

Circular Complex-Valued GMDH-Type Neural Network for Real-Valued Classification Problems

Jin Xiao^{1b}, *Member, IEEE*, Yanlin Jia, Xiaoyi Jiang^{2b}, *Senior Member, IEEE*, and Shouyang Wang^{3b}, *Member, IEEE*

Abstract—Recently, applications of complex-valued neural networks (CVNNs) to real-valued classification problems have attracted significant attention. However, most existing CVNNs are black-box models with poor explanation performance. This study extends the real-valued group method of data handling (RGMDH)-type neural network to the complex field and constructs a circular complex-valued group method of data handling (C-CGMDH)-type neural network, which is a white-box model. First, a complex least squares method is proposed for parameter estimation. Second, a new complex-valued symmetric regularity criterion is constructed with a logarithmic function to represent explicitly the magnitude and phase of the actual and predicted complex output to evaluate and select the middle candidate models. Furthermore, the property of this new complex-valued external criterion is proven to be similar to that of the real external criterion. Before training this model, a circular transformation is used to transform the real-valued input features to the complex field. Twenty-five real-valued classification data sets from the UCI Machine Learning Repository are used to conduct the experiments. The results show that both RGMDH and C-CGMDH models can select the most important features from the complete feature space through a self-organizing modeling process. Compared with RGMDH, the C-CGMDH model converges faster and selects fewer features. Furthermore, its classification performance is statistically significantly better than the benchmark complex-valued and real-valued models. Regarding time complexity, the C-CGMDH model is comparable with other models in dealing with the data sets that have few features. Finally, we demonstrate that the GMDH-type neural network can be interpretable.

Index Terms—Complex-valued external criterion, complex-valued group method of data handling (GMDH)-type neural network, parameter estimation, real-valued classification.

I. INTRODUCTION

COMPLEX numbers are used to express real-world phenomena (such as signal amplitude and phase) and to analyze various mathematical and geometrical relationships. To process directly complex values with artificial neural networks (ANNs), complex-valued neural networks (CVNNs) have been developed [1]–[3]. CVNNs' states, connection weights, and activation functions are complex-valued. Thus, they not only deal with complex-valued signals but also have the following excellent properties [4]–[6]: 1) the average convergence speed is two or three times faster than the real-valued neural networks (RVNNs); 2) the number of required hidden parameters is approximately half that of the RVNNs; 3) the orthogonal decision boundaries of the CVNNs help them to solve classification problems more efficiently than their real-valued counterparts; and 4) the CVNNs have a better generalization ability than the RVNNs. In addition, the CVNNs, as extensions of RVNNs, have some other advantages, such as their strong nonlinear modeling and generalization capabilities. Recently, CVNNs have attracted an increasing attention. Therefore, it is very important to study the CVNNs further.

A. Literature Review

In the last two decades, CVNNs have been widely applied for tasks such as fading channel prediction [7], skeletal wireframe pattern recognition [8], signal coherence [6], biomedical pattern classification [9], image processing [10], optical label recognition [11], and developmental motion learning [12]. As research continues, several CVNN models have been proposed and studied. The models include the complex-valued convolutional neural network [13], complex-valued feedforward neural network [6], [14], complex least mean square algorithm [15], [16], complex-valued recurrent neural network [17], complex-valued projection neural network [18], complex-valued Hopfield neural network [19]–[21], complex-valued radial basis function neural network [22], [23], complex-valued self-organizing map [24], and complex-valued associative memories [25], [26].

Most of these studies directly dealt with complex-valued data. Recently, researchers have also applied CVNNs to real-valued classification problems, finding that a CVNN can outperform RVNNs in this task [27]. Therefore, the application of CVNNs to real-valued classification has attracted increasing attention [27]–[42]. These studies often map the inputs and outputs of a real-valued classification problem to the complex field with a function; subsequently, a CVNN model is constructed for classification. In general, these functions can be classified into two types:

Manuscript received May 4, 2018; revised April 23, 2019 and August 19, 2019; accepted January 7, 2020. Date of publication February 17, 2020; date of current version December 1, 2020. This work was supported in part by the Major Project of the National Social Science Foundation of China under Grant 18VZL006, in part by the EU Horizon 2020 RISE Project ULTRACEPT under Grant 778062, in part by the National Natural Science Foundation of China under Grant 71471124, in part by the Tianfu Ten-Thousand Talents Program of Sichuan Province, in part by the Excellent Youth Fund of Sichuan University under Grant skqx201607, Grant sksyl201709, and Grant skzx2016-rcrw14, in part by the Leading Cultivation Talents Program of Sichuan University, and in part by the Teacher and Student Joint Innovation Project of Business School of Sichuan University under Grant LH2018011. (*Corresponding author: Shouyang Wang.*)

Jin Xiao and Yanlin Jia are with the Business School, Sichuan University, Chengdu 610064, China (e-mail: xiaojin@scu.edu.cn; yelinjyl@126.com).

Xiaoyi Jiang is with the Faculty of Mathematics and Computer Science, University of Münster, 48149 Münster, Germany (e-mail: xjiang@uni-muenster.de).

Shouyang Wang is with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: sywang@amss.ac.cn).

This article has supplementary downloadable material available at <https://ieeexplore.ieee.org>, provided by the authors.

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNNLS.2020.2966031

The first type is a phase-encoded transformation [27]–[36]. First, real-valued input feature x^R is encoded to be phase φ of the unity magnitude complex number. Next, the complex-valued input feature x is obtained by Euler's formula: $e^{i\varphi} = \cos \varphi + i * \sin \varphi$. For example, $x = \exp(i\pi x^R)$, where x^R is normalized in $[0, 1]$, and x is bounded and distributed along the unit circle in quadrants I and II of the complex plane [30].

The second type is a nonlinear transformation [37]–[42]. First, the real-valued input feature x^R is mapped to be the complex number by a linear function, $cx^R + d$, where c is a random complex number and d is a constant. Next, a final complex-valued input feature x is obtained by a nonlinear function. For instance, $x = \sin(ax^R + ibx^R + \alpha)$, where x^R is normalized in $[0, 1]$, $a, b \in [0, 1]$ are the randomly chosen scaling constants, and $\alpha \in [0, 2\pi]$ is used to shift the origin to effectively use the four quadrants of the complex plane [39].

B. Motivation

Previous work significantly advanced the state of the art for the CVNNs and their applications, especially for real-valued classification problems. However, some limitations of the CVNNs remain. First, these models pose difficulties in confirming the optimal training parameters [43], such as the number of hidden layers and the number of neurons in each layer. Thus, the model must be debugged repeatedly to find the optimal parameters. Second, they are prone to overfitting the noise data in the training set [44]. The overfitted neural network model will fit the training data perfectly, but it fails to predict well the unseen testing data. Third, because most ANNs are black-box models [45], so are the CVNNs, these networks cannot be interpretable. Thus, the explanatory power of the modeling results is poor.

In recent years, deep learning has received widespread attention [46]–[49], and the group method of data handling (GMDH)-type neural network, a heuristic self-organizing modeling technology, is perhaps the first deep learning system [50]. To overcome the above shortcomings, the GMDH-type neural network is a good approach. It was first developed by Ivakhnenko [51] as a method of multivariate analysis for modeling and identification of complex systems. In the GMDH-type neural networks, training set T_r is randomly divided into two subsets: model learning set A and model selecting set B . The basic idea of this algorithm is to build a multilayer feedforward network structure. It starts with the input layer, generates new candidate models in each layer by combining two models of the previous layer, and estimates the model parameters in A . In particular, the transfer function between the input and output variables can be expressed by the Kolmogorov–Gabor (KG) polynomial [52], [53]. Next, the algorithm uses an external criterion to evaluate and select the middle candidate models in B . Finally, a model with an optimal complexity is found by the termination principle [54], [55]. Therefore, GMDH-type neural networks can automatically determine the number of layers of the neural network, the variables that enter the optimal complexity model, and the model parameters. In summary, GMDH-type neural networks add several advantages to contemporary ANNs [52], [56]–[59]: 1) they have strong antinoise capabilities and generate polynomial equations that are interpretable better than any other ANN model (i.e., white-box models) and 2) they learn the weights rapidly in a single step by standard ordinary least squares (LS), which eliminates the need to search for their

values and guarantees finding locally good weights thanks to the reliability of the fitting technique. Thus, this approach can avoid the limitations of traditional ANNs to some extent. In recent years, the real-valued GMDH (RGMDH)-type neural network has been successfully applied in various fields, such as engineering, science, and economics [60]–[66].

Therefore, if an RGMDH model is extended to the complex field, the complex-valued GMDH-type neural network may outperform the RGMDH and avoid the limitations of the existing CVNNs to some extent. Recently, Xiao *et al.* [67] proposed a phase-encoded complex-valued GMDH (PE-CGMDH)-type neural network for real-valued classification. The experimental results show that PE-CGMDH outperforms RGMDH and other four CVNN and RVNN models.

C. Our Contributions

In fact, several shortcomings of the PE-CGMDH model remain as follows.

- 1) A complex-valued symmetric regularity criterion (CSRC) was constructed, but the properties of the CSRC were not studied, that is, whether it has sufficient property similarity to the real external criterion [68] or its global minimum value exists.
- 2) The model used a simple mean square error deviation between the actual and predicted outputs (i.e., regularity criterion) as the external criterion in the complex field. This approach has problems with phase approximation, because it explicitly minimizes only the magnitude error.
- 3) The phase encoding for the transformation of real-valued inputs may be ineffective, because this phase encoding only restricts the projection to I and II quadrants.

To overcome the shortcomings of the PE-CGMDH model, a new model, the circular complex-valued GMDH (C-CGMDH)-type neural network, is proposed in this study. First, the parameter estimation method of the RGMDH model is extended to the complex field. Next, a new reasonable complex-valued external criterion is proposed and studied theoretically to solve the first and second problems. Finally, a circular transformation is adopted to transform the real-valued input features into the complex field to use effectively all four quadrants of the complex plane. The experimental results on 25 real-valued UCI classification data sets show that C-CGMDH significantly outperforms PE-CGMDH and RGMDH, as well as other complex-valued and real-valued models.

The novelty of this article can be summarized as follows.

- 1) This article provides sufficient theoretical research of the properties of the CSRC, which makes the work technically sound and complete, and proposes a new CSRC (NCSRC) with the logarithmic function to overcome the limitations of the CSRC.
- 2) The circular transformation is introduced for the first time to construct a C-CGMDH model for overcoming the limitations of phase-encoded transformation.

D. Structure of the Article

This article is organized as follows. Section II describes the model of RGMDH. Section III presents the C-CGMDH model in detail, including the parameter estimation, the construction of the external criterion, and the modeling process. The experiments are shown in Section IV to demonstrate the performance of the C-CGMDH model, including the ablation

study, convergence speed, feature selection, classification performance, time complexity, and explanation performance. Finally, the conclusions are summarized in Section V.

II. RGMDH-TYPE NEURAL NETWORK

A model can be represented as a set of neurons by the RGMDH model; different pairs of neurons in each layer are connected with a polynomial to produce new neurons in the next layer. Let $X^R = (x_1^R, x_2^R, \dots, x_n^R)$ and Y^R be the real-valued input vector and the actual output, respectively. Given m observations of “multidimensional input, single output” data pairs $\{(y_j^R, x_{j1}^R, x_{j2}^R, \dots, x_{jn}^R)\} (j = 1, 2, \dots, m)$ in the model learning set, $A^R \in R$, we hope to train an RGMDH model to predict the output values \hat{y}_j^R

$$\hat{y}_j^R = \hat{f}(x_{j1}^R, x_{j2}^R, \dots, x_{jn}^R), \quad j = 1, 2, \dots, m. \quad (1)$$

The challenge now is to validate an RGMDH model

$$\sum_{j=1}^m [\hat{f}(x_{j1}^R, x_{j2}^R, \dots, x_{jn}^R) - y_j^R]^2 \rightarrow \min. \quad (2)$$

The transfer function between the input and output variables can be expressed by a complicated discrete form of the Volterra functional series, as follows:

$$y^R = a_0 + \sum_{i=1}^n a_i x_i^R + \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i^R x_j^R + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n a_{ijk} x_i^R x_j^R x_k^R + \dots \quad (3)$$

which is also known as the KG polynomial [52], [53]. In particular, the linear transfer function only consists of two variables (i.e., features and neurons). We have x_{jp}^R, x_{jq}^R for different vales of $p, q \in \{1, 2, \dots, n\}, p \neq q$, in the form of

$$\hat{y}_j^R = g_j(x_{jp}^R, x_{jq}^R) = a_0 + a_1 x_{jp}^R + a_2 x_{jq}^R \quad (4)$$

where a_0, a_1, a_2 are calculated with LS [52], [53]. Thus, the coefficients of each linear function, g_j , are obtained to optimally fit output y_j^R in set A^R . That is

$$E = \sum_{j=1}^m (y_j - g_j(x_{jp}, x_{jq}))^2 \rightarrow \min. \quad (5)$$

In the basic form of the RGMDH model, any two out of n input variables are combined to construct the regression polynomial in the form of (4). Consequently, $C_n^2 = n(n-1)/2$ candidate models will be built up in the first layer of the feedforward network from the observations, $\{(y_j^R, x_{jp}^R, x_{jq}^R)\}$. In other words, it is now possible to construct m data triples, $\{(y_j^R, x_{jp}^R, x_{jq}^R)\}$, from the observations

with p, q in the form $\begin{bmatrix} y_1^R & x_{1p}^R & x_{1q}^R \\ y_2^R & x_{2p}^R & x_{2q}^R \\ \vdots & \vdots & \vdots \\ y_m^R & x_{mp}^R & x_{mq}^R \end{bmatrix}_{m \times 3}$. Furthermore,

the corresponding matrix equation can be readily obtained as $D^R a = Y^R$, where $a = (a_0, a_1, a_2)^T$ is the coefficient vector, $Y^R = (y_1^R, y_2^R, \dots, y_m^R)^T$ is the vector of observations, and

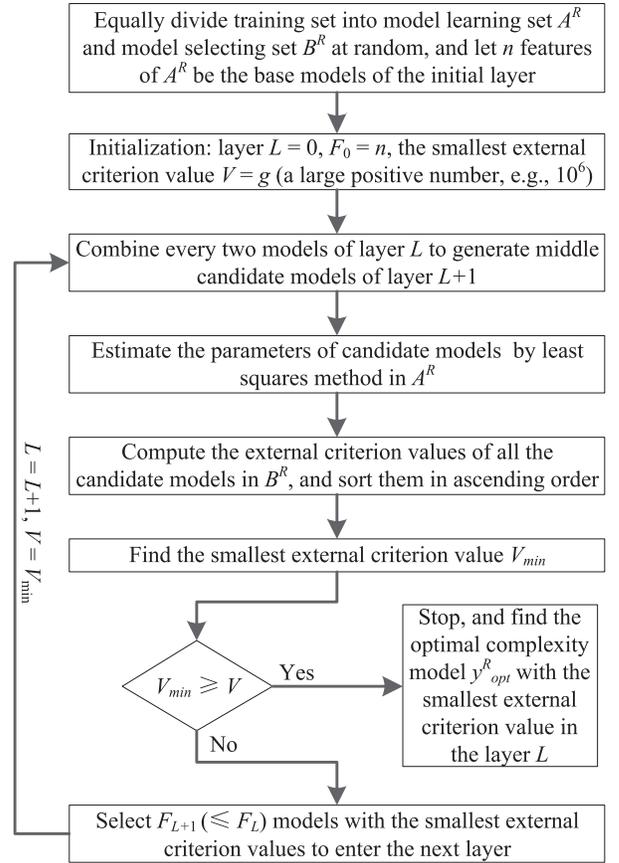


Fig. 1. Modeling process of the RGMDH-type neural network.

$D^R = \begin{bmatrix} 1 & x_{1p}^R & x_{1q}^R \\ 1 & x_{2p}^R & x_{2q}^R \\ \vdots & \vdots & \vdots \\ 1 & x_{mp}^R & x_{mq}^R \end{bmatrix}_{m \times 3}$. The LS leads to the solution of the normal equations, as follows:

$$a = (D^{RT} D^R)^{-1} D^{RT} Y^R \quad (6)$$

which determines the vector of the best coefficients of (4) for the entire set of m data triples.

The above procedure is repeated for each neuron of the next hidden layer per the connectivity topology of the network. In each layer, the algorithm uses LS to estimate the parameters of the candidate models in the model learning set, A^R , and uses an external criterion to evaluate and select the candidate models in the model selecting set, B^R . The algorithm continues and stops when it finds the optimal model by the termination principle, presented by the theory of optimal complexity [52]. Along with the increase in model complexity, the value of the external criterion will first decrease and then increase, and the global extreme value will correspond to the optimal complexity model [53]. The modeling process of the RGMDH model is shown in Fig. 1.

III. C-CGMDH-TYPE NEURAL NETWORK

By Occam's razor principle, entities should not be multiplied beyond necessity [69]; thus, complex models are not necessarily superior to simple models. Hence, in the proposed

C-CGMDH model for real-valued classification, the transfer function between the input and output variables is also described by the linear function in (4), as in the RGMDH model. In the C-CGMDH model, parameter estimation and selection of the external criterion are two key steps. Therefore, in this section, we first extend the LS parameter estimation method of the RGMDH model to the complex field. Furthermore, we introduce complex-valued external criterion construction for the C-CGMDH model, including a relation between the complex-valued external criterion and the complexity of the model. We propose a new reasonable complex-valued external criterion. Finally, we summarize the detailed modeling steps of the C-CGMDH model for real-valued classification.

A. Complex LS Parameter Estimation

In the RGMDH model, all the middle candidate models are trained on the model learning set, A^R ; the parameters are estimated by LS. In the C-CGMDH model, we need to adopt a parameter estimation method similar to the one of RGMDH.

To meet this requirement, let $A = (Y_A, x_1, x_2, \dots, x_n) \in C^{m \times (n+1)}$ be the model learning set in the complex field and let A contain m training samples. Suppose x_p, x_q ($p, q \in \{1, 2, \dots, n\}$, and $p \neq q$) are the two initial variables. The transfer function between the inputs x_p, x_q and the output Y_A can be expressed by (4), as follows:

$$Y_A = g(x_p, x_q) = a_0 + a_1 x_p + a_2 x_q \quad (7)$$

where $a_i \in C, i = 0, 1, 2$.

Similarly, the corresponding complex matrix equation can be readily obtained as $Da = Y_A$, where $a = (a_0, a_1, a_2)^T$ is the coefficient vector, $Y_A = (y_1, y_2, \dots, y_m)^T$ is the vector of

the complex-valued output, and $D = \begin{bmatrix} 1 & x_{1p} & x_{1q} \\ 1 & x_{2p} & x_{2q} \\ \vdots & \vdots & \vdots \\ 1 & x_{mp} & x_{mq} \end{bmatrix}_{m \times 3}$.

Then, by the theory of matrices [70], we can obtain a solution to the complex matrix equation, as follows:

$$a = (D^* D)^{-1} D^* Y_A \quad (8)$$

where $(.)^*$ and $(.)^{-1}$ denote the conjugate transpose matrix and inverse matrix of some matrix, $(.)$, respectively. The rank of D mostly reflects the column number, because the number of the samples is usually much larger than that of the variables. Thus, matrix $D^* D$ is mostly nonsingular. Because $D^* D$ may be singular, we compute the so-called Moore–Penrose pseudoinverse [71]. The method mentioned in [67] can also be used.

B. Selection of the External Criterion

When we model real-valued systems, different requirements may be proposed, reflecting the aims of the model or prior knowledge of the systems. In a GMDH-type neural network, an external criterion is a mathematical description of these specific requirements that can select the optimal model among the candidate models according to the theory of optimal complexity [53]. Therefore, when we construct a GMDH-type neural network model, the selection of the external criterion is a key issue [52].

In this study, we select the SRC [53] to measure the performance of the middle candidate models. For the training

set T_r^R of the real-valued problems, the SRC has the following form:

$$\begin{aligned} d^2(T_r^R) &= \Delta^2(B^R|A^R) + \Delta^2(A^R|B^R) \\ &= \|X_{B^R} \hat{a}_{A^R} - y_{B^R}\|^2 + \|X_{A^R} \hat{a}_{B^R} - y_{A^R}\|^2 \end{aligned} \quad (9)$$

where y_{B^R} and y_{A^R} are the actual outputs of the model selecting set B^R and the model learning set A^R . Next, X_{B^R} and X_{A^R} are the input vectors of B^R and A^R . Then, \hat{a}_{A^R} and \hat{a}_{B^R} are the coefficient vectors of the model trained on A and B , respectively. Therefore, $\Delta^2(B^R|A^R)$ in (9) indicates the classification error on B^R of the model constructed on A^R , and $\Delta^2(A^R|B^R)$ indicates the classification error on A^R of the model constructed on B^R . As shown, the SRC uses the information in sets A^R and B^R equally and considers the error of the model in different parts. Mueller and Lemke demonstrated that the SRC satisfied the theory of optimal complexity [53].

To construct a CSRC, the complete complex-valued training set T_r is equally but randomly partitioned into two subsets: model learning set A and model selecting set B . Furthermore, let Y_A be the actual output of A , let \hat{Y}_A be the prediction output on A by the model trained on B , let Y_B be the actual output on B , and let \hat{Y}_B be the prediction output on B by the model trained on A . Then, we define the CSRC as follows:

$$d^2(T_r) = \Delta^2(B|A) + \Delta^2(A|B) = \varepsilon_B^* \varepsilon_B + \varepsilon_A^* \varepsilon_A \quad (10)$$

where $\varepsilon_B = \hat{Y}_B - Y_B$ and $\varepsilon_A = \hat{Y}_A - Y_A$ are the prediction errors in B and A , respectively. The next question is whether the CSRC has a good property that is similar to the SRC [53].

To solve this problem, we want to first investigate the relation between the complexity of the model and the former part of the CSRC [i.e., $\Delta^2(B|A)$], which is also called the complex-valued asymmetric regularity criterion (CARC). Hence, consider (3) and (7) again, and suppose that the number of variables for learning the model (i.e., the complexity of the model) is $S - 1$; then the model is of the form

$$Y = (l_{m \times 1} \quad X_{m \times (S-1)}) a_S = D_S a_S \quad (11)$$

where $l_{m \times 1} = (1, 1, \dots, 1)^T$, $X_{m \times (S-1)} \in C^{m \times (S-1)}$, $D_S = (l_{m \times 1} \quad X_{m \times (S-1)})$, $Y \in C^{m \times 1}$, and $a_S = (a_0, a_1, \dots, a_{S-1})^T \in C^{S \times 1}$. Then, the rank of D_S is predominantly S , because the number of the samples is usually much larger than that of the variables.

In the CARC, data set A is used to learn the model first. The estimates of the coefficients are obtained by (8)

$$\hat{a}_{AS} = (D_{AS}^* D_{AS})^{-1} D_{AS}^* Y_A = H_{AS}^{-1} D_{AS}^* Y_A \quad (12)$$

where $H_{AS} = D_{AS}^* D_{AS}$. Therefore, the estimate of the output in G is

$$\hat{Y}_G = D_{GS} \hat{a}_{AS} = D_{GS} (D_{AS}^* D_{AS})^{-1} D_{AS}^* Y_A = P_{GAS} Y_A \quad (13)$$

where $P_{GAS} = D_{GS} (D_{AS}^* D_{AS})^{-1} D_{AS}^*$, and G can be A , B , or $T_r (= A \cup B)$.

Next, the property of the CARC in the mean sense (i.e., the mathematical expectation) is analyzed in this study to discuss the relation between the CARC and the complexity. In addition, a natural approach is to check the change in the CARC when the complexity increases.

Typically, data contain noise. Hence, we have the following propositions about CARC.

Proposition 1: The expected value of the CARC is related to the model structure and the noise. The proof is given in supplementary material.

Theorem 1: If U is an $n \times m$ complex matrix and V is an $m \times n$ complex matrix, then $\text{tr}(UV) = \text{tr}(VU)$.

Proof: According to the definition of trace, we have

$$\text{tr}(UV) = \sum_{i=1}^n \sum_{j=1}^m U_{ij} V_{ji} = \sum_{j=1}^m \sum_{i=1}^n V_{ji} U_{ij} = \text{tr}(VU).$$

Here, we first analyze the special situation $G = A$ and then expand to $G = B$.

Proposition 2: If $G = A$, then as the complexity of the model increases; there is a minimum expected value for $\Delta^2(G|A)$. It first monotonically decreases and then monotonically increases. The proof is given in supplementary material.

Proposition 3: If $G = B$, then as the complexity of the model increases, there is a minimum expected value for $\Delta^2(B|A)$. It first decreases, but the decrease may not be monotonic. Then, it monotonically increases. The proof is given in supplementary material.

Because (10) is symmetrical, there is a similar conclusion to the latter part of CSRC, $\Delta^2(A|B)$, with the increase in the complexity, S . Thus, we can obtain Proposition 4.

Proposition 4: The property of the CSRC is similar to the SRC. The proof is given in supplementary material.

In summary, the CSRC satisfies the theory of optimal complexity [52] and can be used as an external criterion for the CGMDH-type neural network.

However, CSRC is limited: it explicitly minimizes only the magnitude error. For a better phase approximation, we try to improve the CSRC in this study. To minimize simultaneously both the magnitude and phase errors, we use the logarithmic function to represent explicitly the magnitude and phase of the actual and predicted complex outputs

$$\begin{aligned} \Delta_{\ln}^2(B|A) &= \|\ln \hat{Y}_B - \ln Y_B\|^2 \\ &= \sum_{j=1}^{m'} ((\ln r_{\hat{Y}_{B_j}} - \ln r_{Y_{B_j}})^2 + (\varphi_{\hat{Y}_{B_j}} - \varphi_{Y_{B_j}})^2) \end{aligned} \quad (14)$$

where $\ln(\cdot)$ represents the natural logarithmic function, m' is the sample number of B , and $\hat{Y}_{B_j} = r_{\hat{Y}_{B_j}} \exp(i\varphi_{\hat{Y}_{B_j}})$, $Y_{B_j} = r_{Y_{B_j}} \exp(i\varphi_{Y_{B_j}})$, respectively.

Proposition 5: The improved CARC [see (14)] satisfies Propositions 1 and 3 of the CARC. The proof is given in supplementary material.

Therefore, we construct an NCSRC as follows:

$$\begin{aligned} \text{NCSRC} &= \Delta_{\ln}^2(B|A) + \Delta_{\ln}^2(A|B) \\ &= \|\ln \hat{Y}_B - \ln Y_B\|^2 + \|\ln \hat{Y}_A - \ln Y_A\|^2. \end{aligned} \quad (15)$$

In addition, to facilitate the comparison of experimental results, we first divide the NCSRC by the sample number of $A \cup B$ (i.e., a constant) and then obtain the square root of the corresponding result. Thus, the final form of the NCSRC used in this study is of the form

$$\text{NCSRC} \leftarrow \sqrt{\frac{\text{NCSRC}}{m + m'}}. \quad (16)$$

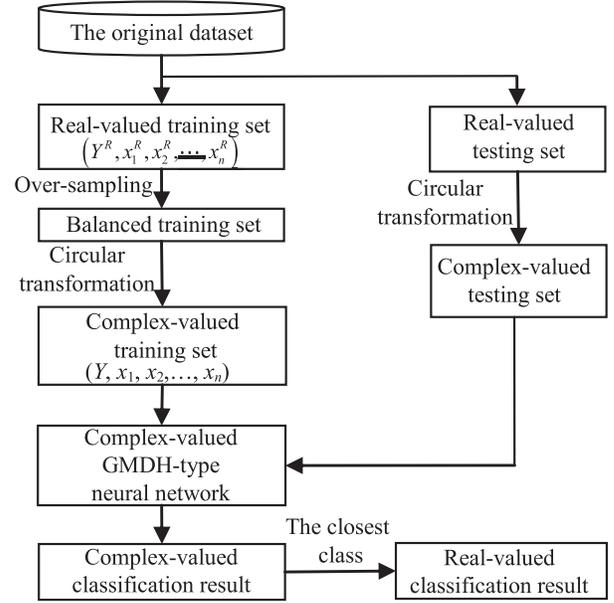


Fig. 2. Process of C-CGMDH model for real-valued classification.

Importantly, (16) does not affect the monotonicity of the NCSRC about the complexity of the model theoretically by the calculus.

C. Modeling Process

Suppose that a real-valued classification data set, $D = \{y^R, x_1^R, x_2^R, \dots, x_n^R\} \in R^{n+1}$, includes two parts: training set T_r^R and testing set T_e^R . There are $m + m'$ and u samples in T_r^R and T_e^R , respectively. Then, the process of the C-CGMDH model for real-valued classification proposed in this study is shown in Fig. 2.

In many real-world classification problems, the data sets are usually imbalanced. Some classes have much more instances than others [72]. The level of imbalance (i.e., ratio of the majority class to the minority class) can be as huge as 10^6 [73]. The class imbalance has a serious impact on the model performance. To solve this issue, scholars have proposed several methods [74], mostly resampling techniques and cost-sensitive learning. Among those, resampling techniques are used more often, and the commonly used ones are random undersampling and random oversampling. Marqués *et al.* [75] compared logistic regressions and support vector machine (SVM) with and without resampling on five real customer classification data sets and demonstrated superior performance with oversampling compared with undersampling techniques. Hence, we first adopt random oversampling to balance the class distribution of the training data set T_r^R in this study.

In general, training a C-CGMDH model for real-valued classification problems requires the class labels to be coded in the complex field and the real-valued input features to also be mapped onto the complex space, $R^m \rightarrow C^m$. As for the class labels, taking the training set T_r^R of a binary classification problem as an example, with labels 1 and -1 , coded class label y_j in the complex field is given by

$$y_j = \begin{cases} 1 + 1i, & \text{if } y_j^R = 1 \\ -1 - 1i, & \text{if } y_j^R = -1 \end{cases} \quad (17)$$

where $y_j^R (j = 1, 2, \dots, m + m')$ is the real-valued actual output. That is, in the complex field, label 1 and label -1 are $L_1 = 1 + 1i$ and $L_2 = -1 - 1i$, respectively.

For real-valued input feature $x_{jk}^R (j = 1, 2, \dots, m + m', k = 1, 2, \dots, n)$ of T_r^R , we adopt the circular transformation in this study, because it performs a one-to-one mapping of the real-valued inputs to the complex field and effectively uses all the four quadrants of the complex plane. It can overcome the issues in the existing transformations [39]. Let $x_{jk}^R \in [r_1, r_2]$, $r_1, r_2 \in R$, then $x_{jk}^{R'} = (x_{jk}^R - r_1)/(r_2 - r_1)$ and the complex-valued input feature, x_{jk} , is obtained by the circular transformation

$$x_{jk} = \sin((a + ib)x_{jk}^{R'} + \alpha_k) \quad (18)$$

where $a, b \in [0, 1]$, $\alpha_k \in [0, 2\pi]$ are all randomly chosen.

Next, we train the C-CGMDH model with the complex-valued inputs and outputs until the optimal complexity model y_{opt} is found. Next, we classify the testing set with y_{opt} . The prediction output of the model is a complex number. Thus, the final prediction result (i.e., the real-valued class label) is obtained as follows:

$$\hat{y}_j^R = \arg \min_{i \in \{1, 2\}} d(\hat{y}_j, L_i) \quad (19)$$

where \hat{y}_j is the prediction output by the model y_{opt} , and $d(\hat{y}_j, L_i) = \|\ln \hat{y}_j - \ln L_i\|^2$, $j = 1, 2, \dots, u$.

Finally, the detailed steps of the C-CGMDH model for real-valued classification are shown next.

1) Data Preprocessing:

- a) Balance the class distribution of the real-valued training set T_r^R by random oversampling.
- b) Transform real-valued training set T_r^R and testing set T_e^R into complex-valued sets, T_r and T_e , by (17) and (18).

2) Train the CGMDH-Type Neural Network:

- a) Equally divide the complex-valued training set T_r into model learning set A and model selecting set B at random, and let n features of A be the base models of the initial layer.
- b) Set the layer $L = 0$, $F_0 = n$, and the smallest external criterion value $V = g$ (g is a large positive number, such as $g = 10^6$).
- c) Combine every two models of layer L by (7) to generate $n_{L+1} = C_{F_L}^2$ middle candidate models of layer $L + 1$ and estimate the parameters by (8) in A .
- d) Compute the NCSRC values of all candidate models by (16) in B and sort them in ascending order.
- e) Consider the smallest external criterion value V_{\min} of layer $L + 1$, and if $V_{\min} \geq V$, then STOP and find the optimal complexity model y_{opt} with the smallest external criterion value in the layer L ; else CONTINUE.
- f) Select $F_{L+1} (\leq F_L)$ models with the smallest external criterion values to enter the next layer.
- g) Repeat Steps 3)–6) with $L = L + 1$ and $V = V_{\min}$.

3) Classify the Testing Set:

- a) Classify the complex-valued testing set, T_e , by the optimal complexity model y_{opt} and obtain the complex-valued classification result.
- b) Obtain the real-valued classification result by (19).

TABLE I
DATA SETS USED IN THE EXPERIMENTS

Datasets	Features	Classes	Instances	IF
Adult	14	2	48842	3.18
Australia-c	15	2	690	1.25
Bank-mark	20	2	41188	7.88
Breast-c	9	2	683	1.9
Bupa	6	2	345	1.38
Connectionist	60	2	208	1.14
Contraceptive	9	3	1473	1.89
Defaultcredit	23	2	29999	3.52
Haberman-s	3	2	306	2.78
Hayes-roth	4	3	132	1.7
Hepatitis	19	2	155	1.2
HTRU2	8	2	17898	9.92
Ionosphere	34	2	351	1.79
Iris	4	3	150	1
MAGIC	10	2	19020	1.84
Monk3	6	2	554	1.08
Pima	8	2	768	1.87
QSAR	41	2	1055	1.96
Qualitative-b	6	2	250	1.34
Seeds	7	3	210	1
Teaching-a	5	3	151	1.06
Vehicle	18	4	846	1.1
Vote	16	2	435	1.6
Wine	13	3	178	1.48
Zoo	16	7	101	10.25

IV. EXPERIMENTS

We demonstrated the classification performance of the C-CGMDH model on a set of binary and multiple-category classification data sets. First, we did the ablation study to analyze the influence of the external criterion NCSRC and circular transformation on the performance of the C-CGMDH model. Second, we analyzed the convergence speed and feature selection performance of the C-CGMDH, PE-CGMDH, and RGMDH models. Next, we compared the classification performance of the C-CGMDH model with that of existing models, including four CVNNs: PE-CGMDH [67], fully complex-valued relaxation network (FCRN) [39], fully complex-valued fast learning classifier (FC-FLC): AF₁ and AF₂ [41]; and three real-valued classification models: RGMDH [54], SVM, and real-valued multilayer perceptron (RMLP). Then, we analyzed the time complexity of the above eight models. Furthermore, we analyzed the explanation performance of three GMDH-type neural networks on a data set.

A. Data Sets and Experimental Setup

We used 25 classification data sets from the UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/datasets.ph-p>). Table I summarizes the characteristics of the data sets used in this study. Among them, there are 17 binary classification problems and eight multiple-category ones.

Because the class distributions of most data sets are imbalanced, we defined the imbalanced factor (IF) [73] as follows: $IF = \max_{j=1,2,\dots,M} N_j / \min_{j=1,2,\dots,M} N_j$, where M is the number of classes in the data set and N_j is the total number

of samples belonging to class j . Then, $IF \geq 1$, and a larger value of IF means a worse imbalance of the class distribution. If $IF = 1$, the class distribution is balanced. The last column of Table I is the IF of each data set. Data sets “Seeds” and “Iris” are balanced, whereas data set “Zoo” has the worst imbalance (its IF value is 10.25).

For the parameters in the FCRN, FC-FLC(AF₁), FC-FLC(AF₂), SVM, and RMLP models, we let their values reflect the best performance from the repeated experiments. The external criteria of the RGMDH, PE-CGMDH, and C-CGMDH models were SRC, CSRC, and NCSRC, respectively. Meanwhile, the eight models did not consider the impact of class imbalance on performance, except for the C-CGMDH and PE-CGMDH models. To ensure fairness of comparison, we balanced the class distribution of the training data set using the random oversampling technique before training the six classification models. Furthermore, we adopted fivefold cross-validation in our experiments to evaluate the classification performance of different models. All experiments were performed on an MATLAB 2016a platform with a dual-processor, 2.40-GHz Core i7 Windows 10 PC. In each case, the result was the average of ten experiments. We implemented the SVM with the radial basis function kernel function and the RMLP with two hidden layers. Finally, regarding the C-CGMDH and RGMDH models, a one-versus-one approach was used to deal with multiple-category classification problems, as in the SVM model.

To evaluate the classification performance of each model, we used the following performance measures.

- 1) Total accuracy (T_{ac})

$$T_{ac} = u_c/u \times 100\% \quad (20)$$

where u is the number of total samples in the testing set and u_c is the number of correctly classified samples.

- 2) Average accuracy (A_{ac})

$$A_{ac} = 1/M \sum_j^M u_{cj}/u_j \times 100\% \quad (21)$$

where u_j is the number of samples with class label j in the testing set and u_{cj} is the number of correctly classified samples with class label j .

B. Influence of the External Criterion NCSRC and Circular Transformation on C-CGMDH Performance

To analyze the influence of the external criterion NCSRC and circular transformation on the performance of the C-CGMDH model, we did the ablation study comparing it with the following three models: 1) PE-CGMDH; 2) PE-CGMDH₁, which is similar to the PE-CGMDH model, but uses the circular transformation; and 3) PE-CGMDH₂, which is also similar to the PE-CGMDH model, but uses the NCSRC. Table II shows the T_{ac} of the four GMDH-type models on 25 data sets, and the last row shows the average T_{ac} . For each data set, the boldface marks the highest T_{ac} .

To analyze whether the difference in T_{ac} of the above four models is statistically significant, we used the nonparametric Wilcoxon signed-rank test [76] to perform the pairwise test. The null hypothesis is that the two compared classification models have equivalent performance. In this study, we let R^+ be the sum of ranks for the data sets, in which the former

TABLE II
COMPARISON OF T_{ac} FOR FOUR GMDH-TYPE
MODELS ON 25 DATA SETS (%)

Datasets	C-CGMDH	PE-CGMDH	PE-CGMDH ₁	PE-CGMDH ₂
Adult	79.19	77.64	77.35	70.96
Australia-c	85.51	85.51	85.51	85.51
Bank-mark	84.93	84.53	84.53	80.52
Breast-c	97.86	96.71	96.71	88.84
Bupa	63.48	62.03	62.61	61.16
Connectionist	75.96	72.81	74.05	69.73
Contraceptive	53.63	52.10	52.27	22.61
Defaultcredit	78.03	76.44	73.99	63.20
Haberman-s	74.45	74.85	74.52	72.24
Hayes-roth	65.16	64.50	65.24	42.28
Hepatitis	74.67	72.75	74.02	70.77
HTRU2	97.55	97.20	97.45	85.24
Ionosphere	91.44	91.17	90.03	76.39
Iris	97.33	97.30	97.33	84.00
MAGIC	79.75	78.97	79.00	75.19
Monk3	96.39	92.59	94.95	82.30
Pima	78.25	77.22	77.21	76.30
QSAR	81.90	77.25	76.68	71.94
Qualitative-b	100.00	100.00	100.00	98.40
Seeds	95.71	95.00	94.76	56.67
Teaching-a	58.28	54.21	55.66	34.45
Vehicle	62.65	61.01	61.59	37.24
Vote	97.62	95.62	95.62	95.62
Wine	99.43	98.10	97.78	73.62
Zoo	95.00	94.01	94.04	81.24
Average	82.57	81.18	81.32	70.26

TABLE III
RESULTS OF WILCOXON TEST FOR COMPARING THE T_{ac} OF FOUR MODELS

Comparison	Min(R^+ , R^-)	CV	Hypothesis
C-CGMDH vs PE-CGMDH	Min(271.5, 4.5)=4.5	89	Reject
C-CGMDH vs PE-CGMDH ₁	Min(250, 3)=3	89	Reject
C-CGMDH vs PE-CGMDH ₂	Min(300, 0)=0	89	Reject
PE-CGMDH vs PE-CGMDH ₁	Min(80, 130)=80	89	Reject
PE-CGMDH vs PE-CGMDH ₂	Min(276, 0)=0	89	Reject
PE-CGMDH ₁ vs PE-CGMDH ₂	Min(276, 0)=0	89	Reject

algorithm outperformed the latter, R^- the sum of ranks for the opposite case, and the significance level $\alpha = 0.05$ in all cases. Then, when the number of data sets is 25, the corresponding critical value (CV) is 89. Table III shows the test results. In each row, if $T = \min(R^+, R^-)$ is less than or equal to the corresponding CV, we reject the null hypothesis, that is, there are significant differences between the two models. Furthermore, if $T = R^- \leq 89$, it means that the former model significantly outperforms the latter, and $T = R^+ \leq 89$, for the opposite case. From Table III, we can obtain the following conclusions.

- 1) C-CGMDH significantly outperforms the PE-CGMDH₁, PE-CGMDH₂, and PE-CGMDH models, which indicates that the C-CGMDH model that simultaneously considers the circular transformation and NCSRC achieves the best performance, and the model proposed in this study is reasonable and effective.

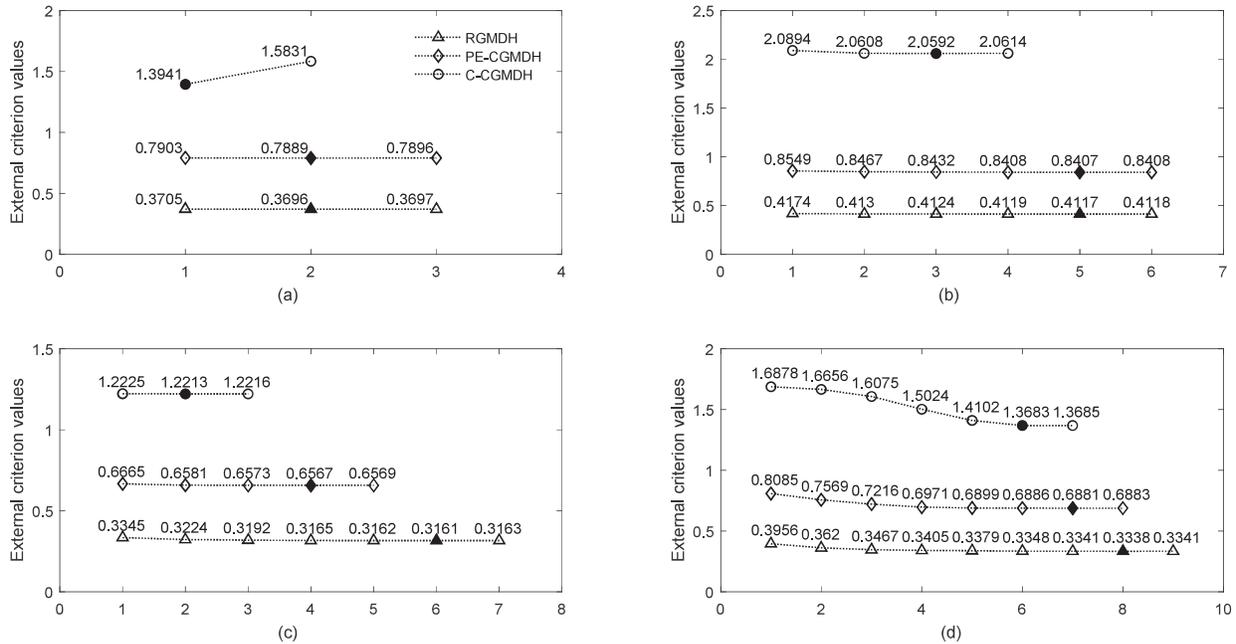


Fig. 3. Learning curves of the C-CGMDH, PE-CGMDH, and RGMDH models in four data sets. The curves with triangle, diamond, and circle correspond to the RGMDH, PE-CGMDH, and C-CGMDH models, respectively. (a) Layers in data set Monk3. (b) Layers in data set MAGIC. (c) Layers in data set Australia-C. (d) Layers in data set Ionosphere.

2) PE-CGMDH₁ significantly outperforms PE-CGMDH, and the performance of the PE-CGMDH₂ model is significantly worse than that of the PE-CGMDH model. This indicates that the circular transformation can significantly improve the performance of the PE-CGMDH model. The reason may be that it can use all the four quadrants of the complex plane effectively. Furthermore, the NCSRC degrades the performance of the PE-CGMDH model. The reason may be that it does not conform to the class label prediction method in the PE-CGMDH model, especially in the multiple-category classification problems.

We can also obtain similar conclusions for A_{ac} .

C. Convergence Speed of C-CGMDH, PE-CGMDH, and RGMDH Models

Because the convergence speed of the GMDH-type neural networks is usually related to the number of features, we selected data sets with fewer features (such as “Monk3” and “MAGIC”) and more features (such as “Australia-c” and “Ionosphere”) to comprehensively analyze the convergence speed of the C-CGMDH, PE-CGMDH, and RGMDH models. In addition, to facilitate the comparison of experimental results, we performed the same process as in (16) for the external criteria of the PE-CGMDH and RGMDH models.

The learning curves of the three models in one experiment on the four selected data sets are shown in Fig. 3. Taking “Monk3” as an example [see Fig. 3(a)], the vertical axis denotes the external criterion values, and the horizontal axis denotes the number of layers of the GMDH-type neural networks. According to its modeling mechanism, the maximum number of layers is the number of features of the data set. Hence, the possible maximum number of layers is 6 for “Monk3.” The external criterion values for three

models are of no practical significance, because they are different. We are more interested when the external criterion value is the minimum, which corresponds to the optimal complexity model. From Fig. 3(a), the optimal complexity model of the RGMDH and PE-CGMDH models is found in the second layer. Meanwhile, that of C-CGMDH is found in the first layer, meaning that the external criterion value of the C-CGMDH does not have a decreasing phase on this data set. It is a special case of the theory of optimal phase. Similar conclusions can be reached for three other data sets. In short, the convergence speed of the C-CGMDH model is faster than that of the RGMDH and PE-CGMDH models.

Considering that there may be deviations in the conclusions based on only one experiment, we repeated our experiment ten times and found interesting results. In data set “Australia-c,” the three models can be ranked according to the convergence speed, from fast to slow: C-CGMDH, PE-CGMDH, and RGMDH. The optimal complexity model of C-CGMDH was mostly found in the first or second layer. Otherwise, it was approximately 0.9 of the probability that the convergence speed of the C-CGMDH model was the fastest. Therefore, according to our experiments, the convergence speed of the C-CGMDH model proposed in this study was the fastest.

D. Feature Selection Performance of C-CGMDH, PE-CGMDH, and RGMDH Models

Feature selection was related to the convergence speed: a faster convergence speed typically meant fewer selected features. In Section IV-C, we compared the convergence speed of different GMDH-type neural networks. In this section, we further analyze the feature selection performance of the three models.

We experimented on the four data sets from Section IV-C. On each data set for each model, we performed feature selection ten times and counted the number of selected features

TABLE IV

NUMBER OF SELECTED FEATURES ON THE DATA SET AUSTRALIA-C

Models	1	2	3	4	5	6	7	8	9	10	Mode
RGMDH	10	6	8	9	10	9	9	9	10	8	9
PE-CGMDH	6	5	7	7	6	6	7	3	6	6	6
C-CGMDH	3	3	3	2	2	3	4	2	4	3	3

TABLE V

MODE OF THE NUMBER OF SELECTED FEATURES ON FOUR DATA SETS

Datasets	C-CGMDH	PE-CGMDH	RGMDH
Monk3	2	3	3
MAGIC	6	9	9
Australia-c	3	6	9
Ionosphere	13	13	14

each time. For the ‘‘Australia-c’’ data set, the results are shown in Table IV. It would be inconvenient to compare directly the results of ten selections for each model. Thus, in the last column of Table IV, we give the mode of the ten experiments for each model. In most of the cases, the RGMDH model selected 9 out of 15 features from this data set. The PE-CGMDH model selected six, whereas the C-CGMDH (the proposed) model only selected three.

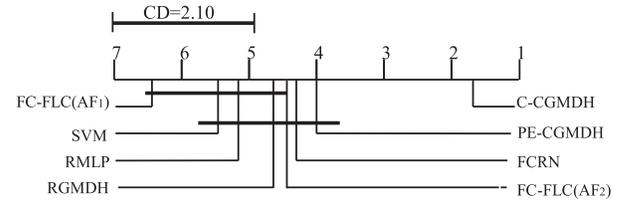
Table V shows the mode of the number of features selected by each GMDH-type neural network in ten experiments on our selected data sets. On the four data sets, the number of features selected by the C-CGMDH model was typically smaller than that of the PR-CGMDH and RGMDH models.

E. Comparison of Classification Performance

Table VI reports T_{ac} of eight classification models on 25 data sets, and the average T_{ac} value of each model is in the last row. From this table, it can be observed that the average T_{ac} value of the C-CGMDH model is the largest. In ‘‘Monk3,’’ compared with the RGMDH model, the improvement of the PE-CGMDH model is approximately 12.45%, whereas that of the C-CGMDH model is approximately 16.25%.

To find significant differences among the results obtained by the eight models, a statistical analysis is necessary. Thus, to check whether there are significant differences in the classification performance of the eight models, we employed the Friedman test [77], [78] and the Iman–Davenport test [79]. If there are statistically significant differences in the performance, we can proceed with the Nemenyi test [80] *post hoc*.

As shown in Table VI, we first conducted the Friedman test to establish the statistical significance of the C-CGMDH model. The rank of performance of different models was calculated for each data set. For every model, we ordered the accuracy from the largest to the smallest with ranks, 1, 2, . . . , 8. If the accuracy of two or more models was the same, we selected their average values as their rank values. For example, in the second data set, ‘‘Australia-c,’’ T_{ac} of the five models was 85.51, the largest. Thus, their rank values are the same, $(1 + 2 + 3 + 4 + 5)/5 = 3$. Similarly, the average rank of T_{ac} of the eight models on 25 data sets is shown in the last row of Table VI. A smaller average rank means a better model performance. The null hypothesis states that all models are equivalent, and thus, their ranks must be equal. The Friedman

Fig. 4. Nemenyi *post hoc* test results for eight models on T_{ac} .

statistic is 74.54. A better statistic, derived by Iman and Davenport, following the F-distribution, is 17.81. The modified statistic follows the F-distribution with 7 and 168 degrees of freedom; the CV for rejecting the null hypothesis is 2.06. Because the modified Friedman statistic is greater than the CV (17.81 > 2.06), we can reject the null hypothesis. Thus, it can be inferred that the models used in this study are not equivalent.

After the null hypothesis was rejected, we applied the Nemenyi test. This test assumes that the performance of the two models is considered significantly different if the corresponding average ranks differ by at least the critical difference. When the number of models is equal to 8 and the significance level is 0.05, the CV is 3.03 and the critical distance (CD) is $3.03(8 * 9 / (6 * 25))^{1/2} = 2.10$. Fig. 4 shows the test results, where the models connected by a line segment are not significantly different. Thus, from Fig. 4, we can identify that T_{ac} of the C-CGMDH model is statistically significantly better than the others.

Similarly, the average accuracy among the classes (A_{ac}) and their ranks of eight models on 25 data sets are summarized in Table VII. The last row shows the average A_{ac} value and the average rank of different models. The average A_{ac} value and the average rank of the C-CGMDH model remain the best. Furthermore, we can also infer that the performance of the C-CGMDH model is statistically significantly better than the others in A_{ac} , using the Friedman test and the Nemenyi *post hoc* test.

F. Comparison of Time Complexity

To compare the time complexity of the eight models, we compare the computation time in the same conditions.

Table VIII presents the average computation time for each model on every data set. As in Tables VI and VII, the ranks are marked with superscripts. Furthermore, considering the modeling characteristics of the GMDH-type neural networks, Table VIII is divided into three parts. The last row of each part shows the average computation time and the average rank of different models. From Table VIII, we obtain the following conclusions.

- 1) When the number of features in the data set is greater than ten (displayed in the first part of Table VIII), the eight models can be ranked according to their average rank, from low to high: FC-FLC(AF₁), SVM, FCRN, RMLP, FC-FLC(AF₂), RGMDH, PE-CGMDH, and C-CGMDH. The computation time of three GMDH-type neural networks is not ideal, perhaps owing to their modeling mechanism. When modeling with the GMDH-type neural networks, the new candidate model in each layer is a combination of any two models of the previous layer. Thus, its time complexity is high if the classification problem has many features.

TABLE VI
COMPARISON OF T_{ac} FOR EIGHT CLASSIFICATION MODELS ON 25 DATA SETS (%)

Datasets	C-CGMDH	PE-CGMDH	RGMDH	SVM	FCRN	FC-FLC(AF ₁)	FC-FLC(AF ₂)	RMLP
Adult	79.19 ⁽⁶⁾	77.64 ⁽⁷⁾	77.38 ⁽⁸⁾	84.06⁽¹⁾	83.14 ⁽³⁾	82.92 ⁽⁵⁾	83.26 ⁽²⁾	83.01 ⁽⁴⁾
Australia-c	85.51⁽³⁾	85.51⁽³⁾	85.51⁽³⁾	85.50 ^(6.5)	85.50 ^(6.5)	85.47 ⁽⁸⁾	85.51⁽³⁾	85.51⁽³⁾
Bank-mark	84.93 ⁽⁶⁾	84.53 ⁽⁷⁾	83.94 ⁽⁸⁾	88.25⁽¹⁾	87.35 ⁽²⁾	85.75 ⁽⁵⁾	86.45 ⁽⁴⁾	87.31 ⁽³⁾
Breast-c	97.86⁽¹⁾	96.71 ^(4.5)	96.71 ^(4.5)	96.70 ^(6.5)	96.70 ^(6.5)	96.81 ⁽³⁾	97.03 ⁽²⁾	96.57 ⁽⁸⁾
Bupa	63.48⁽¹⁾	62.03 ^(4.5)	62.03 ^(4.5)	61.70 ⁽⁷⁾	62.21 ⁽³⁾	61.91 ⁽⁶⁾	62.23 ⁽²⁾	60.58 ⁽⁸⁾
Connectionist	75.96⁽¹⁾	72.81 ⁽²⁾	71.61 ⁽³⁾	70.60 ⁽⁶⁾	71.18 ⁽⁴⁾	70.02 ⁽⁸⁾	71.11 ⁽⁵⁾	70.27 ⁽⁷⁾
Contraceptive	53.63⁽⁴⁾	52.10 ⁽⁶⁾	50.24 ⁽⁸⁾	51.30 ⁽⁷⁾	52.27 ⁽⁵⁾	53.19 ⁽²⁾	53.01 ⁽³⁾	52.75 ⁽⁴⁾
Defaultcredit	78.03⁽⁴⁾	76.44 ⁽³⁾	75.11 ⁽⁷⁾	77.72 ⁽²⁾	76.13 ⁽⁴⁾	74.42 ⁽⁸⁾	75.62 ⁽⁶⁾	76.01 ⁽⁵⁾
Haberman-s	74.45 ⁽²⁾	74.85⁽¹⁾	74.20 ⁽³⁾	73.90 ⁽⁶⁾	74.00 ⁽⁴⁾	72.78 ⁽⁸⁾	73.95 ⁽⁵⁾	72.87 ⁽⁷⁾
Hayes-roth	65.16 ⁽²⁾	64.50 ^(3.5)	64.47 ⁽⁵⁾	62.90 ⁽⁸⁾	64.50 ^(3.5)	63.11 ⁽⁷⁾	64.37 ⁽⁶⁾	72.74⁽¹⁾
Hepatitis	74.67⁽¹⁾	72.75 ⁽²⁾	72.73 ⁽³⁾	70.80 ⁽⁷⁾	71.11 ⁽⁵⁾	70.51 ⁽⁸⁾	72.13 ⁽⁴⁾	70.82 ⁽⁶⁾
HTRU2	97.55 ⁽³⁾	97.20 ⁽⁴⁾	97.61 ⁽²⁾	97.90⁽¹⁾	97.19 ⁽⁵⁾	96.51 ⁽⁷⁾	97.03 ⁽⁶⁾	89.38 ⁽⁸⁾
Ionosphere	91.44⁽¹⁾	91.17 ⁽³⁾	87.46 ⁽⁷⁾	87.50 ⁽⁶⁾	89.15 ⁽⁵⁾	86.72 ⁽⁸⁾	90.27 ⁽⁴⁾	91.18 ⁽²⁾
Iris	97.33⁽¹⁾	97.30 ⁽²⁾	96.00 ⁽⁶⁾	95.33 ⁽⁷⁾	96.19 ⁽⁴⁾	96.10 ⁽⁵⁾	95.22 ⁽⁸⁾	96.67 ⁽³⁾
MAGIC	79.75⁽¹⁾	78.97 ⁽⁵⁾	79.14 ⁽³⁾	79.10 ⁽⁴⁾	79.41 ⁽²⁾	78.37 ⁽⁷⁾	78.91 ⁽⁶⁾	71.80 ⁽⁸⁾
Monk3	96.39⁽¹⁾	92.59 ⁽²⁾	80.14 ⁽⁷⁾	80.10 ⁽⁸⁾	89.71 ⁽³⁾	85.44 ⁽⁶⁾	88.98 ⁽⁴⁾	87.73 ⁽⁵⁾
Pima	78.25⁽¹⁾	77.22 ⁽⁵⁾	76.83 ⁽⁶⁾	76.60 ⁽⁷⁾	78.05 ⁽³⁾	77.97 ⁽⁴⁾	78.21 ⁽²⁾	64.45 ⁽⁸⁾
QSAR	81.90⁽¹⁾	77.25 ⁽⁶⁾	80.28 ⁽³⁾	80.74 ⁽²⁾	78.25 ⁽⁵⁾	74.21 ⁽⁸⁾	76.29 ⁽⁷⁾	79.27 ⁽⁴⁾
Qualitative-b	100.00^(1.5)	100.00^(1.5)	99.60 ⁽⁵⁾	99.60 ⁽⁵⁾	99.60 ⁽⁵⁾	98.50 ⁽⁸⁾	99.60 ⁽⁵⁾	99.60 ⁽⁵⁾
Seeds	95.71^(1.5)	95.00 ⁽⁶⁾	95.71^(1.5)	94.80 ⁽⁷⁾	95.52 ^(3.5)	94.52 ⁽⁸⁾	95.52 ^(3.5)	95.24 ⁽⁵⁾
Teaching-a	58.28⁽¹⁾	54.21 ⁽⁵⁾	55.63 ⁽²⁾	52.30 ⁽⁷⁾	53.18 ⁽⁶⁾	54.29 ⁽⁴⁾	54.41 ⁽³⁾	50.39 ⁽⁸⁾
Vehicle	62.65⁽¹⁾	61.01 ⁽⁷⁾	61.35 ⁽⁶⁾	60.80 ⁽⁸⁾	61.70 ^(3.5)	61.58 ⁽⁵⁾	61.70 ^(3.5)	62.41 ⁽²⁾
Vote	97.62⁽¹⁾	95.62 ^(4.5)	95.62 ^(4.5)	95.60 ⁽⁷⁾	95.62 ^(4.5)	95.47 ⁽⁸⁾	95.62 ^(4.5)	95.65 ⁽²⁾
Wine	99.43⁽¹⁾	98.10 ⁽²⁾	97.75 ⁽⁴⁾	97.80 ⁽³⁾	96.61 ⁽⁵⁾	95.91 ⁽⁷⁾	96.57 ⁽⁶⁾	91.05 ⁽⁸⁾
Zoo	95.00⁽¹⁾	94.01 ^(3.5)	94.05 ⁽²⁾	93.10 ^(6.5)	94.01 ^(3.5)	93.07 ⁽⁸⁾	93.10 ^(6.5)	93.19 ⁽⁵⁾
Average	82.57^(1.68)	81.18 ⁽⁴⁾	80.44 ^(4.64)	80.58 ^(5.46)	81.13 ^(4.3)	80.22 ^(6.44)	81.04 ^(4.44)	79.77 ^(5.16)

Note: The boldface in this table marks the highest T_{ac} in each dataset. The superscript denotes the rank.

- When the number of features in the data set is between 7 and 10 (displayed in the second part of Table VIII), the eight models can be ranked according to the average rank, from low to high: FCRN, FC-FLC(AF₁), PE-CGMDH, RGMDH, SVM, RMLP, C-CGMDH, and FC-FLC(AF₂).
- When the number of features in the data set is fewer than seven (displayed in the third part of Table VIII), the eight models can be ranked according to the average rank, from low to high: FCRN, RGMDH, PE-CGMDH, C-CGMDH, FC-FLC(AF₁), SVM, FC-FLC(AF₂), and RMLP. Three GMDH-type neural networks are only poorer than the FCRN model.

Regarding time complexity, the C-CGMDH model is comparable with others when dealing with the data sets having few features.

G. Explanation Performance

In this section, we present the final models of the three GMDH-type neural networks on the same data set to show their explanatory power.

For convenience, we take the data set ‘‘Monk3’’ as an example. Its IF is 1.08, which does not require balancing of the training set and, thus, excludes the impact of randomness of resampling on the experimental results. In addition, there is a substantial difference among the three models on this data set, as shown in Tables VI and VII. This data set contains

554 samples and two classes. For reproducibility of results, the data set was divided in its original order. We first selected the former 222 samples from each class to constitute the training set, and the remaining 110 samples constituted the testing set. Furthermore, the training set was equally divided into a model learning set and a model selecting set. Finally, the modeling results of the three GMDH-type neural networks are shown as follows:

- The final form of the RGMDH model

$$y = 0.2547 + 0.3350x_2 + 0.0427x_4 + 0.1889x_5$$

$$T_{ac} = 76.36\%, \quad A_{ac} = 75.00\%.$$

- The final form of the PE-CGMDH model

$$y = (-0.1478 - 0.3474i) + (0.4804 - 0.1018i)x_2$$

$$+ (0.0575 + 0.0007i)x_4 + (0.4697 - 0.2146i)x_5$$

$$T_{ac} = 87.27\%, \quad A_{ac} = 88.64\%.$$

- The final form of the C-CGMDH model

$$\text{Parameters: } a = 0.8979, \quad b = 0.6512, \quad \alpha = (1.5460, 1.4704, 3.4459, 2.4682, 4.7425, 2.4666)^T.$$

$$y = (-5.1766 + 3.9064i) + (3.4467 - 1.7800i)x_2 + (-2.8244 + 0.9457i)x_5.$$

$$T_{ac} = 92.73\%, \quad A_{ac} = 92.05\%.$$

The explanatory power of three GMDH-type neural networks is strong.

TABLE VIII
COMPARISON OF COMPUTATION TIME FOR EIGHT CLASSIFICATION MODELS ON 25 DATA SETS (UNIT: SECONDS)

Datasets	Features	Instances	C-CGMDH	PE-CGMDH	RGMDH	SVM	FCRN	FC-FLC(AF_1)	FC-FLC(AF_2)	RMLP
Part 1										
Adult	14	48842	438.78 ⁽⁴⁾	871.49 ⁽⁷⁾	545.07 ⁽⁵⁾	639.85 ⁽⁶⁾	57.69 ⁽³⁾	34.36⁽¹⁾	5313.43 ⁽⁸⁾	40.77 ⁽²⁾
Australia-c	15	690	6.58 ⁽⁷⁾	9.50 ⁽⁸⁾	5.75 ⁽⁶⁾	0.53⁽¹⁾	0.80 ⁽²⁾	1.19 ⁽³⁾	5.08 ⁽⁵⁾	4.43 ⁽⁴⁾
Bank-mark	20	41188	1057.40 ⁽⁵⁾	1228.20 ⁽⁶⁾	935.01 ⁽⁴⁾	5515.10 ⁽⁸⁾	77.83 ⁽²⁾	34.97⁽¹⁾	5123.11 ⁽⁷⁾	90.56 ⁽³⁾
Connectionist	60	208	1297.10 ⁽⁸⁾	828.99 ⁽⁷⁾	709.77 ⁽⁶⁾	0.02⁽¹⁾	2.59 ⁽⁴⁾	0.75 ⁽²⁾	0.97 ⁽³⁾	6.07 ⁽⁵⁾
Defaultcredit	23	29999	1309.20 ⁽⁶⁾	1507.60 ⁽⁷⁾	841.34 ⁽⁵⁾	570.25 ⁽⁴⁾	67.94 ⁽³⁾	27.97⁽¹⁾	4979.13 ⁽⁸⁾	37.35 ⁽²⁾
Hepatitis	19	155	3.47 ⁽⁸⁾	3.56 ⁽⁷⁾	2.95 ⁽⁶⁾	0.32⁽¹⁾	1.08 ⁽⁴⁾	0.47 ⁽²⁾	0.72 ⁽³⁾	2.72 ⁽⁵⁾
Ionosphere	34	351	106.45 ⁽⁸⁾	70.42 ⁽⁷⁾	66.17 ⁽⁶⁾	0.38⁽¹⁾	4.80 ⁽⁵⁾	1.66 ⁽²⁾	3.58 ⁽³⁾	3.98 ⁽⁴⁾
QSAR	41	1055	306.71 ⁽⁸⁾	229.18 ⁽⁷⁾	193.63 ⁽⁶⁾	0.08⁽¹⁾	6.48 ⁽⁴⁾	1.55 ⁽²⁾	9.69 ⁽⁵⁾	2.13 ⁽³⁾
Vehicle	18	846	47.39 ⁽⁸⁾	9.98 ⁽⁵⁾	43.00 ⁽⁷⁾	1.02⁽¹⁾	5.16 ⁽³⁾	2.03 ⁽²⁾	7.91 ⁽⁴⁾	10.02 ⁽⁶⁾
Vote	16	435	5.38 ⁽⁷⁾	7.38 ⁽⁸⁾	3.59 ⁽⁶⁾	0.35⁽¹⁾	2.24 ⁽⁴⁾	1.14 ⁽²⁾	2.13 ⁽³⁾	2.89 ⁽⁵⁾
Wine	13	178	3.09 ⁽⁷⁾	0.70 ⁽⁴⁾	3.50 ⁽⁸⁾	0.61 ⁽³⁾	0.22⁽¹⁾	0.44 ⁽²⁾	0.75 ⁽⁵⁾	2.50 ⁽⁶⁾
Zoo	16	101	35.83 ⁽⁸⁾	1.80 ⁽⁴⁾	27.75 ⁽⁷⁾	2.01 ⁽⁵⁾	0.75 ⁽³⁾	0.45^(1.5)	0.45^(1.5)	3.95 ⁽⁶⁾
Ave. of Part 1	—	—	384.78 ⁽⁷⁾	397.4 ^(6.42)	281.46 ⁽⁶⁾	560.88 ^(2.75)	18.97 ^(3.17)	8.92^(1.88)	1287.25 ^(4.63)	17.28 ^(4.33)
Part 2										
Breast-c	9	683	2.27 ⁽⁶⁾	2.14 ⁽⁵⁾	1.25 ⁽⁴⁾	0.36⁽¹⁾	0.47 ⁽²⁾	0.89 ⁽³⁾	3.94 ⁽⁸⁾	2.89 ⁽⁷⁾
Contraceptive	9	1473	8.16 ⁽⁶⁾	1.53 ⁽²⁾	5.53 ⁽⁵⁾	1.64 ⁽³⁾	0.98⁽¹⁾	2.09 ⁽⁴⁾	44.63 ⁽⁸⁾	8.80 ⁽⁷⁾
HTRU2	8	17898	24.08 ⁽⁶⁾	21.03 ⁽⁵⁾	15.56 ⁽³⁾	47.56 ⁽⁷⁾	9.30⁽¹⁾	11.39 ⁽²⁾	581.53 ⁽⁸⁾	16.63 ⁽⁷⁾
MAGIC	10	19020	43.66 ⁽⁶⁾	35.13 ⁽⁵⁾	23.03 ⁽⁴⁾	207.22 ⁽⁷⁾	12.69 ⁽²⁾	13.56 ⁽³⁾	617.15 ⁽⁸⁾	11.33⁽¹⁾
Pima	8	768	2.16 ⁽⁶⁾	2.11 ⁽⁵⁾	1.30 ⁽⁴⁾	0.52⁽¹⁾	0.53 ⁽²⁾	0.91 ⁽³⁾	4.73 ⁽⁸⁾	3.55 ⁽⁷⁾
Seeds	7	210	0.52 ⁽⁶⁾	0.16^(1.5)	0.19 ⁽³⁾	0.60 ⁽⁷⁾	0.16^(1.5)	0.28 ⁽⁴⁾	0.48 ⁽⁵⁾	3.72 ⁽⁸⁾
Ave. of Part 2	—	—	13.48 ⁽⁶⁾	10.35 ^(3.92)	7.81 ^(3.83)	42.98 ^(4.33)	4.02^(1.58)	4.85 ^(3.17)	208.74 ^(7.5)	7.82 ^(5.67)
Part 3										
Bupa	6	345	0.44 ⁽⁵⁾	0.45 ⁽⁶⁾	0.16⁽¹⁾	0.33 ⁽⁴⁾	0.19 ⁽²⁾	0.31 ⁽³⁾	1.06 ⁽⁷⁾	2.00 ⁽⁸⁾
Haberman-s	3	306	0.17 ⁽⁴⁾	0.16 ⁽³⁾	0.06⁽¹⁾	0.53 ⁽⁶⁾	0.11 ⁽²⁾	0.31 ⁽⁵⁾	0.63 ⁽⁷⁾	2.22 ⁽⁸⁾
Hayes-roth	4	132	0.17 ⁽⁴⁾	0.13 ⁽²⁾	0.19 ⁽⁵⁾	0.58 ⁽⁷⁾	0.08⁽¹⁾	0.14 ⁽³⁾	0.27 ⁽⁶⁾	3.22 ⁽⁸⁾
Iris	4	150	0.14 ⁽³⁾	0.11 ⁽²⁾	0.17 ^(4.5)	4.03 ⁽⁷⁾	0.08⁽¹⁾	0.17 ^(4.5)	0.27 ⁽⁶⁾	2.39 ⁽⁸⁾
Monk3	6	554	0.56 ⁽⁶⁾	0.52 ⁽⁴⁾	0.17⁽¹⁾	0.33 ⁽³⁾	0.25 ⁽²⁾	0.53 ⁽⁵⁾	2.50 ⁽⁸⁾	2.03 ⁽⁷⁾
Qualitative-b	6	250	0.14⁽¹⁾	0.22 ⁽⁴⁾	0.20 ⁽³⁾	0.31 ⁽⁶⁾	0.16 ⁽²⁾	0.28 ⁽⁵⁾	0.53 ⁽⁷⁾	1.95 ⁽⁸⁾
Teaching-a	5	151	0.23 ⁽⁴⁾	0.11^(1.5)	0.27 ^(5.5)	0.68 ⁽⁷⁾	0.11^(1.5)	0.22 ⁽³⁾	0.27 ^(5.5)	6.63 ⁽⁸⁾
Ave. of Part 3	—	—	0.26 ^(3.86)	0.24 ^(3.21)	0.17 ⁽³⁾	0.97 ^(5.71)	0.14^(1.64)	0.28 ^(4.07)	0.79 ^(6.64)	2.95 ^(7.86)

Note: The superscript denotes the rank.

Thus, the time complexity may be high, although it is suitable for classification problems with relatively low dimensions.

Finally, compared with the RGMDH model, the C-CGMDH model has the following differences.

- 1) This study extends the GMDH method from real field to complex field, and it constructs a complex-valued classification model C-CGMDH for real-valued classification problems.
- 2) To improve the classification performance of the C-CGMDH model, a new complex-valued external criterion NCSRC is proposed and proved to satisfy the theory of optimal complexity, which indicates its effectiveness and rationality.
- 3) Its convergence speed is faster.
- 4) It can select fewer important features.
- 5) Its classification performance is significantly better.

V. CONCLUSION

To overcome the shortcomings of existing CVNNs, this study presented a C-CGMDH model. First, we proposed a complex LS for parameter estimation. Next, a new complex-valued external criterion NCSRC was constructed with the logarithmic function to represent explicitly the magnitude and phase of the actual and predicted complex outputs, to evaluate and select the middle candidate models. Its property was

proven to be similar to that of the real external criterion. Finally, before training this model, a circular transformation was used for data preprocessing to transform the real-valued input features to the complex-valued input features. Experiments on 25 UCI data sets showed that our improvement was quite necessary. Among the three GMDH-type neural networks, the C-CGMDH model's convergence speed was the fastest, and the number of features selected by it was the smallest. Furthermore, its classification performance was statistically significantly better than that of the four complex-valued and three real-valued models in terms of both T_{ac} and A_{ac} . The time complexity of the C-CGMDH model was comparable with that of other models when dealing with the data sets that have few features. In addition, it is a white-box model that displays an interpretable expression.

The C-CGMDH model proposed in this study uses the first-order linear transfer function, and its classification performance remains poor on some data sets. For example, on the data set "Teaching-a," as shown in Table VI, its T_{ac} increases by 2.65%, but this value is still less than 60%, compared with the best of the other seven alternative models. In fact, for data sets with a complex data structure, nonlinear transfer functions may achieve better classification performances. Therefore, to improve the classification performance, we should adopt more complex transfer functions on these data sets in the future, such as the second-order and third-order

KG polynomials. In addition, due to the unique modeling mechanism of GMDH-type neural networks, the middle candidate model in the first layer is a combination of any two input features. Thus, the time complexity may be very high for data sets with many features. To address this issue, a possible approach is to calculate the correlation coefficients between any two features and rank them in the ascending order. In theory, combining pairs of features with a lower correlation for modeling may achieve better performance. Therefore, if we select a part of the feature pairs ranking in the top to construct the model, then it is expected to reduce the time complexity without affecting the model's performance. We plan to work on this in the future.

ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewers and the editor for their constructive comments.

REFERENCES

- [1] A. Hirose, *Complex-Valued Neural Networks*. New York, NY, USA: Springer, 2012.
- [2] I. Aizenberg, *Complex-Valued Neural Networks With Multi-Valued Neurons*. Berlin, Germany: Springer, 2011.
- [3] S. Suresh, N. Sundararajan, and R. Savitha, *Supervised Learning With Complex-Valued Neural Networks*. Berlin, Germany: Springer, 2013.
- [4] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Netw.*, vol. 10, no. 8, pp. 1391–1415, Nov. 1997.
- [5] T. Nitta, "Orthogonality of decision boundaries in complex-valued neural networks," *Neural Comput.*, vol. 16, no. 1, pp. 73–97, Jan. 2004.
- [6] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.
- [7] T. Ding and A. Hirose, "Fading channel prediction based on combination of complex-valued neural networks and chirp Z-transform," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1686–1695, Sep. 2014.
- [8] A. R. Hafiz, A. Y. Al-Nuaimi, M. F. Amin, and K. Murase, "Classification of skeletal wireframe representation of hand gesture using complex-valued neural network," *Neural Process. Lett.*, vol. 42, no. 3, pp. 649–664, Dec. 2015.
- [9] R. Ceylan, M. Ceylan, Y. Özbay, and S. Kara, "Fuzzy clustering complex-valued neural network to diagnose cirrhosis disease," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9744–9751, Aug. 2011.
- [10] G. Tanaka and K. Aihara, "Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1463–1473, Sep. 2009.
- [11] K. Mizote, H. Kishikawa, N. Goto, and S.-I. Yanagiya, "Optical label routing processing for BPSK labels using complex-valued neural network," *J. Lightw. Technol.*, vol. 31, no. 12, pp. 1867–1876, Jun. 15, 2013.
- [12] A. Hirose, Y. Asano, and T. Hamano, "Developmental learning with behavioral mode tuning by carrier-frequency modulation in coherent neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1532–1543, Nov. 2006.
- [13] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177–7188, Dec. 2017.
- [14] R. Wu, H. Huang, X. Qian, and T. Huang, "A L-BFGS based learning algorithm for complex-valued feedforward neural networks," *Neural Process. Lett.*, vol. 47, no. 3, pp. 1271–1284, Jun. 2018.
- [15] Y.-M. Shi, L. Huang, C. Qian, and H. C. So, "Shrinkage linear and widely linear complex-valued least mean squares algorithms for adaptive beamforming," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 119–131, Jan. 2015.
- [16] A. Khalili, A. Rastegarnia, and S. Sanei, "Quantized augmented complex least-mean square algorithm: Derivation and performance analysis," *Signal Process.*, vol. 121, pp. 54–59, Apr. 2016.
- [17] R. Rakkiyappan, K. Sivaranjani, and G. Velmurugan, "Passivity and passification of memristor-based complex-valued recurrent neural networks with interval time-varying delays," *Neurocomputing*, vol. 144, pp. 391–407, Nov. 2014.
- [18] S. Zhang, Y. Xia, and J. Wang, "A complex-valued projection neural network for constrained optimization of real functions in complex variables," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3227–3238, Dec. 2015.
- [19] M. Kobayashi, "Symmetric complex-valued Hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 1011–1015, Apr. 2017.
- [20] M. Kobayashi, "Fixed points of symmetric complex-valued Hopfield neural networks," *Neurocomputing*, vol. 275, pp. 132–136, Jan. 2018.
- [21] M. Kobayashi, "Decomposition of rotor Hopfield neural networks using complex numbers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1366–1370, Apr. 2018.
- [22] M. Rajendra and K. Shankar, "Improved complex-valued radial basis function (ICRBF) neural networks on multiple crack identification," *Appl. Soft Comput.*, vol. 28, pp. 285–300, Mar. 2015.
- [23] T. Xiong, Y. Bao, Z. Hu, and R. Chiong, "Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms," *Inf. Sci.*, vol. 305, pp. 77–92, Jun. 2015.
- [24] S. Onojima, Y. Arima, and A. Hirose, "Millimeter-wave security imaging using complex-valued self-organizing map for visualization of moving targets," *Neurocomputing*, vol. 134, pp. 247–253, Jun. 2014.
- [25] P. Zheng, "Threshold complex-valued neural associative memory," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1714–1718, Sep. 2014.
- [26] M. Kobayashi, "Gradient descent learning rule for complex-valued associative memories with large constant terms," *IEEE Trans. Elect. Electron. Eng.*, vol. 11, no. 3, pp. 357–363, May 2016.
- [27] T. Nitta, "Solving the XOR problem and the detection of symmetry using a single complex-valued neuron," *Neural Netw.*, vol. 16, no. 8, pp. 1101–1105, Oct. 2003.
- [28] H. Michel and A. Awwal, "Enhanced artificial neural networks using complex numbers," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jan. 2003, pp. 456–461.
- [29] I. Aizenberg and C. Moraga, "Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm," *Soft Comput.*, vol. 11, no. 2, pp. 169–183, Jan. 2007.
- [30] M. Fajil Amin and K. Murase, "Single-layered complex-valued neural network for real-valued classification problems," *Neurocomputing*, vol. 72, nos. 4–6, pp. 945–955, Jan. 2009.
- [31] M. F. Amin, M. M. Islam, and K. Murase, "Ensemble of single-layered complex-valued neural networks for classification tasks," *Neurocomputing*, vol. 72, nos. 10–12, pp. 2227–2234, Jun. 2009.
- [32] R. Savitha, S. Suresh, N. Sundararajan, and H. J. Kim, "Fast learning fully complex-valued classifiers for real-valued classification problems," in *Advances in Neural Networks (Lecture Notes in Computer Science)*, vol. 6675. Berlin, Germany: Springer, 2011, pp. 602–609.
- [33] S. Suresh, R. Savitha, and N. Sundararajan, "A sequential learning algorithm for complex-valued self-regulating resource allocation network-CRSAN," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1061–1072, Jul. 2011.
- [34] R. Savitha, S. Suresh, N. Sundararajan, and H. Kim, "A fully complex-valued radial basis function classifier for real-valued classification problems," *Neurocomputing*, vol. 78, no. 1, pp. 104–110, Feb. 2012.
- [35] R. Savitha, S. Suresh, and N. Sundararajan, "Fast learning complex-valued classifiers for real-valued classification problems," *J. Mach. Learn. Cybern.*, vol. 4, no. 5, pp. 469–476, Oct. 2013.
- [36] J. Zhang and Y. Wu, "A new method for automatic sleep stage classification," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 5, pp. 1097–1110, Oct. 2017.
- [37] R. Savitha, S. Suresh, and N. Sundararajan, "A fast learning complex-valued neural classifier for real-valued classification problems," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 2243–2249.
- [38] R. Savitha, S. Suresh, and N. Sundararajan, "Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems," *Inf. Sci.*, vol. 187, pp. 277–290, Mar. 2012.
- [39] R. Savitha, S. Suresh, and N. Sundararajan, "Projection-based fast learning fully complex-valued relaxation neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 4, pp. 529–541, Apr. 2013.
- [40] K. Subramanian, R. Savitha, and S. Suresh, "A complex-valued neuro-fuzzy inference system and its learning mechanism," *Neurocomputing*, vol. 123, pp. 110–120, Jan. 2014.
- [41] M. Sivachitra, R. Savitha, S. Suresh, and S. Vijayachitra, "A fully complex-valued fast learning classifier (FC-FLC) for real-valued classification problems," *Neurocomputing*, vol. 149, pp. 198–206, Feb. 2015.

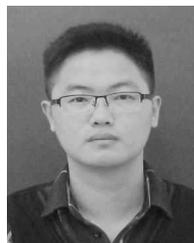
- [42] M. Sivachitra and S. Vijayachitra, "A metacognitive fully complex valued functional link network for solving real valued classification problems," *Appl. Soft Comput.*, vol. 33, pp. 328–336, Aug. 2015.
- [43] A. Da Silva, U. Rodrigues, A. Reis, and L. Moulin, "NeuroDem—A neural network based short term demand forecaster," in *Proc. IEEE Porto Power Tech. Conf.*, Nov. 2002, pp. 1–6.
- [44] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.
- [45] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *CSURACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Aug. 2018.
- [46] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 4s, pp. 1–22, Nov. 2016.
- [47] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Computer Vision (Lecture Notes in Computer Science)*, vol. 9908. Cham, Germany: Springer, 2016, pp. 525–542.
- [48] X. Shu, G.-J. Qi, J. Tang, and J. Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 35–44.
- [49] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.
- [50] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
- [51] A. Ivakhnenko, "Heuristic self-organization in problems of engineering cybernetics," *Automatica*, vol. 6, no. 2, pp. 207–219, Mar. 1970.
- [52] H. R. Madala and A. G. Ivakhnenko, *Inductive Learning Algorithms for Complex Systems Modeling*. Boca Raton, FL, USA: CRC Press, 1994.
- [53] J. A. Mueller and F. Lemke, *Self-Organising Data Mining: An Intelligent Approach to Extract Knowledge from Data*. Hamburg, Germany: Libri, 2000.
- [54] A. G. Ivakhnenko, "The group method of data handling in prediction problems," *Soviet Autom. Control*, vol. 9, no. 6, pp. 21–30, 1976.
- [55] A. P. Sarychev, "An averaged regularity criterion for the group method of data handling in the problem of searching for the best regression," *Sov. J. Automat. Inf. Sci.*, vol. 23, no. 5, pp. 24–29, 1990.
- [56] N. Y. Nikolaev and H. Iba, "Polynomial harmonic GMDH learning networks for time series modeling," *Neural Netw.*, vol. 16, no. 10, pp. 1527–1540, Dec. 2003.
- [57] G. C. Onwubolu, "Hybrid computational intelligence and GMDH systems," in *Hybrid Self-Organizing Modeling Systems (Studies in Computational Intelligence)*, vol. 211. Berlin, Germany: Springer, 2009, pp. 1–26.
- [58] P. Kordík, "GAME—Hybrid self-organizing modeling system based on GMDH," in *Hybrid Self-Organizing Modeling Systems (Studies in Computational Intelligence)*, vol. 211. Berlin, Germany: Springer, 2009, pp. 233–280.
- [59] G. C. Onwubolu, *GMDH-Methodology and Implementation in MATLAB*. London, U.K.: Imperial College Press, 2015.
- [60] J. Xiao, Y. Xiao, A. Huang, D. Liu, and S. Wang, "Feature-selection-based dynamic transfer ensemble model for customer churn prediction," *Knowl. Inf. Syst.*, vol. 43, no. 1, pp. 29–51, Apr. 2015.
- [61] J. Xiao, X. Jiang, C. He, and G. Teng, "Churn prediction in customer relationship management via GMDH-based multiple classifiers ensemble," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 37–44, Mar. 2016.
- [62] J. Xiao, H. Cao, X. Jiang, X. Gu, and L. Xie, "GMDH-based semi-supervised feature selection for customer classification," *Knowl.-Based Syst.*, vol. 132, pp. 236–248, Sep. 2017.
- [63] M. Darvish Damavandi, S. M. Mousavi, and H. Safikhani, "Pareto optimal design of swirl cooling chambers with tangential injection using CFD, GMDH-type of ANN and NSGA-II algorithm," *Int. J. Thermal Sciences*, vol. 122, pp. 102–114, Dec. 2017.
- [64] H. Jafarian, H. Sayyaadi, and F. Torabi, "Modeling and optimization of dew-point evaporative coolers based on a developed GMDH-type neural network," *Energy Convers. Manage.*, vol. 143, pp. 49–65, Jul. 2017.
- [65] L. Mo, L. Xie, X. Jiang, G. Teng, L. Xu, and J. Xiao, "GMDH-based hybrid model for container throughput forecasting: Selective combination forecasting in nonlinear subseries," *Appl. Soft Comput.*, vol. 62, pp. 478–490, Jan. 2018.
- [66] J. Xiao, Y. Li, L. Xie, D. Liu, and J. Huang, "A hybrid model based on selective ensemble for energy consumption forecasting in China," *Energy*, vol. 159, pp. 534–546, Sep. 2018.
- [67] J. Xiao, Y. Hu, and S. Wang, "Complex-valued GMDH-type neural network for real-valued classification problems," in *Proc. 6th Int. Conf. Bus. Intell. Financial Eng.*, Nov. 2013, pp. 70–74.
- [68] A. G. Ivakhnenko and V. Stepashko, *Noise Immunity of Modeling*. Kiev, Ukraine: Naukova Dumka, 1985.
- [69] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory*. Hoboken, NJ, USA: Wiley, 2007.
- [70] D. Serre, *Matrices: Theory and Applications*. New York, NY, USA: Springer, 2010.
- [71] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: JHU Press, 2012.
- [72] B. Zhu, B. Baesens, and S. K. Vanden Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Inf. Sci.*, vol. 408, pp. 84–99, Oct. 2017.
- [73] J. Wu, S. Brubaker, M. Mullin, and J. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 369–382, Mar. 2008.
- [74] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [75] A. I. Marqués, V. García, and J. S. Sánchez, "On the suitability of resampling techniques for the class imbalance problem in credit scoring," *J. Oper. Res. Soc.*, vol. 64, no. 7, pp. 1060–1070, Jul. 2013.
- [76] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, p. 80, Dec. 1945.
- [77] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.
- [78] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, Mar. 1940.
- [79] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the Friedman statistic," *Commun. Statist.-Theory Methods*, vol. 9, no. 6, pp. 571–595, 1980.
- [80] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton Univ., Princeton, NJ, USA, 1963.



Jin Xiao (Member, IEEE) received the Ph.D. degree from the Business School, Sichuan University, Chengdu, China, in 2010.

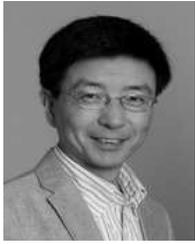
He is currently a Professor with the Business School, Sichuan University, and a Post-Doctoral Research Fellow with the Chinese Academy of Sciences, Beijing, China. He is an Outstanding Contribution Expert of Sichuan Province, Tianfu Ten-thousand Talents Program of Sichuan Province, and the Candidate of Academic and Technological Leaders of Sichuan Province, China. He has authored or coauthored more than 70 articles in leading journals. His research interests include business intelligence, data mining, customer relationship management, and economic forecasting.

Dr. Xiao serves as an anonymous reviewer for more than 50 journals.



Yanlin Jia is currently pursuing the Ph.D. degree in management science with the Business School, Sichuan University, Chengdu, China.

His research interests include business intelligence and data mining.



Xiaoyi Jiang (Senior Member, IEEE) received the bachelor's degree in computer science from Peking University, Beijing, China, in 1983, and the Ph.D. and Venia Docendi (Habilitation) degrees in computer science from the University of Bern, Bern, Switzerland, in 1989 and 1997, respectively.

He was an Associate Professor with the Technical University of Berlin, Berlin, Germany. Since 2002, he has been a Full Professor of computer science with the University of Münster, Münster, Germany, where he is currently the Dean of the Faculty of

Mathematics and Computer Science.

Dr. Jiang is a fellow of the International Association for Pattern Recognition. He also serves on the Advisory Board and Editorial Board of several journals, including the IEEE TRANSACTIONS ON MEDICAL IMAGING and the *International Journal of Neural Systems*. He is also the Editor-in-Chief of the *International Journal of Pattern Recognition and Artificial Intelligence*.



Shouyang Wang (Member, IEEE) received the Ph.D. degree in operations research from the Institute of Systems Science, Chinese Academy of Sciences (CAS), Beijing, China, in 1986.

He is currently a Bairen Distinguished Professor of management science with the Academy of Mathematics and Systems Science, CAS, and a Changjiang Chair Professor of management science and engineering with the University of Chinese Academy of Sciences, Beijing. He has authored or coauthored 30 monographs, of which 15 published by Springer-Verlag, and over 300 articles in leading journals. He is/was a coeditor of 16 journals, including *Information and Management and Energy Economics*. His current research interests include decision analysis, risk management, economic analysis, and forecasting.

Dr. Wang is a fellow of the World Academy of Sciences and the Asia Pacific Industrial Engineering and Management Society and an Academician of the International Academy of Systems and Cybernetics Sciences. He was the President of the International Society of Knowledge and Systems Sciences and is currently the President of the China Society of Systems Engineering. He was a Guest Editor of special issues/volumes of over 15 journals, including *Decision Support Systems*, the *Annals of Operations Research* and the *European Journal of Operational Research*.