5733×10^{-3}	-	-
Elman-NARX [92]	1.4078×10^{-4}	-	200
NFS [66]	8.5112×10^{-5}	5	300
CNFS	4.1490×10^{-5}	5	300
RACFIS	4.1045×10^{-5}	3	30

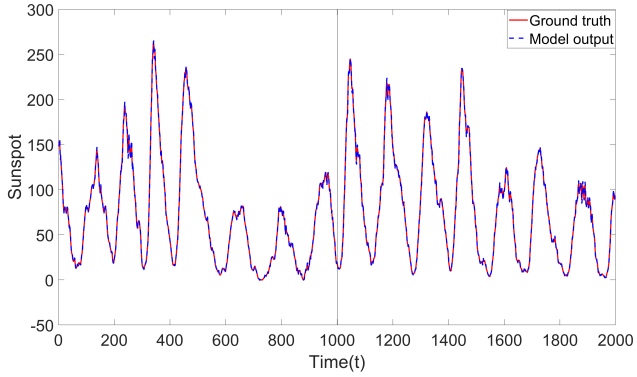

Fig. 4. RACFIS performance on Sunspot test. The left side of the black dividing line in the figure is the prediction effect on the training set, and the right side is the performance on the test set.

According to TABLE II, the accuracy of RACFIS is far beyond SARIMA [91] (an ARIMA variant for periodic time series forecast). Elman-NARX [92] (a branch of recurrent neural network) and NFS [66] (ANFIS with PSO-RLSE optimization) are also largely inferior to RACFIS regarding sunspot forecasting, which is in line with the advantage that complex fuzzy models show when dealing with periodicity. Compared with CNFS, which is also based on complex fuzzy logic, the performance between the two is basically the same, but RACFIS only uses 3 rules and 30 iterations of training to achieve the performance that CNFS needs 300 iterations and 5 rules to achieve. The superiority of the proposed hybrid optimization algorithm for RACFIS is self-evident.

D. Charpy Impact Data Test

Metals are crystalline substances, in which physical and chemical properties depend on the structure and type of atoms that constitute their crystal lattice [93]. Since all phenomena happen on the level of atomic size, even subtle changes in alloy composition may have a great impact on its attributes, which also indicates that the influence of composition on characteristics is highly non-linear. The Charpy impact property measures the capacity of assimilating the fracture energy of a material. Here, a data set with 16 input dimensions, single output and 830 data points is applied to implement the experiment. Such data set is tricky for traditional modeling methods due to its discontinuity, scatter in measurements and non-linearity, which

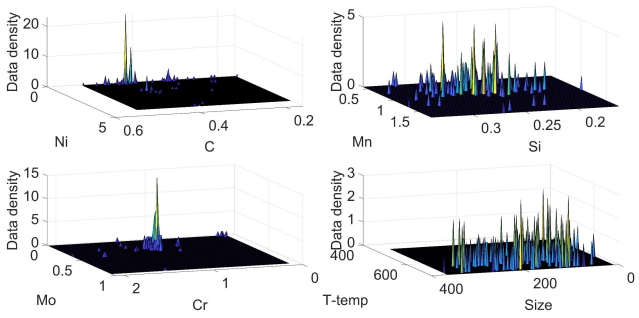

Fig. 5. 3D density plots between variable pairs "C" and "Ni", "Mn" and "Si", "Mo" and "Cr", and "T-temp" and "Size".

is partially visualized in Fig. 5. In addition, the MIV algorithm is used to sort it first on the purpose of ranking the contribution of each variable. The results are given in TABLE III.

TABLE III
MIV OF EACH VARIABLE IN CHARPY IMPACT TEST.

Input Variable	**Adjustment Rate**		
	10%	**20%**	**30%**
C	0.9866	1.9678	2.9380
Si	-1.2596	-2.5154	-3.7635
Mn	1.6863	3.3569	4.9896
S	0.0565	0.1129	0.1694
Cr	3.5733	6.9862	10.0912
Mo	1.4240	2.8452	4.2610
Ni	0.6321	1.2629	1.8956
Al	-0.0041	-0.0082	-0.0123
V	-0.2822	-0.5644	-0.8465
Hardening Temp	0	0	0
Tempering Temp	148.9641	-45.7167	1.5452×10^{-6}
Impact Temp	12.6622	29.7544	-46.9750
Sample Size	-2.0193	-1.8219	0.0686
Test Depth	0.6174	0.0207	0.0207
Category symbol			
Coded Site	-0.7666	-1.2536	-1.3499
Cooling Medium	-0.2611	-0.4680	-0.5806

According to the absolute value of MIV, the following 9 variables are shown to be more significantly influencing factors, which are C, Si, Mn, Cr, Mo, Ni, Sample Size, Tempering Temperature, and Impact Temperature. To this point, the analysis of the variables in the data set is completed. In this experiment, the first 600 sample-points are used as the training set, while the rest 230 are taken as the test set. The network setting for this experiment is given in TABLE IV.

As it can be seen from Fig. 6, after pre-training of the antecedent parameters by the bisecting k-means clustering, plus the proposed QHM-RLS hybrid algorithm adopted for refining, the network converged to a very small MSE value at the second epoch, which is rather close to the final optimal solution after 10 epochs. The comparison between the network prediction and the target output value is given in Fig. 7 and the result regression plot under 90% confidence is displayed in Fig. 8. Moreover, in TABLE V the overall performance of the proposed RACFIS regarding performance indices as well as the number of epochs required in the experiment is compared with other nine algorithms, either. RACFIS outperforms algorithms listed in the table for comparison from the perspective

TABLE IV
THE PARAMETER SETTING OF RACFIS FOR CHARPY TEST.

Bisecting k-means clustering for antecedence initialization			
Number of clusters for each variable k	6	Expansion coefficient ρ	0.95
QHM		RLS estimator	
Learning rate α	10^{-5}	Initial consequent parameter a_i, for $i = 1, 2, \ldots, q$	0.3
Immediate discount factor v	0.7	P_0	$\alpha * I$
Momentum discount factor β	0.999	α	10^6
Max epoch	10	I	60*60 Identity matrix
Initial momentum buffer $g(0)$	0	θ_0	60-dimension zero vector

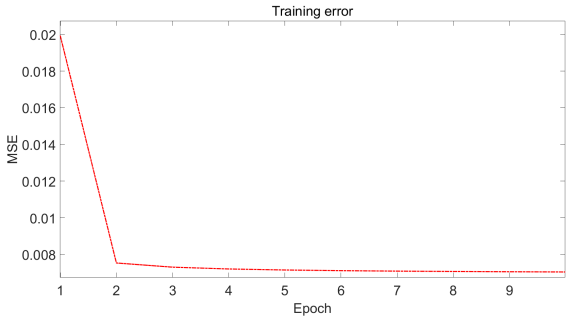

Fig. 6. The value of the MSE cost function over the learning process for Charpy data test.

of RMSE, MAE and SMAPE, and achieves this performance with only 10 iterations. STD indicator reflects the degree of dispersion of the output data itself. Obviously, the outputs from RACFIS have a lower prediction error with a higher degree of dispersion, which indicates a remarkable generalization capability. In terms of the comparisons with other fuzzy systems, the performance of RACFIS also surpasses GrC-NF [100] (granular modeling); Q-ANFIS [101] (fuzzy C-Means clustering and quantum membership functions); interval type-2 model [102] (Sugeno defuzzification); and CNFS which has the similar complex fuzzy logic. Except for the same number of rules as Q-ANFIS, the number of rules required by RACFIS is less than that of all non-complex fuzzy inference systems.

TABLE V
COMPARISON OF THE PERFORMANCE (CHARPY DATA TEST).

	STD	MAE	SMAPE	RMSE	**Rules**	**Epoch**
BP [94]	25.2233	16.9010	21.2874	20.8389	–	200
RBF [95]	30.9238	14.7211	19.5694	20.1443	–	100
GRNN [96]	22.7647	16.0598	19.0444	22.0339	–	100
LSTM [97]	24.9117	17.0228	21.5417	21.2242	–	200
DBN [98]	27.2085	16.2536	20.2341	20.1211	–	50
SVR [99]	26.3198	13.9911	17.8943	19.2947	–	30
GrC-NF [100]	–	–	–	20.4200	9	200
Q-ANFIS [101]	–	–	–	18.1700	6	100
IT2Sugeno [102]	–	–	–	19.6500	8	100
CNFS	25.7738	16.3243	22.4266	20.5506	6	100
RACFIS	27.2434	13.5313	17.8632	17.1872	6	10

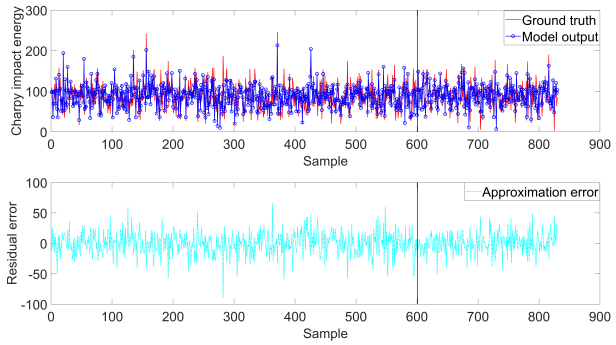

Fig. 7. Performance of RACFIS in Charpy energy prediction. The part on the left of the black line is the comparison between the predicted results of the training set and the actual value. The right side is the same but for the performance on the test set. Approximation error is also given in the figure.

Fig. 8. Result regression plot for Charpy energy prediction.

E. Ultimate Tensile Strength (UTS) Data Test

The ultimate tensile strength is defined as the peak of engineering stress in a stress-strain curve [103]. Within the data set, 15 input variables that correspond to an output dimension, with 3760 data points. This high dimensional data is tricky for its non-linearity, large interaction between input variables, measurement error of industrial process, and the data sparsity. Such characteristics of the data can be partly reflected in Fig. 9. Apart from this, extra 12 data points with similar input values but different outputs are taken as the validation set in order to measure the generalization capacity of the model. Regarding these 12 *abnormal* points, there are three reasons for them to exist, including human errors in data entry, missing dimensions, and measurement errors, which are also inevitable in other real-world data sets. This validation process can further test the performance from a higher level of universal applicability. The result of MIV analysis on the UTS data is shown in TABLE VI.

Here, according to the MIV value, 10 variables should be selected as the inputs to the network, which are C, Mn, Cr, Mo, Ni, Site, Tempering Temperature, Cooling Medium, Sample Size and Test Depth, respectively. 1000 randomly selected data points from the original data set are given to form the training set, while another 500 are taken as the test set. The network parameter settings of the experiment are shown in TABLE VII.

The training curve is presented in Fig. 10, the prediction result of the network is shown in Fig. 11 and the result regression with 90% confidence bonds is given in Fig. 12. Analogous to

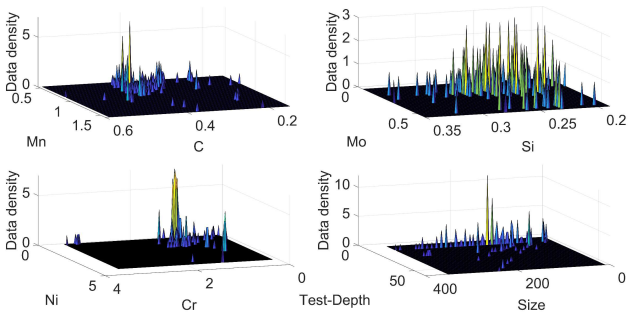

Fig. 9. 3D density plots of variable pairs "C" and "Mn", "Mo" and "Si", "Ni" and "Cr", and "Test-Depth" and "Size".

TABLE VI
MIV OF EACH VARIABLE IN ULTIMATE TENSILE STRENGTH TEST.

Input Variable	Adjustment Rate		
	10%	20%	30%
C	8.1864	16.3605	24.5097
Si	0.0794	0.1594	0.2404
Mn	-10.3150	-20.4747	-30.3276
S	-0.0723	-0.1445	-0.2168
Cr	-15.0656	-29.7948	-43.8615
Mo	-3.0059	-6.0067	-8.9937
Ni	-37.5670	-74.1685	-108.8752
Al	-0.3721	-0.7443	-1.1164
V	-0.1084	-0.2169	-0.3253
Hardening Temp	-0.0017	0	0
Tempering Temp	224.3830	274.6474	-482.3313
Sample Size	-830.1928	-624.7182	1321.7
Test Depth	-25.7766	-35.7784	-64.4520
Category symbol			
Site	-242.2889	-454.9862	-609.9973
Cooling Medium	-179.3221	-344.4547	-482.3313

the case in the Charpy data test, the network error rapidly decreased to a relatively low level with the first 2 iterations and after that the error of the network still improved noticeably over the increase of the number of iterations, until this trend to be almost flat around the 20th epoch. In TABLE VIII, the performance of RACFIS is compared with other algorithms. It can be seen from the table that referring to the prediction error indicators, RACFIS has a very good performance, surpassing all the algorithms mentioned in the comparison. In comparison with other fuzzy algorithms, RACFIS only needs 5 rules to obtain a comparable accuracy that is achieved by IMOFM

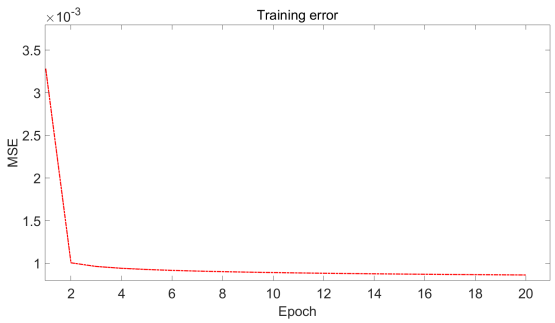

Fig. 10. The value of the MSE cost function over the learning process for UTS data test.

TABLE VII
THE PARAMETER SETTING OF RACFIS FOR UTS TEST.

Bisecting k-means clustering for antecedence initialization			
Number of clusters for each variable k	5	Expansion coefficient ρ	0.95
QHM		RLS estimator	
Learning rate α	10^{-6}	Initial consequent parameter a_i, for $i = 1, 2, \ldots, q$	0.3
Immediate discount factor v	0.7	P_0	$\alpha * I$
Momentum discount factor β	0.995	α	10^6
Max epoch	10	I	55*55 Identity matrix
Initial momentum buffer $g(0)$	0	θ_0	55-dimension zero vector

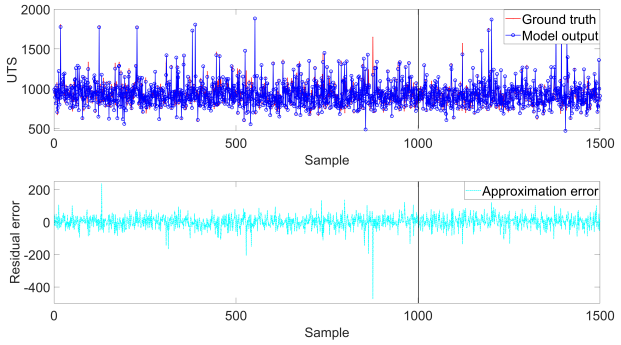

Fig. 11. Performance of RACFIS in UTS prediction. The left side of the black line is the comparison between the network outputs of the 1000 training samples and the target values. The right side is with the same narrative for 500 testing data points. Approximation error is shown in the figure as well.

[104] (Mamdani type-1), MOIT2FM [105] (Mamdani type-2) and IT2-Sugeno with 6 rules, not to mention that it only requires 20 epochs to converge. Even though CNFS includes a similar complex fuzzy logic as RACFIS, it needs far more iterations than RACFIS owing to the limitations of the PSO optimization as well as the random initialization of parameters. The adoption of the clustering algorithm in RACFIS solves this problem well. In summary, RACFIS has proven to be able to achieve an excellent performance with the smallest number of iterations compared with other algorithms listed in the case of the UTS test.

As already stated, the UTS data set also has a validation set consisting of 12 *abnormal* data points to test the generalization capacity of the network. Fig. 13 displays the linear regression between the target output and the prediction with 90% confidence band. The experimental results of all benchmark algorithms and RACFIS are given in TABLE IX. In this case, STD has become a very important indicator. We can clearly see from the same table that GRNN, LSTM and DBN have almost no processing capabilities for this validation set. The STD value being close to 0 indicates that these three algorithms hardly respond to subtle changes of information in the validation set. In contrast, the other four algorithms can process these data more effectively, where RACFIS has the smallest prediction error, which is surprisingly good for this tricky test.

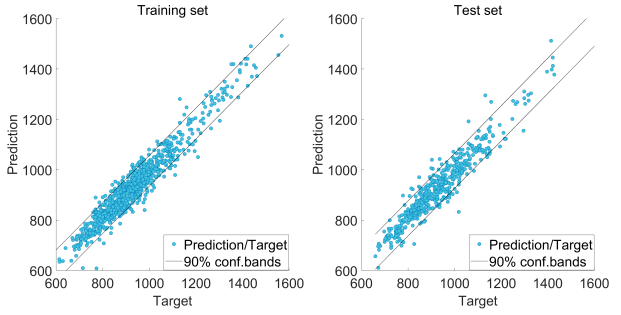

Fig. 12. Result regression plot for UTS prediction.

TABLE VIII
COMPARISON OF THE PERFORMANCE (UTS DATA TEST).

	STD	MAE	SMAPE	RMSE	**Rules**	**Epoch**
BP	150.9674	32.1335	3.4460	44.4965	–	200
RBF	149.2079	41.2173	4.4341	54.1319	–	100
GRNN	153.8146	39.2092	4.1841	56.5168	–	100
LSTM	137.5868	42.4865	4.5160	56.5765	–	200
DBN	142.1892	36.1535	3.9260	47.7999	–	50
SVR	149.0028	30.0553	3.1829	40.1637	–	30
IMOFM [104]	–	–	–	45.5200	6	200
MOIT2FM [105]	–	–	–	40.5200	6	100
IT2Sugeno	–	–	–	38.7600	6	100
CNFS	142.8011	37.6474	4.0346	51.3500	5	100
RACFIS	151.4953	30.4602	3.3001	39.4170	5	20

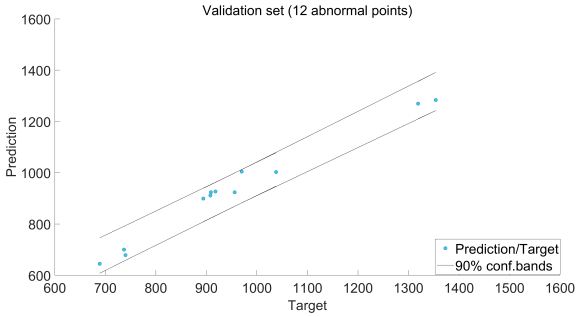

Fig. 13. Results regression plot for UTS prediction (12 abnormal data points).

TABLE IX
COMPARISON OF THE PERFORMANCE (12 ABNORMAL DATA POINTS TEST).

	STD	MAE	SMAPE	RMSE
BP	227.1069	40.5753	4.4768	53.0111
RBF	165.6330	44.7274	4.4971	55.8941
GRNN	5.9167	250.5156	28.0084	316.9053
LSTM	9.1936	276.9733	26.4932	400.6833
DBN	0.8827	244.8736	27.2302	312.2535
CNFS	184.4885	46.9601	4.8166	59.3112
RACFIS	202.1760	33.0322	3.6206	39.0780

However, the challenge does not stop there. The fifth point and the eighth point in this validation set share almost the same output, whereas the most input values of them are rather different. This suggests some unknown factors have actively

influenced the data, which makes it extremely difficult to predict. Specific information of such two points is shown in TABLE X. TABLE XI presents the performance of different models on these two abnormal data points, in which the prediction result by RACFIS is the closest to the actual value. RACFIS exhibits excellent generalization capabilities, which benefited from the two-dimensional properties of complex fuzzy logic. The hybrid optimization method proposed in this paper has also played a key role, enabling its performance to exceed the CNFS architecture which also relies on the two-dimensional complex fuzzy logic.

TABLE X
TWO ABNORMAL POINTS WITH THE IDENTICAL UTS (OUTPUT) BY DIFFERENT INPUT.

	Test Depth	Size	Site	C	Mn
Point 5	12.7	360	3	0.4100	0.8468
Point 8	12.7	250	3	0.4065	0.5704
	Cr	Mo	Ni	T-temp	UTS
	0.9806	0.1852	0.2500	520	908
	1.2149	0.2616	1.3606	615	909

TABLE XI
PREDICTION OF TWO ABNORMAL POINTS BY DIFFERENT MODELS.

BP	RBF	GRNN	LSTM	DBN	CNFS	RACFIS	Target
1047.21	1005.05	716.24	516.20	714.18	973.86	911.06	908
920.87	927.41	697.93	516.20	711.14	928.69	923.68	909

V. CONCLUSION AND FUTURE WORK

In this paper, the network architecture and the optimization policy are presented for RACFIS. This model inherits the complex Gaussian fuzzy membership function from CNFS, with the necessary new modifications to facilitate the implementation of the first-order derivative optimization. Regarding the optimization method, the algorithm applies the bisecting k-means clustering as an unsupervised learning method to pre-train the data for the purpose of approximately determining the antecedent parameters to avoid problems caused by random initialization. A hybrid QHM−RLS algorithm is then used to further refine the network parameters. Experimental results show that this combined optimization method enables RACFIS to quickly converge with a very small number of iterations, while achieving the necessary competitive performance. The above modifications also allow for a superior generalization capability with a smaller rule-base as compared to non-complex fuzzy models. In addition, the MIV algorithm, which is designed using RBF network, can also lead to a good performance when dealing with tricky real-world data in ranking the importance of variables, which outperforms the traditional MIV algorithm based on BP network. In summary, RACFIS is a powerful method when applied to real-world regression modelling.

Future work will focus on the following two aspects to facilitate the application of complex fuzzy architectures. First, we will explore the semantic interpretability of complex fuzzy sets and develop new complex membership functions if necessary, therefore, building complex fuzzy systems that can be completely understood by humans. Secondly, the current complex fuzzy systems are relatively at their infancy, and there are already theories that prove that fuzzy systems can also benefit from the increase in network depth to obtain performance improvements. We will as well investigate the possibility of increasing network depth in situations deemed to be relevant and appropriate.

REFERENCES

[1] J.-S.R. Jang and J. Roger, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1993.

[2] Chia-Feng Juang and Chin-Teng Lin, "An online self-constructing neural fuzzy inference network and its applications," in *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 12-32, Feb. 1998, doi: 10.1109/91.660805.

[3] N. Talpur, S. J. Abdulkadir, H. Alhussian, ·Mohd H. Hasan, N. Aziz, and A. Bamhdi, "A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods," *Neural computing & applications*, vol. 34, no. 3, pp. 1837–1875, 2022, doi: 10.1007/s00521-021-06807-9.

[4] W. R. Jeng, C. Yeh and S. Lee, "General type-2 fuzzy neural network with hybrid learning for function approximation," *2009 IEEE International Conference on Fuzzy Systems*, 2009, pp. 1534-1539, doi: 10.1109/FUZZY.2009.5277250.

[5] Chi-Hsu Wang, Chun-Sheng Cheng and Tsu-Tian Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1462-1477, June 2004, doi: 10.1109/TSMCB.2004.825927.

[6] E. . Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *International journal of human-computer studies*, vol. 51, no. 2, pp. 135–147, 1999, doi: 10.1006/ijhc.1973.0303.

[7] M. Sugeno, "An introductory survey of fuzzy control," *Information sciences*, vol. 36, no. 1, pp. 59-83, 1985.

[8] M. Sugeno and G.T. Kang, "Structure identification of fuzzy model," *Fuzzy sets and systems*, vol. 28, no. 1, pp. 15-33, 1988.

[9] G. Acampora, R. Schiattarella, and A. Vitiello, "On the Implementation of Fuzzy Inference Engines on Quantum Computers," *IEEE transactions on fuzzy systems*, vol. 31, no. 5, pp. 1–15, 2023, doi: 10.1109/TFUZZ.2022.3202348.

[10] G. G. Rigatos and S. G. Tzafestas, "Parallelization of a fuzzy control algorithm using quantum computation," in *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 451-460, Aug. 2002, doi: 10.1109/TFUZZ.2002.800690.

[11] J. M. Mendel, "Uncertain rule-based fuzzy systems: introduction and new directions, 2nd ed," *Springer*, 2017.

[12] N. N. Karnik, J. M. Mendel and Qilian Liang, "Type-2 fuzzy logic systems," in *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643-658, Dec. 1999, doi: 10.1109/91.811231.

[13] C. Chen, R. John, J. Twycross and J. M. Garibaldi, "Type-1 and interval type-2 ANFIS: A comparison," *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2017.8015555.

[14] K. . Shihabudheen and G. . Pillai, "Recent advances in neuro-fuzzy system: A survey," *Knowledge-based systems*, vol. 152, pp. 136–162, 2018, doi: 10.1016/j.knosys.2018.04.014.

[15] L. Ljung, *System identification: theory for the user, 2nd ed*. Upper Saddle River, N.J.: London: Prentice Hall PTR; Prentice-Hall International, 1999.

[16] D. Simon, "Training fuzzy systems with the extended Kalman filter," *Fuzzy sets and systems*, vol. 132, no. 2, pp. 189-199, 2002, doi: 10.1016/S0165-0114(01)00241-X.

[17] Z. -L. Sun, K. -F. Au and T. -M. Choi, "A Neuro-Fuzzy Inference System Through Integration of Fuzzy Logic and Extreme Learning Machines," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1321-1331, Oct. 2007, doi: 10.1109/TSMCB.2007.901375.

[18] Yixin Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," in *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 716-728, Dec. 2003, doi: 10.1109/TFUZZ.2003.819843.

[19] M. Shoorehdeli, M. Teshnehlab, and A. . Sedigh, "A novel training algorithm in ANFIS structure," in *2006 American Control Conference*, 2006, vol. 2006, p. 6 pp.-5064, doi: 10.1109/ACC.2006.1657525.

[20] E. G. Carrano, R. H. C. Takahashi, W. M. Caminhas and O. M. Neto, "A genetic algorithm for multiobjective training of ANFIS fuzzy networks," *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 3259-3265, doi: 10.1109/CEC.2008.4631239.

[21] D. Karaboga and E. Kaya, "Training ANFIS using artificial bee colony algorithm," *2013 IEEE INISTA*, 2013, pp. 1-5, doi: 10.1109/INISTA.2013.6577625.

[22] J. Chen, A. Kyrillidis, "Decaying Momentum Helps Neural Network Training," arXiv preprint arXiv:1910.04952, 2019.

[23] D. Ramot, R Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE transactions on fuzzy systems*, vol. 10, no. 2, pp. 171-186, 2002.

[24] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *IEEE transactions on fuzzy systems*, vol. 11, no. 4, pp. 450-461, 2003.

[25] S. Dick, "Toward complex fuzzy logic," *IEEE transactions on fuzzy systems*, vol. 13, no. 3, pp. 405-414, 2005.

[26] O. Yazdanbakhsh and S. Dick, "A systematic review of complex fuzzy sets and logic," *Fuzzy sets and systems*, vol. 338, pp. 1-22, 2018.

[27] R.E. Bellman,*Adaptive control processes*. *Princeton University Press*, 1961.

[28] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, Department of Computer Science and Engineering University of Minnesota. Citeseer, 2000, pp. 525-526.

[29] J. Ma and D. Yarats, *Quasi-hyperbolic momentum and Adam for deep learning*. Published as a conference paper at *ICLR 2019*, 2019.

[30] G. Dombi, P. Nandi, J. Saxe, A. Ledgerwood and C. Lucas, "Prediction of rib fracture injury outcome by an artificial neural network," *The Journal of trauma*, vol. 395, pp. 915-921, 1995.

[31] A. Kaufman and M. M. Gupta, *Introduction to Fuzzy Arithmetic*.New York, NY, USA: Van Nostrand Reinhold, 1985.

[32] D. Moses, O. Degani, H.-N. Teodorescu, M. Friedman, and A. Kandel, "Linguistic coordinate transformations for complex fuzzy sets," in *International Fuzzy Systems. Conference Proceedings (Cat. No.99CH36315)*, 1999, vol. 3, pp. 1340-1345 vol.3, doi: 10.1109/FUZZY.1999.790097.

[33] H. Nguyen, A. Kandel, and V. Kreinovich, "Complex fuzzy sets: towards new foundations," in *Ninth IEEE International Conference on Fuzzy Systems*, 2000, vol. 2, pp. 1045-1048 vol.2, doi: 10.1109/FUZZY.2000.839195.

[34] D. Ramot, R Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE transactions on fuzzy systems*, vol. 10, no. 2, pp. 171-186, 2002.

[35] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *IEEE transactions on fuzzy systems*, vol. 11, no. 4, pp. 450-461, 2003.

[36] S. Dick, "Toward complex fuzzy logic," *IEEE transactions on fuzzy systems*, vol. 13, no. 3, pp. 405-414, 2005.

[37] Grätzer, G, "General Lattice Theory: 1979 Problem Update," *Algebra universalis*, vol. 11, no. 1, pp. 396-402, 1980.

[38] D.E. Tamir, L. Jin and A. Kandel, "A new interpretation of complex membership grade," *International journal of intelligent systems*, vol. 26, no. 4, pp. 285-312, 2011.

[39] D.E. Tamir and A. Kandel, "Axiomatic Theory of Complex Fuzzy Logic and Complex Fuzzy Classes," *International Journal of Computers Communications and Control*, vol. 6, no. 3, p. 562, 2011.

[40] A.S. Alkouri and A.R. Salleh, "Complex intuitionistic fuzzy sets," in *AIP Conference Proceedings*, vol. 1482, pp. 464-470, 2012.

[41] R.R. Yager and A.M. Abbasov, "Pythagorean Membership Grades, Complex Numbers, and Decision Making," *International journal of intelligent systems*, vol. 28, no. 5, pp. 436-452, 2013.

[42] R.R. Yager, "Pythagorean Membership Grades in Multicriteria Decision Making," *IEEE transactions on fuzzy systems*, vol. 22, no. 4, pp. 958-965, 2014.

[43] T. Kumar and R.K. Bajaj, "On Complex Intuitionistic Fuzzy Soft Sets with Distance Measures and Entropies," *Journal of mathematics (Hidawi)*, vol. 2014, pp. 1-12, 2014.

[44] Maji, Pabitra Kumar, "More on Intuitionistic Fuzzy Soft Sets," in *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, vol. 5908, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 231-240.

[45] P. Thirunavukaras and S. Rengarajulu, "Parameterized Soft Complex Fuzzy Sets," *Journal of Progressive Research in Mathematics*, vol. 4, no. 2, pp. 303-308, 2015.

[46] C. Li and C.-H. Tu, "Complex neural fuzzy system and its application on multi-class prediction - A novel approach using complex fuzzy sets, IIM and multi-swarm learning," *Applied soft computing*, vol. 84, pp. 105735, 2019.

[47] C.-H. Tu and C. Li, "Multitarget prediction-A new approach using sphere complex fuzzy sets," *Engineering Applications of Artificial Intelligence*, vol. 79, pp. 45-57, 2019.

[48] M. Ali and F. Smarandache, "Complex neutrosophic set," *Neural Computing and Applications*, vol. 28, no. 7, pp. 1817-1834, 2017.

[49] Z. Ali and T. Mahmood, "Complex neutrosophic generalised dice similarity measures and their application to decision making," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 2, pp. 78-87, 2020.

[50] P. Liu, T. Mahmood, and Z. Ali, "Complex q-rung orthopair fuzzy aggregation operators and their applications in multi-attribute group decision making," *Information*, vol. 11, Art. no. 27., 2020.

[51] P.K. Singh, "Bipolar δ-equal complex fuzzy concept lattice with its application," *Neural computing and applications*, vol. 32, no. 7, pp. 2405-2422, 2019.

[52] S. Greenfield, F. Chiclana, and S. Dick, "Interval-valued complex fuzzy logic," in Proc. *IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, 2016, pp. 2014-2019.

[53] J.Y. Man, Z. Chen and S. Dick, "Towards Inductive Learning of Complex Fuzzy Inference Systems," in *NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society*, 2007, pp. 415-420.

[54] Z. Chen, S. Aghakhani, J. Man and S. Dick, "ANCFIS: A Neurofuzzy Architecture Employing Complex Fuzzy Sets," *IEEE transactions on fuzzy systems*, vol. 19, no. 2, pp. 305-322, 2011.

[55] O. Yazdanbaksh, A. Krahn and S. Dick, "Predicting solar power output using complex fuzzy logic," *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, 2013, pp. 1243-1248, doi: 10.1109/IFSA-NAFIPS.2013.6608579.

[56] O. Yazdanbaksh, S. Dick, "Time-Series Forecasting via Complex Fuzzy Logic," *In: Sadeghian A., Tahayori H. (eds) Frontiers of Higher Order Fuzzy Sets*. Springer, pp. 147-165, 2015.

[57] O. Yazdanbakhsh and S. Dick, "ANCFIS-ELM: A machine learning algorithm based on complex fuzzy sets," *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 2007-2013, doi: 10.1109/FUZZ-IEEE.2016.7737938.

[58] O. Yazdanbakhsh and S. Dick, "Forecasting of Multivariate Time Series via Complex Fuzzy Logic," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2160-2171, Aug. 2017, doi: 10.1109/TSMC.2016.2630668.

[59] O. Yazdanbakhsh and S. Dick, "FANCFIS: Fast adaptive neuro-complex fuzzy inference system," *International journal of approximate reasoning*, vol. 105, pp. 417-430, 2019, doi: 10.1016/j.ijar.2018.10.018.

[60] M. Yeganejou and S. Dick, "Inductive learning of classifiers via complex fuzzy sets and logic," *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2017.8015687.

[61] C. Li and T.W. Chiang, "Complex Neuro-Fuzzy Self-learning Approach to Function Approximation," in *Intelligent Information and Database Systems*, vol. 5991, no. 2, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 289-299.

[62] C. Li and T.W. Chiang, "Function Approximation with Complex Neuro-Fuzzy System Using Complex Fuzzy Sets - A New Approach," *New generation computing*, vol. 29, no. 3, pp. 261-276, 2011.

[63] C. Li and F. Chan, "Complex-Fuzzy Adaptive Image Restoration - An Artificial-Bee-Colony-Based Learning Approach" in *Intelligent Information and Database Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 90-99, 2011.

[64] C. Li, T. Wu and F.-T. Chan, "Self-learning complex neuro-fuzzy system with complex fuzzy sets and its application to adaptive image noise canceling," *Neurocomputing (Amsterdam)*, vol. 94, pp. 121-139, 2012.

[65] C. Li and F.-T. Chan, "Knowledge Discovery by an Intelligent Approach Using Complex Fuzzy Sets," in *Intelligent Information and Database Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 320-329, 2012.

[66] C. Li, T.-W. Chiang and L.-C. Yeh, "A novel self-organizing complex neuro-fuzzy approach to the problem of time series forecasting," *Neurocomputing (Amsterdam)*, vol. 99, pp. 467-476, 2013.

[67] C. Li and T.-W. Chiang, "Complex Neuro-fuzzy ARIMA Forecasting-A New Approach Using Complex Fuzzy Sets," *IEEE transactions on fuzzy systems*, vol. 21, no. 3, pp. 567-584, 2013.

[68] C. Li and C.-H. Tu, "Complex neural fuzzy system and its application on multi-class prediction - A novel approach using complex fuzzy sets, IIM and multi-swarm learning," *Applied soft computing*, vol. 84, p. 105735, 2019, doi: 10.1016/j.asoc.2019.105735.

[69] R. Shoorangiz and M.H. Marhaban, "Complex Neuro-Fuzzy System for Function Approximation," *International Journal of Applied Electronics in Physics and Robotics*, vol. 1, no. 2, pp. 5-9, 2013.

[70] K. Subramanian, R. Savitha, and S. Suresh, "A complex-valued neuro-fuzzy inference system and its learning mechanism," *Neurocomputing (Amsterdam)*, vol. 123, pp. 110-120, 2014.

[71] H. Ryusuke and M. Kazuyuki, "Generation of Fuzzy Rules by a Complex-Valued Neuro-Fuzzy Learning Algorithm," *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, vol. 27, no. 1, pp. 533-548, 2015.

[72] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *MIT Press*, 2016, ch. 5, pp. 130–131. http://www.deeplearningbook.org.

[73] S. Lloyd, "Least squares quantization in PCM," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129-137, 1982, doi: 10.1109/TIT.1982.1056489.

[74] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. . Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881-892, 2002, doi: 10.1109/TPAMI.2002.1017616.

[75] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, New Orleans, Louisiana, 2007, pp. 1027-1035.

[76] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017, doi: 10.1145/3065386.

[77] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. arXiv:1412.6980

[78] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *U.S.S.R. computational mathematics and mathematical physics*, vol. 4, no. 5, pp. 1-17, 1964, doi: 10.1016/0041-5553(64)90137-5.

[79] Yu. E. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," In *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543-547, 1983.

[80] B. Hu and L. Lessard, "Control interpretations for first-order optimization methods," in *2017 American Control Conference (ACC)*, 2017, pp. 3114-3119, doi: 10.23919/ACC.2017.7963426.

[81] K. Pearson, "LIII. On Lines and Planes of Closest Fit to Systems of Points in Space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901, vol. 2, pp. 559-572.

[82] I. T. Jolliffe,*Principal component analysis*. New York: Springer, 1986.

[83] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews. Computational statistics*, vol. 2, no. 4, pp. 433-459, 2010, doi: 10.1002/wics.101.

[84] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural computation*, vol. 10, no. 5, pp. 1299-1319, 1998, doi: 10.1162/089976698300017467.

[85] J. M. Bland and D. G. Altman, "Statistics Notes: Measurement error," *BMJ*, vol. 313, no. 7059, pp. 744-744, 1996, doi: 10.1136/bmj.313.7059.744.

[86] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679-688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.

[87] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79-82, 2005, doi: 10.3354/cr030079.

[88] B. E. Flores, "A pragmatic view of accuracy measurement in forecasting," *Omega (Oxford)*, vol. 14, no. 2, pp. 93-98, 1986, doi: 10.1016/0305-0483(86)90013-7.

[89] W. H. Kruskal and W. A. Wallis, "Use of Ranks in One-Criterion Variance Analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952, doi: 10.1080/01621459.1952.10483441.

[90] SIDC (World Data Center for the Sunspot Index), http://sidc.oma.be/index.php3S

[91] R.J. Hyndman; G. Athanasopoulos, "Seasonal ARIMA models", Forecasting: principles and practice. oTexts. Retrieved 19 May 2015.

[92] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks," *Neurocomputing (Amsterdam)*, vol. 73, no. 13, pp. 2540–2553, 2010, doi: 10.1016/j.neucom.2010.06.004.

[93] D. A. Brandt and J.C. Warner,*Metallurgy fundamentals, 5th edition*. Tinley Park, IL: Goodheart-Willcox, 2009.

[94] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[95] J. Moody and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural computation*, vol. 1, no. 2, pp. 281-294, 1989 , doi: 10.1162/neco.1989.1.2.281.

[96] D. F. Specht, "A general regression neural network," in *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568-576, Nov. 1991, doi: 10.1109/72.97934.

[97] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[98] G. Hinton, "Deep belief networks," *Scholarpedia journal*, vol. 4, no. 5, pp. 5947, 2009, doi: 10.4249/scholarpedia.5947.

[99] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support Vector Regression Machines," in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, 1996, pp. 155–161.

[100] G. Panoutsos and M. Mahfouf, "A neural-fuzzy modelling framework based on granular computing: Concepts and applications," *Fuzzy sets and systems*, vol. 161, no. 21, pp. 2808-2830, 2010, doi: 10.1016/j.fss.2010.06.004.

[101] R. Muscat and M. Mahfouf, "Predicting Charpy Impact Energy for Heat-Treated Steel using a Quantum-Membership-Function-based Fuzzy Model," in *IFAC-Papers[online]*,pp.138-142,2016,https://doi.org/j.ifacol.2016.10.110.

[102] Qilian Liang and J. M. Mendel, "An introduction to type-2 TSK fuzzy logic systems," *FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No.99CH36315)*, 1999, pp. 1534-1539 vol.3, doi: 10.1109/FUZZY.1999.790132.

[103] M. A. Meyers and K. K. Chawla, "Mechanical Behaviour of Materials," *Proceedings of the Fourth International Conference*, Stockholm, Sweden, 15-19 August 1983.

[104] J. Chen, "Biological inspired optimisation algorithms for transparent knowledge extraction allied to engineering materials process," Ph.D. dissertation, Department of Automatic Control and Systems Engineering, The University of Sheffield, U.K., 2009.

[105] S. Wang and M. Mahfouf, "Multi-objective optimisation for fuzzy modelling using interval type-2 fuzzy sets," 2012. doi: 10.1109/FUZZ-IEEE.2012.6251165.