

# Interpretable Constructive Algorithm for Random Weight Neural Networks

Jing Nan, Wei Dai, *Senior Member, IEEE*, Guan Yuan, and Ping Zhou, *Senior Member, IEEE*,

**Abstract**—In this paper, an interpretable construction method (IC) with geometric information is proposed to address a significant drawback of incremental random weight neural networks (IRWNNs), which is the difficulty in interpreting the black-box process of hidden parameter selection. The IC utilises geometric relationships to randomly assign hidden parameters, which improves interpretability. In addition, IC employs a node pooling strategy to select the nodes that will both facilitate network convergence. The article also demonstrates the general approximation properties of IC and presents a lightweight version tailored for large-scale data modelling tasks. Experimental results on six benchmark datasets and one numerical simulation dataset demonstrate the superior performance of IC compared to other constructive algorithms in terms of modelling speed, accuracy and network structure. In addition, the effectiveness of IC is validated by two real-world industrial applications.

**Index Terms**—Interpretable constructive, UAP, geometric, data modeling.

## I. INTRODUCTION

NEURAL networks, are different from traditional model-based methods in that they excel at extracting intricate patterns from complex data. Therefore, it is widely used in various research areas [1]–[3]. Among them, Deep Neural Networks (DNNs) and Flat Neural Networks (FNNs) are the most prominent. DNNs achieve end-to-end learning by combining fine-tuning [4], which endows them with superior expressive and generalisation capabilities. However, DNNs are time-consuming to train. Recently, there has been a surge of interest in stochastic weighted neural networks (RWNNs), which are a typical FNN with universal approximation capabilities [5]–[8]. RWNNs are run through a two-step training mode: firstly, randomly assigning the hidden parameters, and then evaluating the output weights through the solution of a system of linear equations.

Despite the noted advantages, the suitability of network structures for RWNNs in modeling tasks remains largely unexplored. A network structure that is excessively large may lead to poor generalization, whereas one that is overly small may result in inadequate learning capacity. Constructive algorithms typically commence with a modest network structure, typically a single hidden node, and progressively expand it by incrementally adding new hidden nodes until the desired performance criteria are met [9]. Consequently, these algorithms tend to provide more conservative network structures for modeling tasks [10]. As a result, the constructive variant of RWNNs, known as incremental RWNNs (IRWNNs), has

been effectively employed in various data modeling endeavors [11]–[13].

In the realm of probability theory, randomly generated hidden parameters may not always be suitable for Improved Random Weight Neural Networks (IRWNNs). This raises the fundamental question: What constitutes suitable hidden parameters for IRWs? Through an algebraic exploration of multidimensional nonlinear functions, [14] demonstrated that the relationship between input samples and hidden parameters can be characterized by nonlinear weight equations. Furthermore, [15] revealed the existence of a supervisory mechanism between hidden parameters and input samples, which enhances network performance. Building upon this, [16] introduced a constructive algorithm with a supervisory mechanism to randomly allocate hidden parameters within a dynamic interval. Additionally, [17] proposed a method for generating hidden parameters by analyzing the scope of input samples and activation functions. More recently, [18] introduced RWNNs with compact incremental inequality constraints, known as CIRW, to refine the quality of hidden parameters. Despite these advancements, the precise mechanisms through which hidden parameters achieve their objectives remain largely unexplored. Consequently, visualizing the influence of each hidden parameter on residual error (network performance) remains a challenging task. Currently, enhancing the interpretability of predicted behaviors of NNs is an increasingly pertinent and significant area of research [ [19], [20]]. Hence, further investigation into interpretable constructive algorithms is crucial and warranted for advancing RWNN.

This paper introduces an interpretable constructive method (IC). The primary contributions are outlined as follows:

- 1) Utilizing the geometric relationship between hidden parameters and residual error, this paper devises an interpretable geometric information constraint to guide the assignment of randomized hidden parameters during the incremental construction process. Furthermore, comprehensive theoretical analysis of this approach is provided.

- 2) A node pool strategy is employed to systematically search for hidden parameters, enhancing the quality of hidden nodes and facilitating convergence in two algorithm implementations, namely IC and IC+, which leverage different calculation methods for network output weights.

The article is organized as follows: Section 2 provides a brief overview of RWNNs and constructive algorithms. In Section 3, we introduce our proposed interpretable constructive algorithm in detail. This algorithm leverages the geometric relationship between hidden parameters and residual error, presenting a novel approach to assigning randomized hid-

den parameters during the incremental construction process. Additionally, a node pool strategy is developed to enhance the quality of hidden nodes, thus facilitating convergence. Section 4 presents the experimental evaluation of the proposed ICA and ICA+ using various datasets, including a numerical simulation dataset, six real-world datasets, an ore grinding semi-physical simulation platform, and a gesture recognition system. Finally, Section 5 concludes the article, summarizing the findings and discussing potential future directions.

## II. PRELIMINARIES

### A. Random Weight Neural Networks

RWNNs are conceptualized as flattened networks, wherein all hidden parameters (comprising input weights and biases) are randomly assigned from a predetermined interval and remain fixed throughout the training process. The evaluation of output weights is accomplished by solving a system of linear equations. The theoretical underpinnings of RWNNs are expounded as follows.

For a target function  $f : R^d \rightarrow R^m$ , the RWNNs with  $L$  hidden nodes can be written as  $f_L = H\beta$ , where  $H = [g_1(\omega_1^T \cdot x + b_1), \dots, g_L(\omega_L^T \cdot x + b_L)]$ ,  $T$  denotes matrix transpose,  $x$  is the input sample,  $\omega_j$  and  $b_j$  are the input weights and biases of the  $j$ -th hidden node, respectively.  $j = 1, \dots, L$ ,  $g_j$  denotes the nonlinear activation function of the  $j$ -th hidden node. The output weights  $\beta$  are evaluated by  $\beta = H^\dagger f_L$ , where  $\beta = [\beta_1, \beta_2, \dots, \beta_L]^T$ ,  $H^\dagger$  denotes the Moore-Penrose generalized inverse of  $H$ .

### B. Constructive Algorithms

Constructive algorithms, owing to their incremental construction approach, tend to discover the minimal network structure. Consequently, these algorithms have been adapted to RWNNs, leading to the introduction of IRWNNs. In particular, if an IRWNN with  $L - 1$  hidden nodes fails to meet the termination condition, a new hidden node is generated through the following two steps:

1) The input weights  $\omega_L$  and bias  $b_L$  are randomly generated from the fixed interval  $[-\lambda, \lambda]^d$  and  $[-\lambda, \lambda]$ . In particular,  $\lambda$  usually takes the value 1. Then, the output vector  $g_L$  of the  $L$ -th hidden node, which is determined by maximizing  $\Delta = \frac{\langle e_{L-1}, g_L \rangle^2}{\|g_L\|^2}$ , where  $e_{L-1} = f - f_{L-1} = [e_{L-1,1}, e_{L-1,2}, \dots, e_{L-1,m}]$  is the current network residual error.  $f_{L-1}$  is the output of IRWNNs with  $L - 1$  hidden nodes.

2)  $\beta_L$  of the  $L$ -th hidden node can be evaluated by  $\beta_L = \frac{\langle e_{L-1}, g_L \rangle}{\|g_L\|^2}$ .

If  $e_L = f - f_L$  does not meet the predefined residual error, then a new hidden node should be added until either the predefined residual error is achieved or the maximum number of hidden nodes is reached.

## III. INTERPRETABLE CONSTRUCTIVE ALGORITHM

This section establishes the interpretable geometric information constraint by leveraging the geometric relationship between the residual error and the hidden parameters. The

universal approximation property of this constraint is ensured by integrating the residual error. Furthermore, a node pool strategy is adopted to acquire hidden parameters that promote convergence. Lastly, two distinct algorithm implementations are introduced: IC and IC+.

### A. Interpretable Geometric Information Constraint

**Theorem 1:** Suppose that  $\text{span}(\Gamma)$  is dense in  $L^2$  and  $\forall g \in \Gamma$ ,  $0 < \|g\| < v$  for some  $v \in R$ . Given  $0 < \sigma < 1$ ,  $\sigma = \sigma + \text{rand}(1 - \sigma, 1)$ ,  $\tau = \frac{1 + \sigma L}{1 + L}$ . If  $g_L$  is randomly generated under geometric constraint

$$\cos \theta_L \geq \gamma_L \langle e_L, e_L \rangle \quad (1)$$

The output weights  $\beta_L$  are evaluated by  $\beta_L = \frac{\langle e_{L-1}, g_L \rangle}{\|g_L\|^2}$ . Then, we have  $\lim_{L \rightarrow +\infty} \|e_L\| = 0$ .

**Proof:** It has been proved that  $\|e_L\|$  is monotonically decreasing as  $L \rightarrow \infty$ .

It follows from Eq. (1) that

$$\begin{aligned} & \|e_L\|^2 - \tau \|e_{L-1}\|^2 \\ &= \sum_{q=1}^m \langle e_{L-1,q} - \beta_{L,q} g_L, e_{L-1,q} - \beta_{L,q} g_L \rangle \\ &\quad - \sum_{q=1}^m \tau \langle e_{L-1,q}, e_{L-1,q} \rangle \\ &= (1 - \tau) \sum_{q=1}^m \|e_{L-1,q}\|^2 - \sum_{q=1}^m \frac{\langle e_{L-1,q}, g_L \rangle^2}{\|g_L\|^2} \\ &\leq \gamma_L \sum_{q=1}^m \|e_{L-1,q}\|^2 - \sum_{q=1}^m \frac{\langle e_{L-1,q}, g_L \rangle^2}{\|g_L\|^2} \\ &= \gamma_L \|e_{L-1}\|^2 - \frac{\langle e_{L-1}, g_L \rangle^2}{\|g_L\|^2} \end{aligned} \quad (2)$$

According to  $e_L$  and  $\beta_L = \frac{\langle e_{L-1}, g_L \rangle}{\|g_L\|^2}$ , then

$$\begin{aligned} & \langle e_L, g_L \rangle \\ &= \langle e_{L-1} - \beta_L g_L, g_L \rangle \\ &= \langle e_{L-1}, g_L \rangle - \frac{\langle e_{L-1}, g_L \rangle}{\|g_L\|^2} \langle g_L, g_L \rangle \\ &= 0 \end{aligned} \quad (3)$$

Then, Eq. (4) means  $e_L \perp g_L$ . It can be easily observed that  $e_L$ ,  $e_{L-1}$  and  $\beta_L g_L$  satisfy the geometric relationship shown in Fig. 1.

In addition, based on  $f_{L-1} = \sum_{j=1}^{L-1} \beta_j g_j$ , we have

$$\begin{aligned} & \sum_{j=1}^{L-1} \beta_j \langle e_{L-1}, g_j \rangle \\ &= \left\langle e_{L-1}, \sum_{j=1}^{L-1} \beta_j g_j \right\rangle \\ &= \|e_{L-1}\|^2 \end{aligned} \quad (4)$$

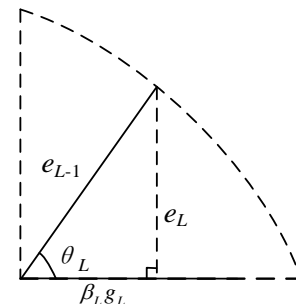


Fig. 1. Relationship between  $e_L$ ,  $e_{L-1}$  and  $\beta_L g_L$ .



TABLE I  
DESCRIPTION OF BENCHMARK DATASETS

Datasets	Training set	Testing set	Various	Label
Sement	2000	400	1	1
HAR	2444	258	561	6
Iris	7200	5330	8	1
Winequality	1120	479	15	3
Concrete	105	45	4	3
Compactiv	7563	2947	15	1
Function	1237	523	1	1

TABLE II  
EXPERIMENTAL PARAMETERS

Datasets	IRWNNs		CIRWN/ICA/ICA+		$\ell$	$L_{\max}$
	$\lambda$	$T_{\max}$	$\zeta$	$T_{\max}$		
HAR	150		15:1:20	15	0.05	100
Winequality	0.5		0.5:0.1:5	2	0.05	200
Sement	1		1:2:13	10	0.05	200
Compactiv	1	1	0.:0.1:5	20	0.05	300
Iris	20		0.5:0.5:5	14	0.05	150
Function	0.1		0.5:0.1:5	12	0.05	400
Concrete	10		10:1:20	10	0.05	50

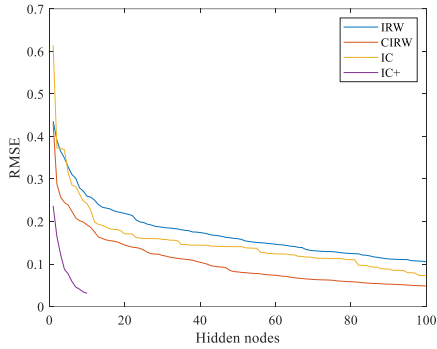


Fig. 4. Convergence of RMSE for four algorithms.

where  $\beta^{\text{previous}}$  denotes the output weights before a new hidden node is added.

#### IV. EXPERIMENTAL RESULTS

In this section, we present a comparative analysis of the proposed methods: IC, IC+ and a detailed comparison of IRWNN and CIRWN. We utilize a function dataset, six benchmark datasets, a milling dataset and a gesture recognition dataset. These datasets are normalised to between [0,1]. Table I shows the details of the datasets. Table II then summarises the experimental parameters of each algorithm. The experimental parameter configurations provided in this paper are the best results obtained after extensive experiments.

$$f(x) = \frac{1}{((x-0.3)^2 + 0.01)} + \frac{1}{((x-0.9)^2 + 0.04)} - 6 \quad (15)$$

In MATLAB 2020a, the comparison experiments were conducted on a PC. Each experiment underwent multiple repetitions, and the final reported results were derived from the average values obtained. The performance evaluation of all stochastic algorithms was based on modeling accuracy, Root Mean Square Error (RMSE), and modeling time.

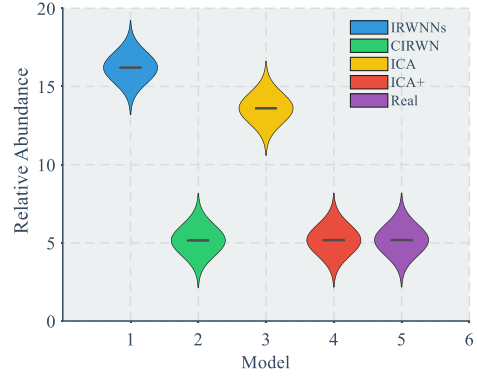


Fig. 5. KDF for four algorithms.

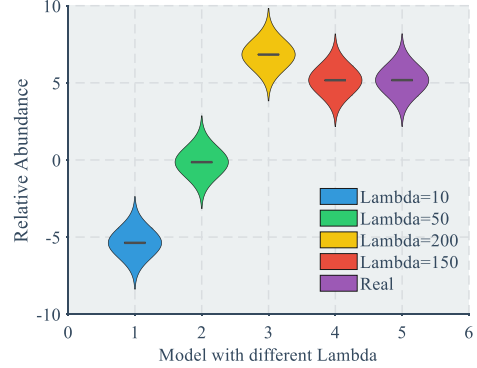


Fig. 6. Impact of different  $\lambda$  on KDF.

#### A. Results

1) *Function Approximation Dataset*: Figure 4 illustrates the RMSE performance of the four algorithms across the function dataset. Evidently, both the proposed method and CIRW demonstrate rapid convergence to the desired RMSE with a minimal number of hidden nodes. Additionally, both IC variants require fewer hidden nodes to achieve the pre-defined objective, underscoring the compactness advantage of the proposed IC approach. Moving to Figure 5, it presents the Kernel Density Function (KDF) of the estimation error. Notably, the KDF of IC closely aligns with the distribution of real data, indicating superior prediction capability compared to other algorithms. Figure 6 explores the impact of different parameters  $\lambda$  on the KDF of IC. Results indicate that varying parameters yield varying effects on the KDF performance of IC, suggesting that maintaining  $\lambda$  as a fixed value may not optimize KDF performance.

2) *Benchmark Datasets*: In this section, we analyze the performance of the four algorithms across various benchmark datasets predominantly sourced from KEEL and UCI repositories. The data presented in Table III offers insights into both the time taken and the RMSE metrics of these algorithms on the benchmark datasets. The analysis of Table III reveals a consistent trend: IC consistently achieves lower training RMSE values compared to CIRW across all datasets. This trend suggests that the integration of geometric information constraints aids in the generation of higher-quality nodes. Moreover, it is noteworthy that ICA demonstrates weaker

TABLE III  
COMPARISON WITH FOUR ALGORITHMS IN TERMS OF TIME, TRAINING RMSE AND TESTING RMSE ON THE BENCHMARK DATASETS

Dataset	Training time Training RMSE Testing RMSE											
	IRW			CIR			IC			IC+		
Compactiv	0.234s	0.283	0.257	1.215s	0.062	0.071	1.122s	0.062	0.072	0.580s	0.062	0.072
Concrete	0.278s	0.253	0.256	1.09s	0.106	0.206	0.852s	0.095	0.206	0.281s	0.134	0.232
Winequality	0.120s	0.308	0.308	0.314s	0.150	0.159	0.287s	0.150	0.185	0.069s	0.120	0.185
HAR	0.059s	0.160	0.239	0.08s	0.019	0.043	0.080s	0.017	0.042	0.022s	0.028	0.043
ORE	56.401s	0.117	0.149	123.90s	0.016	0.055	123.177s	0.314	0.036	40.259s	0.014	0.038
Segment	0.318s	0.277	0.479	0.590s	0.199	0.218	0.536s	0.194	0.216	0.265s	0.193	0.226

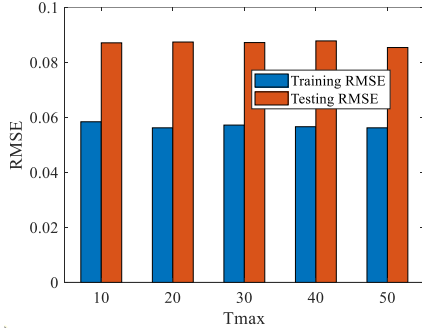


Fig. 7. Impact of node pool on RMSE.

RMSE performance when compared to IC+. This difference can be attributed to IC+'s heavy reliance on the quality of output weights from the initial hidden node. In contrast, IC utilizes the Moore-Penrose generalized inversion method to compute output weights, enabling the acquisition of globally optimal output weights with the addition of each node.

When adding the same hidden nodes, it's observed that the proposed IC requires less time compared to CIRW. Additionally, the training time of IC+ exhibits a significant reduction across most datasets compared to IC, highlighting its superiority in terms of efficiency. Moving on to Figure 7, it illustrates the impact of hidden node pooling on IC. The experimental findings suggest that both excessively large and excessively small node pools result in an increase in RMSE. Moreover, the node pool size directly influences the network's efficiency, underscoring the careful consideration given to this parameter during the experimentation process in this study.

### B. Hand Gesture Recognition Case

Hand Gesture Recognition (HGR) finds applications across various domains such as Augmented Reality (AR), smart homes, and human health monitoring [24]. In this section, we assess the performance of IC within our self-developed HGR system. The HGR system, as depicted in Figure 8, comprises two key modules: hardware and software. The hardware module, illustrated in Figure 9, handles gesture data acquisition and transmission. Meanwhile, the software module, depicted in Figure 10, involves feature extraction. For the evaluation, we utilize the feature extraction module of the software component to obtain the HGR dataset. This dataset comprises a total of 2536 samples [25].

1) *Parameter Configuration:* For IRW, the random interval for the hidden node parameter is [-150,150]. The

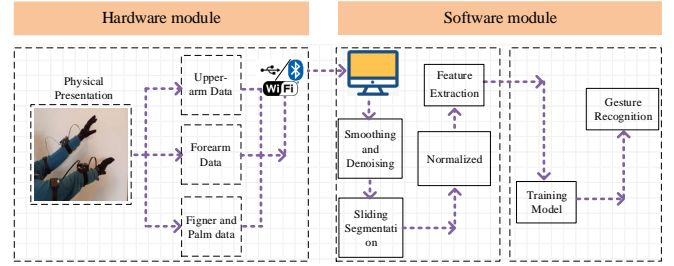


Fig. 8. Framework diagram of gesture recognition system.

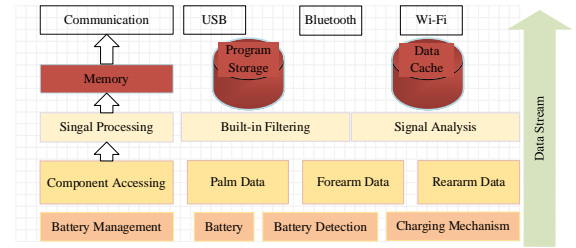


Fig. 9. Hardware module.

TABLE IV  
PERFORMANCE COMPARISON OF FOUR ALGORITHMS ON GESTURE RECOGNITION SYSTEM

Algorithms	Training time	Accuracy	Nodes
IRWNNs	20.37s	82.43%	500
CIRWN	41.63s	95.12%	500
ICA	40.79	96.10%	500
ICA+	14.68s	96.48%	500

random interval for IC and other methods is chosen from  $\zeta = \{150 : 10 : 200\}$ . For IC and CIRW,  $L_{\max} = 500$ , and  $T_{\max} = 20$ .

2) *Comparison and Discussion:* Figure 11 illustrates the average RMSE of IC alongside its comparative methods in the context of HGR, derived from multiple sub-experiments. Notably, IC exhibits commendable stability performance in terms of RMSE. Additionally, the disparity between the maximum and minimum RMSE values is observed to be 0.15 and 0.02 for IRW and CIRW, respectively. Table IV presents the experimental outcomes of the four methods applied to HGR. Notably, IC showcases significant advantages over IRW and CIRW in both training time and accuracy metrics. Through a comprehensive comparison and analysis of these results, it is evident that the proposed IC outperforms IRW and CIRW in the HGR task.

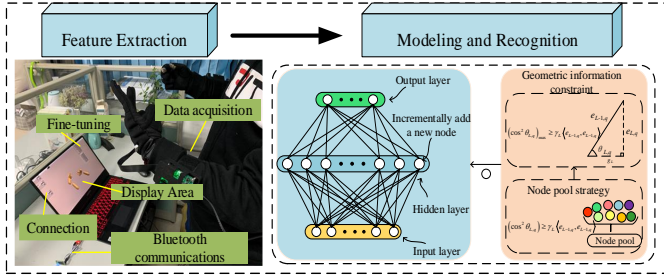


Fig. 10. Software module.

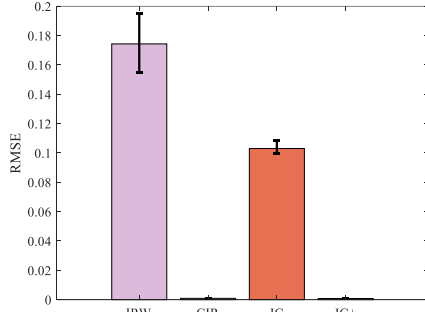


Fig. 11. RMSE of four algorithms.

### C. Ore grinding Case

Ore grinding involves the monolithic dissociation of valuable minerals, as depicted in Figure 12. The mechanism of ore grinding is intricate and challenging to mathematically model. Hence, it is imperative to devise a data-driven model to monitor this process effectively. The ore grinding model is closely intertwined with five key process variables. These variables are obtained from the Ore Grinding Semi-Physical Simulation Platform, illustrated in Figure 13, comprising 1000 training samples and 500 test samples. The primary objective of the ore grinding model is to establish mapping relationships between these variables.

$$PS = f(R_1, R_2, R_3, \alpha_1, \alpha_2) \quad (16)$$

1) *Parameter*: For IRW, the random interval for the hidden node parameter is  $[-150, 150]$ . The random interval for IC and other methods is chosen from  $\zeta = \{150 : 10 : 200\}$ . For IC and CIRW,  $L_{\max} = 500$ , and  $T_{\max} = 20$ .

2) *Discussion*: Figure 14 depicts the PDF of errors for IC and its comparative methods. Notably, IC and IC+ share a PDF curve due to their similar performance. It is evident that both IC and IC+ outperform the other methods, as their curves closely resemble a normal distribution compared to IRW and CIRW. This observation indicates that IC exhibits superior performance in the context of ORE. Table V provides the experimental results of the four methods applied to ORE, with respective modeling times of 3.29 s, 1.91 s, 3.93 s, and 0.57 s. Particularly, IC+ demonstrates the shortest training time while maintaining high accuracy. Although IC exhibits a longer training time compared to CIRW, it performs exceptionally well in terms of accuracy. Based on the experimental findings and analyses, it is evident that the proposed IC excels in terms

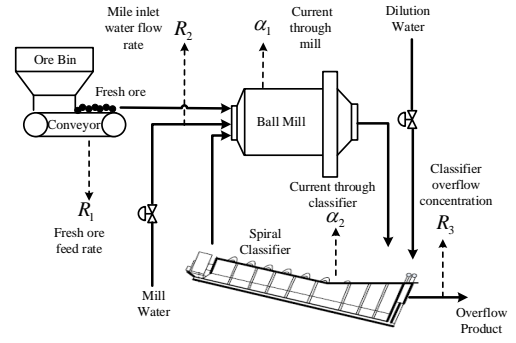


Fig. 12. Flow chart of ORE.

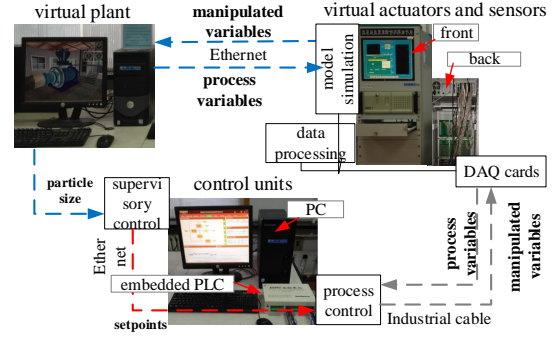


Fig. 13. Simulation Platform.

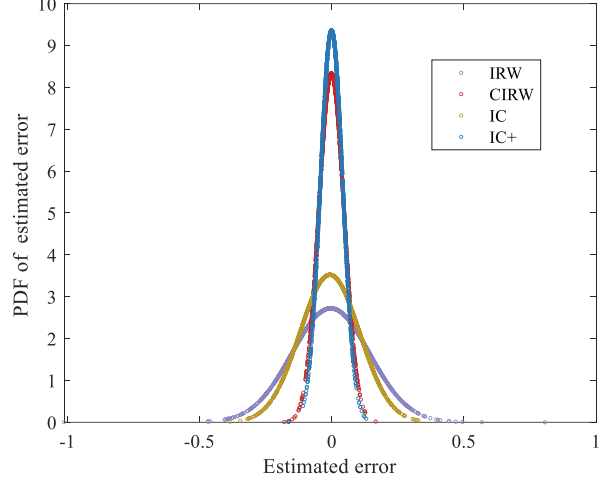


Fig. 14. PDF of IRWNNs, ICA, ICA+, and CIRWN.

TABLE V  
PERFORMANCE COMPARISON OF FOUR ALGORITHMS ON ORE GRINDING SEMI-PHYSICAL SIMULATION PLATFORM

Algorithms	Training time	Accuracy	Nodes
IRWNNs	0.57s	85.89%	100
CIRWN	3.93s	95.53%	100
ICA	3.29s	95.67%	100
ICA+	1.91s	95.81%	100

of generalizability performance and training time. In summary, IC emerges as the optimal choice for ORE.



## V. CONCLUSION

This paper introduces the Interpretable Construction Approach (ICA) aimed at providing a visual representation of the impact of each hidden node on the network residual error. In ICA, nodes are generated randomly adhering to geometric constraints. Additionally, an extension of ICA, termed IC+, is proposed to enhance efficiency. The key distinction lies in the utilization of a more lightweight and efficient method for evaluating output weights. Experimental findings across various public and real datasets demonstrate that ICA effectively mitigates computational overhead while enhancing network performance compared to alternative methods.

## REFERENCES

- [1] X. Meng, J. Tang and J.-F. Fei, "NOx emissions prediction with a brain-inspired modular neural network in municipal solid waste incineration processes," in *IEEE Transactions on Industrial Informatics.*, vol. 18, pp. 4622–4631, 2022.
- [2] Z.-Q. Geng, Z.-W. Chen, Q.-C. Meng and Y.-M. Han, "Novel transformer based on gated convolutional neural network for dynamic soft sensor modeling of industrial processes," in *IEEE Transactions on Industrial Informatics.*, vol. 18, pp. 1521–1529, 2022.
- [3] H. Jung, B. Lee, "Wireless Power and Bidirectional Data Transfer System for IoT and Mobile Devices," in *IEEE Transactions on Industrial Electronics.*, vol. 69, pp. 11832–11836, 2022.
- [4] H.-F. Zhang, Y. Dong, C.-X. Dou and G.-P. Hancke, "PBI based multi-objective optimization via deep reinforcement elite learning strategy for micro-grid dispatch with frequency dynamics," in *IEEE Transactions on Power Systems.*, vol. 38, pp. 488–498, 2023.
- [5] Y.-H. Jia, S. Kwong and R. Wang, "Applying exponential family distribution to generalized extreme learning machine," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*, vol. 50, pp. 1794–1804, 2020.
- [6] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," in *Computer.*, vol. 25, pp. 76–79, 1992.
- [7] Y.-H. Pao, G.-H. Park, D.-J. Sobajic, "Learning and generalization characteristics of the random vector Functional-link net," in *Neurocomputing.*, vol. 6, pp. 163–180, 1994.
- [8] B. Igel'nik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," in *IEEE Transactions on Neural Networks.*, vol. 6, pp. 1320–1329, 1995.
- [9] F. Han, J. Jiang, Q. H. Ling, and B. Y. Su, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," in *Neurocomputing.*, vol. 335, pp. 261–273, 2019.
- [10] X. Wu, P. Rozycki and B. M. Wilamowski, "A hybrid constructive algorithm for single-layer feedforward networks learning," in *IEEE Transactions on Neural Networks and Learning Systems.*, vol. 26, pp. 1659–1668, 2015.
- [11] L.-Y. Ma and K. Khorasani, "Insights into randomized algorithms for neural networks: Practical issues and common pitfalls," in *IEEE Transactions on Neural Networks and Learning Systems.*, vol. 16, pp. 821–833, 2005.
- [12] J.-X. Peng, K. Li, G.-W. Irwin, "A New Jacobian Matrix for Optimal Learning of Single-Layer Neural Networks," in *IEEE Transactions on Neural Networks.*, vol. 19, pp. 119–129, 2018.
- [13] Dudek G, "A constructive approach to data-driven randomized learning for feedforward neural networks," in *Applied Soft Computing.*, vol. 112, pp. 107797, 2021.
- [14] S. Ferrari and R.-F. Stengel, "Smooth function approximation using neural networks," in *IEEE Transactions on Neural Networks.*, vol. 16, pp. 24–38, 2005.
- [15] I.-Y. Tyukin and D.-V. Prokhorov, "Feasibility of random basis function approximators for modeling and control," in *2009 IEEE Control Applications, (CCA) & Intelligent Control.*, 2009, pp. 1391–1396.
- [16] X.-L. Zhu, X.-C. Feng, W.-W. Wang, X. Jia, R. He, "A further study on the inequality constraints in stochastic configuration networks," in *Information Sciences.*, vol. 487, pp. 77–83, 2019.
- [17] Dudek G, "Generating random weights and biases in feedforward neural networks with random hidden nodes," in *Information Sciences.*, vol. 481, pp. 33–56, 2019.
- [18] Q.-J. Wang, W. Dai, P. Lin and P. Zhou, "Compact incremental random weight network for estimating the underground airflow quantity," in *IEEE Transactions on Industrial Informatics.*, vol. 13, pp. 426–436, 2022.
- [19] M. Islam, D.-T. Anderson, A.-J. Pinar, T.-C. Havens, G. Scott and J.-M. Keller, "Enabling explainable fusion in deep learning with fuzzy integral neural networks," in *IEEE Transactions on Fuzzy Systems.*, vol. 28, pp. 1291–1300, 2020.
- [20] H. Sasaki, Y. Hidaka and H. Igarashi, "Explainable deep neural network for design of electric motors," in *IEEE Transactions on Magnetics.*, vol. 57, pp. 1–4, 2021.
- [21] C.-L.-P. Chen and Z.-L. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," in *IEEE Transactions on Neural Networks and Learning Systems.*, vol. 29, pp. 10–24, 2018.
- [22] S.-L. Issa, Q.-M. Peng and X.-G. You, "Emotion classification using EEG brain signals and the broad learning system," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*, vol. 51, pp. 7382–7391, 2021.
- [23] L. Cheng, Y. Liu, Z.-G. Hou, M. Tan, D. Du and M. Fei, "A rapid spiking neural network approach with an application on hand gesture recognition," in *IEEE Transactions on Cognitive and Developmental Systems.*, vol. 13, pp. 151–161, 2021.
- [24] H. Cheng, L. Yang and Z. Liu, "Survey on 3D hand gesture recognition," in *IEEE Transactions on Circuits and Systems for Video Technology.*, vol. 26, pp. 1659–1673, 2016.
- [25] G. Yuan, X. Liu, Q. Yan, S. Qiao, Z. Wang and L. Yuan, "Hand gesture recognition using deep feature fusion network based on wearable sensors," in *IEEE Sensors Journal.*, vol. 21, pp. 539–547, 2021.
- [26] W. Dai, X.-Y. Zhou, D.-P. Li, S. Zhu and X.-S. Wang, "Hybrid parallel stochastic configuration networks for industrial data analytics," in *IEEE Transactions on Industrial Informatics.*, vol. 18, pp. 2331–2341, 2022.