





Neuro-Fuzzy Random Vector Functional Link Neural Network for Classification and Regression Problems

M. Sajid , Graduate Student Member, IEEE, A. K. Malik , M. Tanveer , Senior Member, IEEE, and Ponnuthurai N. Suganthan , Fellow, IEEE

Abstract—The random vector functional link (RVFL) neural network has shown the potential to overcome traditional artificial neural networks' limitations, such as substantial time consumption and the emergence of suboptimal solutions. However, RVFL struggles to provide comprehensive insights into its decision-making processes. We propose the Neuro-fuzzy RVFL (NF-RVFL) model by combining RVFL with neuro-fuzzy system. The proposed NF-RVFL model takes humanlike decisions based on the IF-THEN approach and enhances its transparency in decision-making. Within this framework, input features undergo a fuzzification process as they traverse the fuzzy layer. The resulting fuzzified features then navigate a hidden layer through random projection as well as yielding defuzzified values via defuzzification. The defuzzified values, hidden layer outputs and original input features collectively contribute to the output prediction process. The proposed NF-RVFL model employs three distinct clustering methods to establish fuzzy layer centers: randomly initialized centers (referred to as R-means), K-means clustering centers, and fuzzy C-means clustering centers. This approach generates three distinct model variations, namely NF-RVFL-R, NF-RVFL-K and NF-RVFL-C, each producing a diverse set of fuzzified and defuzzified samples. Our research involves experiments on various UCI benchmark datasets, covering binary, multiclass classification, and regression tasks. The statistical tests and comprehensive experimental analyses consistently show that all variations of the proposed NF-RVFL model outperform baseline models, highlighting their generalization capabilities. The proposed NF-RVFL models show the generic nature by being adeptly applicable and excelling in regression as well as classification tasks.

Index Terms—Broad learning system, extreme learning machine, fuzzy neural network, interpretability, neuro-fuzzy, random vector functional link (RVFL) network.

I. INTRODUCTION

ARTIFICIAL neural networks (ANNs) are a class of machine learning models that take their cues from the neural architecture of the human brain. ANNs are made up of linked nodes or “neurons” arranged in layers that process, analyze, and transmit data to make predictions or decisions. ANNs' ability to learn from data, adaptability, and ability to handle complex and nonlinear relationships in various domains (versatility) like image recognition [1], approximation [2], natural language processing [3], and financial forecasting [4] makes them superior among the other machine learning models.

Apart from many advantages, there are some drawbacks to ANN models. (i) The gradient descent (GD) method is a widely used iterative process in ANN that aims to find the best settings for the model's weights and biases by comparing the predicted outputs with the expected ones. However, this approach has some challenges. It can be slow and occasionally settle for local rather than global optimum weights and biases. The learning rate and initialization point in the GD method are critical factors that can greatly influence the model's training process. (ii) Training conventional neural networks on large datasets can be computationally intensive and time-consuming, necessitating powerful hardware. (iii) Another limitation is their “black-box” nature, making it challenging to interpret and understand the reasoning behind their predictions. This lack of transparency can raise concerns, particularly in critical applications like health care and finance.

Despite the abovementioned limitations, ongoing research is continually improving efficiency, interpretability, and robustness of ANNs. Randomized neural networks (RNNs) [5], [6] were introduced as an alternative to GD-based neural networks to overcome their drawbacks. In general, a touch of randomness is inherent in the RNN model's structure or learning process. This is achieved by holding certain parameters fixed during training and calculating the output layer's parameters using an iterative process or closed-form solution [7], [8]. The randomness grants RNNs the ability to learn with fewer adjustable settings, in less time, and without the need for high-end hardware. Random vector functional link (RVFL) neural network [6], [9] is one of the well-established RNNs. In RVFL, the hidden layer's parameters (weights and biases) are initially set randomly, chosen from

Manuscript received 2 September 2023; revised 19 December 2023; accepted 18 January 2024. Date of publication 29 January 2024; date of current version 3 May 2024. This work was supported in part by the Indian government's Department of Science and Technology (DST) in collaboration with the Ministry of Electronics and Information Technology (MeITy) under Grant DST/NSM/R&D_HPC_App1/2021/03.29 and in part by the Science and Engineering Research Board under the Mathematical Research Impact-Centric Support (MATRICS) scheme under Grant MTR/2021/000787. The work of M. Sajid was supported by the Council of Scientific and Industrial Research (CSIR), New Delhi for providing fellowship under the under Grant 09/1022(13847)/2022-EMR-I. Recommended by Associate Editor Z. Cao. (Corresponding authors: M. Tanveer; Ponnuthurai N. Suganthan.)

M. Sajid, A. K. Malik, and M. Tanveer are with the Department of Mathematics, Indian Institute of Technology Indore, Indore 453552, India (e-mail: phd2101241003@iiti.ac.in; phd1801241003@iiti.ac.in; mtanveer@iiti.ac.in).

Ponnuthurai N. Suganthan is with the KINDI Center for Computing Research, College of Engineering, Qatar University, Doha 2713, Qatar (e-mail: p.n.suganthan@qu.edu.qa).

The source code of the proposed NF-RVFL model is available at <https://github.com/mtanveer1/NeuroFuzzy-RVFL>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TFUZZ.2024.3359652>, provided by the authors.

Digital Object Identifier 10.1109/TFUZZ.2024.3359652

a suitable continuous probability distribution. Throughout the training phase, these parameters remain constant. RVFL stands out from other RNNs due to the direct links between the input and output layers. The direct links characteristic has been observed to significantly enhance the learning performance of RVFL by acting as an implicit regularization for the randomization process [10], [11]. RVFL employs the pseudoinverse or the least-squares approach to find the optimal output parameters (weights connecting the input layer to the output layer and hidden layer to the output layer), which offers a closed-form solution. Furthermore, RVFL has fast training speed and universal approximation capability [12], [13].

RVFL successfully addresses the limitations (i) and (ii) of ANNs mentioned earlier, offering promising solutions. However, it still grapples with limitation (iii). RVFL struggles to interpret its decisions or predictions. As a result, the “black-box” nature of RVFL tends to become more pronounced, raising concerns about its comprehensibility.

The fusion of humanlike reasoning, represented by IF-THEN fuzzy rules in a fuzzy system with an ANN’s learning capabilities and connectivity, gives rise to a hybrid system known as a neuro-fuzzy model [14]. Neuro-fuzzy models use fuzzy logic to represent linguistic terms and rules, which are easily understandable to humans [14]. This linguistic interpretability allows experts and stakeholders to comprehend the decision-making process, making it easier to trust and validate the model’s outputs. Neuro-fuzzy models have a transparent and explicit fuzzy rule base, which clearly shows the connections between the input and output variables. These rules can be interpreted and reviewed to gain insights into how the model arrives at its conclusions. Neuro-fuzzy models have garnered attention for their exceptional traits and wide-ranging applications in classification [15], regression [16], nonlinear system identification, control [17], and so on. Many advanced versions of neuro-fuzzy-based models have been developed based on feature selection and rule extraction [18], [19]. Neuro-fuzzy models have notably contributed to the critical field of time series prediction [20], benefiting various domains such as finance [21], fault diagnosis in self-organizing cellular network [22], weather forecasting [23], and so on.

Although neuro-fuzzy models have many advantages, however, designing neuro-fuzzy models presents several challenges. One such challenge is determining the appropriate architecture, including the number of fuzzy rules and neural network layers. Careful tuning is necessary to optimize both the neural network parameters and fuzzy logic rules, as improper choices may lead to suboptimal performance or difficulties in effective model training [24]. While neural networks can assist in extracting suitable fuzzy rules without human intervention, it remains essential to reduce the number of fuzzy rules without compromising accuracy, especially when dealing with high-dimensional inputs. Additionally, the training process for neuro-fuzzy models using traditional iterative methods like GD can be computationally expensive and time-consuming, especially for large datasets. The challenge further intensifies as the training process involves optimizing nonconvex loss functions, making it difficult to find the global minimum and potentially resulting in suboptimal

solutions. To mitigate these issues, careful selection of optimization algorithms becomes essential for achieving better results in the neuro-fuzzy model.

In this article, we propose the neuro-fuzzy RVFL (NF-RVFL) model to deal with the aforementioned pitfalls of the RVFL and neuro-fuzzy models. The integration of RVFL with neuro-fuzzy systems presents a win–win situation for both RVFL and neuro-fuzzy models. A key advantage of neuro-fuzzy models lies in their universal approximation capability and ability to adapt the interpretable IF-THEN rules to the training data [25]. The RVFL has already demonstrated its potential to overcome the limitations of traditional ANNs, and by combining it with neuro-fuzzy systems, the integrated NF-RVFL model can address the lack of interpretability inherent in RVFL. Moreover, RVFL possesses lesser number of unknown parameters in contrast to conventional neuro-fuzzy models. Consequently, the NF-RVFL model, being developed upon the RVFL framework, also inherits the characteristic of less unknown parameters. As a result, the proposed NF-RVFL model is expected to mitigate the memory-intensive characteristics often associated with traditional neuro-fuzzy models and make the proposed NF-RVFL model capable of handling large datasets with better generalization performance. Additionally, the convex quadratic optimization problem of RVFL overcomes the dilemma of nonconvex optimization problem faced by traditional neuro-fuzzy models and produces more optimal solutions. The NF-RVFL model promises to be a versatile and effective solution, with good generalization performance and better interpretable results than the standard RVFL model.

Within the framework of NF-RVFL, input samples traverse the fuzzy layer employing the fuzzification technique. Subsequently, the fuzzified samples traverse a hidden layer through random projection as well as yielding defuzzified values via the defuzzification technique. These defuzzified values, along with hidden layer outputs and the original inputs, collectively contribute to the output prediction process.

The following are the key highlights of this article.

- 1) We propose neuro-fuzzy RVFL (NF-RVFL) model through the amalgamation of the RVFL neural network and the neuro-fuzzy systems. The proposed NF-RVFL model tries to address the lack of interpretability inherent in the original RVFL model.
- 2) In the proposed NF-RVFL model, we use three distinct clustering methods to establish centers within the fuzzy layer: randomly initialized centers (referred to as R-means), K-means clustering centers, and fuzzy C-means clustering centers. As a result, we obtain three different model variations, namely, NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C, each producing a diverse set of fuzzified and defuzzified samples.
- 3) The conducted experiments encompass a range of 29 binary, 28 multiclass, and 15 regression benchmark UCI datasets sourced from diverse domains with varying sizes. Statistical tests and analysis from these experiments illustrate that the proposed NF-RVFL models exhibit superior performance compared to baseline models.

- 4) Numerous ablation analyses have been conducted, focusing on key elements integral to the NF-RVFL model. These include examinations of the direct link, fuzzy layer, defuzzified layer, the number of fuzzy rules, and activation functions.
- 5) A detailed justification for interpretability is achieved through the exploration of partial dependence plots (PDP) [26] and feature importance-based interpretability in the proposed NF-RVFL model. The findings demonstrate the superior interpretability of the proposed NF-RVFL model compared to the standard RVFL model.

The rest of this article is organized as follows. Section II provides a brief introduction to the Takagi-Sugeno-Kang (TSK) fuzzy system and the RVFL model. Section III presents the rigorous mathematical formulation of the proposed NF-RVFL model. Section IV shows the experimental results and analyses. Finally, Section V concludes this article.

II. RELATED WORKS

This section discusses the TSK fuzzy system, which we integrate with the RVFL in Section III. Then, briefly describe RVFL along with its formulation.

A. Notations

Let the training set be $\{(u_r, t_r) \mid u_r \in \mathbb{R}^{1 \times L}, t_r \in \mathbb{R}^{1 \times C}; r \in \{1, \dots, M\}\}$, where L and C represent the number of features of the input sample and the number of classes, respectively, with M training samples. For the classification task, the variable t_r represents the one-hot version of the label corresponding to the input sample u_r , and for the regression task, t_r represents a real value. Let $U = (u_1^t, u_2^t, \dots, u_M^t)^t \in \mathbb{R}^{M \times L}$ and $T = (t_1^t, t_2^t, \dots, t_M^t)^t \in \mathbb{R}^{M \times C}$ be the collection of all input and target vectors, respectively, where $(\cdot)^t$ is the transpose operator.

B. Takagi-Sugeno-Kang (TSK) Fuzzy System

TSK fuzzy system [27] is an effective method for modeling complicated systems under uncertainty. TSK may approximate nonlinear connections with transparency and interpretability by integrating fuzzy sets, fuzzy rules, and local linear models. Its versatility and usefulness make it a popular option in a wide range of real-world applications [28], [29], [30], making it a vital tool in fuzzy logic and artificial intelligence. The TSK is based on a set of IF-THEN fuzzy rules and is typically defined as:

$$\begin{aligned} &\text{If } u_{r1} \text{ is } \mathcal{F}_{j1} \text{ and } u_{r2} \text{ is } \mathcal{F}_{j2} \dots \text{ and } u_{rL} \text{ is } \mathcal{F}_{jL} \\ &\text{then } y_{rj} = \xi_j(u_{r1}, u_{r2}, \dots, u_{rL}), \quad j = 1, 2, \dots, J \end{aligned}$$

where \mathcal{F}_{jp} is a fuzzy set, u_{rs} is the system input ($s = 1, 2, \dots, L$), and J is the number of fuzzy rules. Here, the function ξ_j in a fuzzy system is often described as any relevant function capable of properly describing the output within the given range of fuzzy rules. However, in a fuzzy system, ξ_j is assumed to be a polynomial of the input variables for practical

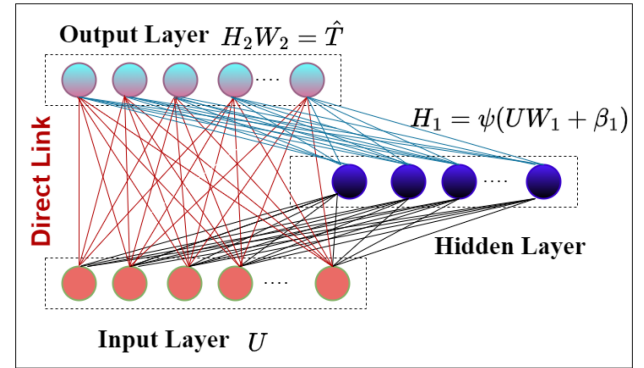


Fig. 1. Architecture of the RVFL model.

purposes. The j th rule's fire strength is calculated as:

$$\mathcal{T}_{rj} = \prod_{s=1}^L \Theta_{js}(u_{rs}), \quad (1)$$

where Θ_{js} is the membership function corresponding to the fuzzy set \mathcal{F}_{js} .

The defuzzification value (Γ) of the input u_r is a real value under the condition that "If u_{r1} is \mathcal{F}_{j1} ... and u_{rL} is \mathcal{F}_{jL} " and given as follows:

$$\Gamma = \frac{\sum_{j=1}^J \mathcal{T}_{rj} y_{rj}}{\sum_{j=1}^J \mathcal{T}_{rj}} = \frac{\sum_{j=1}^J \prod_{s=1}^L \Theta_{js}(u_{rs}) \xi_j(u_{r1}, \dots, u_{rL})}{\sum_{j=1}^J \prod_{s=1}^L \Theta_{js}(u_{rs})}. \quad (2)$$

C. Random Vector Functional Link (RVFL) Network [6]

The RVFL neural network is structured with three layers: input, hidden, and output. Weights connecting the input and hidden layers and biases at the hidden layer are randomly generated and fixed throughout training. The hidden layer features and the original input features (through direct links) are connected to the output layer without any associated bias. The output weights are calculated using analytical methods such as the Moore-Penrose inverse or the least square method. Fig. 1 depicts the architecture of RVFL.

Let H_1 be the hidden layer matrix obtained by the projection of the input matrix using randomly initialized weights and biases followed by the nonlinear activation function ψ , which is defined as:

$$H_1 = \psi(UW_1 + \beta_1) \in \mathbb{R}^{M \times h}, \quad (3)$$

where $W_1 \in \mathbb{R}^{L \times h}$ represents the randomly initialized weights obtained from a normal distribution within the range of $[-1, 1]$ (following [10]) and $\beta_1 \in \mathbb{R}^{M \times h}$ is the bias matrix. Therefore, H_1 is given as:

$$H_1 = \begin{bmatrix} \psi(u_1 w_1 + \zeta_1) & \dots & \psi(u_1 w_h + \zeta_h) \\ \vdots & \vdots & \vdots \\ \psi(u_M w_1 + \zeta_1) & \dots & \psi(u_M w_h + \zeta_h) \end{bmatrix},$$

where $u_i \in \mathbb{R}^{1 \times L}$ is the i th row (sample) of the inputs matrix U , $w_j \in \mathbb{R}^{L \times 1}$ is the j th column vector of the weights matrix

W_1 which connects all the input features to the j th node of the hidden layer and ζ_j is the bias term of the j th hidden node.

The input and hidden features are concatenated as:

$$H_2 = [U \ H_1] \in \mathbb{R}^{(M \times (L+h))}. \quad (4)$$

The predicted output (\hat{T}) of the RVFL model is given using the matrix equation as follows:

$$H_2 W_2 = \hat{T}, \quad (5)$$

where $W_2 \in \mathbb{R}^{(L+h) \times C}$ is the weights matrix connecting the input layer and hidden layer to the output layer. The resultant optimization problem from (5) is formulated as:

$$(W_2)_{\min} = \underset{W_2}{\operatorname{argmin}} \frac{\mathcal{C}}{2} \|H_2 W_2 - T\|_2^2 + \frac{1}{2} \|W_2\|_2^2, \quad (6)$$

where \mathcal{C} is a regularization parameter.

The solution of (6) is given as:

$$(W_2)_{\min} = \begin{cases} (H_2^t H_2 + \frac{1}{\mathcal{C}} I)^{-1} H_2^t T, & (L+h) \leq M \\ H_2^t (H_2 H_2^t + \frac{1}{\mathcal{C}} I)^{-1} T, & M < (L+h) \end{cases}, \quad (7)$$

where I is an identity matrix of conformal dimension.

III. PROPOSED NEURO-FUZZY RANDOM VECTOR FUNCTIONAL LINK (NF-RVFL) NEURAL NETWORK

In this section, we provide a platform for the neuro-fuzzy system (in our case, it is TSK) and RVFL to interact with each other, and we propose a hybrid neuro-fuzzy RVFL (NF-RVFL) model that can make a logical decision based on the IF-THEN rule. In the proposed NF-RVFL's topological structure, input samples undergo the fuzzy layer using the technique of fuzzification. Afterward, the fuzzified samples traverse a hidden layer via random projection as well as produce defuzzified values using the defuzzification technique. These defuzzified values, the hidden layer outputs, and the initial inputs all contribute to the overall output prediction process.

Suppose J denotes the total number of fuzzy rules in the fuzzy layer. Let \mathcal{F}_{js} be the fuzzy set with the membership function Θ_{js} of s th component of j th fuzzy rule of the fuzzy layer, where $s = 1, 2, \dots, L, j = 1, 2, \dots, J$.

If u_{r1} is \mathcal{F}_{j1} and u_{r2} is \mathcal{F}_{j2} ... and u_{rL} is \mathcal{F}_{jL}

then $y_{rj} = \xi_j(u_{r1}, u_{r2}, \dots, u_{rL}), j = 1, 2, \dots, J$.

We consider the first-order TSK fuzzy system (polynomial ξ_j of degree one) then

$$y_{rj} = \xi_j(u_{r1}, u_{r2}, \dots, u_{rL}) = \sum_{s=1}^L \alpha_{js} u_{rs}. \quad (8)$$

Here, α_{js} are the randomly generated coefficients, where $s = 1, 2, \dots, L$ and $j = 1, 2, \dots, J$. The fire strength of the j th fuzzy rule in the fuzzy layer is calculated using (1). For each fuzzy rule, we take the weighted values of the fire strength as:

$$\omega_{rj} = \frac{\mathcal{T}_{rj}}{\sum_{j=1}^J \mathcal{T}_{rj}}. \quad (9)$$

In our work, we consider the Gaussian membership function within fuzzy sets of the fuzzy layer. The membership value of u_{rs} for the j th fuzzy rule is defined as:

$$\Theta_{js}(u_{rs}) = e^{-\left(\frac{u_{rs} - c_{js}}{\sigma_{js}}\right)^2}, \text{ for } s = 1, 2, \dots, L, \quad (10)$$

where $c_j = (c_{j1}, \dots, c_{jL})$ is the center and $\sigma_j = (\sigma_{j1}, \dots, \sigma_{jL})$ is the standard deviation in the j th fuzzy rule. In the proposed NF-RVFL model, we use clustering methods to determine the centers c_j in the fuzzy layer. Here, j ranges from 1 to J , where J represents the total number of clusters (centers) in the fuzzy layer. We employed the following three clustering methods:

- 1) randomly initialized centers (say, R-means);
- 2) using K-means clustering;
- 3) using Fuzzy C-means clustering.

Noteworthy remarks:

- 1) Three types of neuro-fuzzy RVFL emerge as a result of these three approaches of initializing the centers. As a result, it adds diversity to our proposed model.
- 2) Neuro-fuzzy RVFL with R-means, K-means, and fuzzy C-means clustering techniques are named NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C, respectively.
- 3) The number of fuzzy rules (centers) in the fuzzy layer is equal to the number of centers. More specifically, the "K" of K-means is equal to J [number of clusters (centers)] needed in the fuzzy layer.

The topology of the proposed NF-RVFL model is divided into three components. In the first component, fuzzification of the input vectors takes place by projecting the input layer into the fuzzy layer. Then, the fuzzy layer traverses the hidden layer of the RVFL through randomization. In the second component, the calculation of the output of the fuzzy layer takes place through the process of defuzzification. The output layer generation occurs in the third component, incorporating the available features, namely the original features, hidden layer features (generated through fuzzification), and the defuzzified features (generated through defuzzification). The idea behind the output layer generation involves projecting the original and hidden layer features using output layer weights, followed by translation using the defuzzified features. Fig. 2 depicts the structure of the proposed NF-RVFL model.

Component-1. Fuzzification followed by nonlinear transformation: The fuzzified vector of the r th training sample u_r is given as:

$$Y_r = (\omega_{r1} y_{r1}, \omega_{r2} y_{r2}, \dots, \omega_{rJ} y_{rJ}), r = 1, 2, \dots, M. \quad (11)$$

The fuzzy layer matrix corresponding to the input matrix U is given as:

$$Y = (Y_1^t, Y_2^t, \dots, Y_M^t)^t \in \mathbb{R}^{M \times J}. \quad (12)$$

After the fuzzification process, the fuzzy layer features undergo the hidden layer. Let there be h number of nodes in the hidden layer, $W_H \in \mathbb{R}^{J \times h}$ a randomly generated weights matrix connecting the fuzzy layer to the hidden layer and $\beta_H \in \mathbb{R}^{M \times h}$ be the randomly generated bias matrix in the hidden layer with all

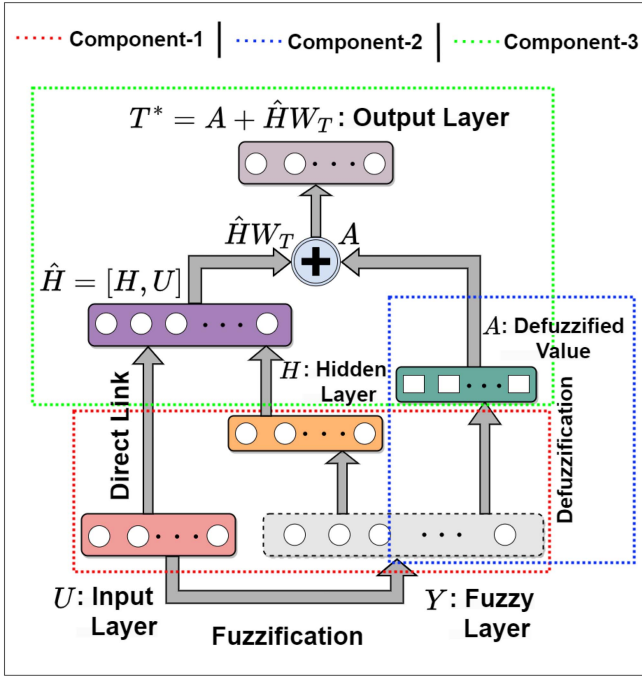


Fig. 2. Architecture of the neuro-fuzzy RVFL model.

columns identical. The hidden layer matrix H is defined as:

$$H = \psi(YW_H + \beta_H) \in \mathbb{R}^{M \times h}, \quad (13)$$

where ψ is the activation function.

Component-2. Defuzzification of the fuzzy subsystems: For the THEN part of the fuzzy layer, we calculate the defuzzified output of the fuzzy layer, which participates in anticipating the final output along with original features (through direct links) and the hidden layer. Since the target matrix $T = (t_1^t, t_2^t, \dots, t_M^t)^t \in \mathbb{R}^{M \times C}$, therefore, the output of the fuzzy layer must have C features. The defuzzified output $\mathcal{A}_r \in \mathbb{R}^{1 \times C}$ of the sample u_r of the fuzzy layer is defined as follows:

$$\begin{aligned} \mathcal{A}_r &= \left(\sum_{j=1}^J \omega_{rj} \lambda_{j1} y_{rj}, \dots, \sum_{j=1}^J \omega_{rj} \lambda_{jC} y_{rj} \right) \\ &= \left(\sum_{j=1}^J \omega_{rj} \lambda_{j1} \sum_{s=1}^L \alpha_{js} u_{rs}, \dots, \sum_{j=1}^J \omega_{rj} \lambda_{jC} \sum_{s=1}^L \alpha_{js} u_{rs} \right) \\ &= \left(\omega_{r1} \sum_{s=1}^L \alpha_{1s} u_{rs} \lambda_{11} + \dots + \omega_{rJ} \sum_{s=1}^L \alpha_{Js} u_{rs} \lambda_{J1}, \right. \\ &\quad \dots, \\ &\quad \left. \omega_{r1} \sum_{s=1}^L \alpha_{1s} u_{rs} \lambda_{1C} + \dots + \omega_{rJ} \sum_{s=1}^L \alpha_{Js} u_{rs} \lambda_{JC} \right) \\ &= \left(\omega_{r1} \sum_{s=1}^L \alpha_{1s} u_{rs}, \dots, \omega_{rJ} \sum_{s=1}^L \alpha_{Js} u_{rs} \right) \Lambda, \end{aligned} \quad (14)$$

$$\text{where } \Lambda = \begin{bmatrix} \lambda_{11} & \dots & \lambda_{1C} \\ \vdots & \vdots & \vdots \\ \lambda_{J1} & \dots & \lambda_{JC} \end{bmatrix}, \text{ and } \lambda_{jc} \text{ is the parameter used}$$

in THEN part of the neuro-fuzzy system to defuzzify it ($j = 1, 2, \dots, J$ and $c = 1, 2, \dots, C$). The computation of the λ_{jc} 's is done in the component-3 using the closed form solution. Therefore, for the input matrix U , the defuzzified output matrix of the fuzzy layer is given as

$$\begin{aligned} \mathcal{A} &= (\mathcal{A}_1^t, \mathcal{A}_2^t, \dots, \mathcal{A}_M^t)^t \in \mathbb{R}^{M \times C} \quad (15) \\ \mathcal{A} &= \begin{bmatrix} \left(\omega_{11} \sum_{s=1}^L \alpha_{1s} u_{1s}, \dots, \omega_{1J} \sum_{s=1}^L \alpha_{Js} u_{1s} \right) \Lambda \\ \vdots \\ \left(\omega_{M1} \sum_{s=1}^L \alpha_{1s} u_{Ms}, \dots, \omega_{MJ} \sum_{s=1}^L \alpha_{Js} u_{Ms} \right) \Lambda \end{bmatrix} \\ \Rightarrow \mathcal{A} &= \left(\Omega \odot \begin{bmatrix} \sum_{s=1}^L \alpha_{1s} u_{1s} & \dots & \sum_{s=1}^L \alpha_{Js} u_{1s} \\ \vdots & \vdots & \vdots \\ \sum_{s=1}^L \alpha_{1s} u_{Ms} & \dots & \sum_{s=1}^L \alpha_{Js} u_{Ms} \end{bmatrix} \right) \Lambda \\ \Rightarrow \mathcal{A} &= (\Omega \odot (U\alpha^t)) \Lambda, \end{aligned} \quad (16)$$

where \odot represents the componentwise matrix multiplication,

$$\alpha^t = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{J1} \\ \vdots & \vdots & \vdots \\ \alpha_{1L} & \dots & \alpha_{JL} \end{bmatrix} \text{ and } \Omega = \begin{bmatrix} \omega_{11} & \dots & \omega_{1J} \\ \vdots & \vdots & \vdots \\ \omega_{M1} & \dots & \omega_{MJ} \end{bmatrix}. \quad (17)$$

Component-3. Final output: The final prediction is made by transmitting the defuzzified vectors of component-2 in conjunction with the initial features and the features extracted from the hidden layer. This is accomplished by concatenating matrices U and H , which are subsequently passed to the output layer using the output weights $W_T \in \mathbb{R}^{(h+L) \times C}$.

$$\text{Let } \hat{H} = [H, U] \quad (18)$$

$$\text{then } T^* = \mathcal{A} + \hat{H}W_T \quad (19)$$

represents the anticipated output matrix of the proposed model. Equation (19) is rewritten as:

$$\begin{aligned} T^* &= (\Omega \odot (U\alpha^t)) \Lambda + \hat{H}W_T \\ &= \left((\Omega \odot (U\alpha^t)), \hat{H} \right) \begin{bmatrix} \Lambda \\ W_T \end{bmatrix} \\ &= \mathcal{D}W \end{aligned} \quad (20)$$

$$\text{where } \mathcal{D} = \left((\Omega \odot (U\alpha^t)), \hat{H} \right) \in \mathbb{R}^{M \times (L+J+h)} \quad (21)$$

$$\text{and } W = \begin{bmatrix} \Lambda \\ W_T \end{bmatrix} \in \mathbb{R}^{(L+J+h) \times C} \quad (22)$$

is the matrix of the knowns and the unknowns of the proposed NF-RVFL model, respectively. Therefore, we need to find the W only for the proposed model. The resultant optimization problem of (20) is formulated as:

$$(W)_{\min} = \underset{W}{\operatorname{argmin}} \frac{\mathfrak{C}}{2} \| \mathcal{D}W - T \|^2 + \frac{1}{2} \| W \|^2, \quad (23)$$

Algorithm 1: Algorithm to Find the Output Layer Weights of the Proposed NF-RVFL Model.

Require: Input matrix $U = (u_1^t, u_2^t, \dots, u_M^t)^t \in \mathbb{R}^{M \times L}$ and output matrix $T = (t_1^t, t_2^t, \dots, t_M^t)^t \in \mathbb{R}^{M \times C}$, where L and C represent the number of features of the input sample and the number of classes, respectively, with M training samples. J is number of fuzzy rules for the fuzzy layer.

- 1: Let u_r be the r^{th} training sample, where $r = 1, 2, \dots, M$.
 - 2: **Generate:** $\alpha \in \mathbb{R}^{J \times L}$ randomly.
 - 3: **Calculate:** the centers of the j^{th} fuzzy rule, i.e., $c_j = (c_{j1}, c_{j2}, \dots, c_{jL})$ for $j = 1, 2, \dots, J$ of the fuzzy layer by employing R-means or K-means, or fuzzy C-means clustering technique.
 - 4: **Calculate:** ω_{rj} using (9).
 - 5: **Calculate:** Y_r using (11).
 - 6: **Calculate:** Y using (12).
 - 7: **Generate:** $W_H \in \mathbb{R}^{J \times h}$ and $\beta_H \in \mathbb{R}^{M \times h}$ randomly with all columns identical in β_H .
 - 8: **Calculate:** H using (13).
 - 9: **Construct:** $\Omega \in \mathbb{R}^{M \times J}$ using (17).
 - 10: **Construct:** \hat{H} using (18).
 - 11: **Construct:** \mathcal{D} using (21).
 - 12: **Find:** the unknown matrix, output layer weights, W using (24).
-

Source code link:

<https://github.com/mtanveer1/NeuroFuzzy-RVFL>.

where \mathfrak{C} is a regularization parameter.

The weight matrix W is computed as follows:

$$(W)_{\min} = \begin{cases} (\mathcal{D}^t \mathcal{D} + \frac{1}{\mathfrak{C}} I)^{-1} \mathcal{D}^t T, & (L + J + h) \leq M, \\ \mathcal{D}^t (\mathcal{D} \mathcal{D}^t + \frac{1}{\mathfrak{C}} I)^{-1} T, & M < (L + J + h) \end{cases}, \quad (24)$$

where I is an identity matrix of conformal dimension.

A. Time Complexity of the Proposed NF-RVFL Models

The time complexity of the proposed NF-RVFL models depends on three components: 1) clustering technique; 2) fuzzy layer and hidden layer; and 3) $(W)_{\min}$ calculated on (24). Following [31], the time complexity of K-means and fuzzy C-means is $\mathcal{O}(MLJv)$ and $\mathcal{O}(MLJ^2v)$, respectively, where v is the number of iterations. The time complexity of R-means is $\mathcal{O}(J)$. The time complexity of the generation of the fuzzy and hidden layer is $\mathcal{O}(MJh)$. Time complexity in computing $(W)_{\min}$ is $\mathcal{O}(W)$, where $\mathcal{O}(W) = \mathcal{O}((L + J + h)^2 M) + \mathcal{O}((L + J + h)^3)$ if $(L + J + h) \leq M$ or $\mathcal{O}(W) = \mathcal{O}(M^2(L + J + h)) + \mathcal{O}(M^3)$ if $M < (L + J + h)$. Therefore, the time complexity of the NF-RVFL-R is $\mathcal{O}(J) + \mathcal{O}(MJh) + \mathcal{O}(W) \approx \mathcal{O}(MJh) + \mathcal{O}(W)$, the time complexity of NF-RVFL-K is $\mathcal{O}(MLJv) + \mathcal{O}(MJh) + \mathcal{O}(W)$ and the time complexity of NF-RVFL-C is $\mathcal{O}(MLJ^2v) + \mathcal{O}(MJh) + \mathcal{O}(W)$.

IV. EXPERIMENTS AND RESULTS

The detailed experimental setup and a comprehensive rationale for the selection of datasets for the experiment are provided in Section I of the supplementary material. In the next subsection, we discuss the details of the compared baseline models. Then, we discuss the experimental results and statistical analysis on binary, multiclass classification, and regression datasets. In the last subsection, the interpretability of the proposed NF-RVFL models is discussed. In Section III of the supplementary material, we conduct an ablation study on the proposed NF-RVFL models, examining their dependence on key components such as direct link (connecting the input and output layers), fuzzification (fuzzy layer), defuzzification (defuzzified layer), number of fuzzy rules (clusters), and activation functions.

A. Compared Models

We compare the proposed NF-RVFL models to the other nine RNNs. The enumerated RNN models under consideration are as follows.

- 1) RVFL: Standard shallow RVFL model [6].
- 2) ELM: Shallow extreme learning machine (ELM) [32].
- 3) IFRVFL: Intuitionistic fuzzy RVFL [33].
- 4) GEELM-LDA: Graph-embedded ELM with linear discriminant analysis (LDA) [34].
- 5) GEELM-LFDA: Graph-embedded ELM with local Fisher discriminant analysis (LFDA) [34].
- 6) Total-Var-RVFL: Total variance minimization-based RVFL [35].
- 7) MCVELM: Minimum class variance-based ELM [36].
- 8) BLS: Standard broad learning system [37].
- 9) Fuzzy-BLS: BLS based on neuro-fuzzy system [16].
- 10) NF-RVFL-R: The proposed neuro-fuzzy RVFL based on R-means clustering.
- 11) NF-RVFL-K: The proposed neuro-fuzzy RVFL based on K-means clustering.
- 12) NF-RVFL-C: The proposed neuro-fuzzy RVFL based on fuzzy C-means clustering.

B. For Binary Classification: Experimental Results and Statistical Analysis on UCI Datasets

Table I presents experimental results for the proposed NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C models, along with baseline models, namely, RVFL, ELM, IFRVFL, GEELM-LDA, GEELM-LFDA, Total-Var-RVFL, and MCVELM. Tables IX and X of the supplementary materials possess the standard deviations and best hyperparameter setting, respectively, *w.r.t.* each model and dataset. Among them, NF-RVFL-C achieves the highest average accuracy of 83.5591%, followed by NF-RVFL-K at 82.5816%, and NF-RVFL-R at 82.2584%, outperforming other state-of-the-art models. Notably, NF-RVFL-K and NF-RVFL-C exhibit the lowest standard deviations (6.9893 and 7.3014, respectively), indicating the high certainty of their predictions. Average accuracy can be deceptive as it might obscure a model's mixed performance across different datasets. To address this, we followed Demšar [38] recommendations and applied a suite of statistical tests. These included the ranking test, the Friedman

TABLE I

CLASSIFICATION ACCURACIES OF THE PROPOSED NF-RVFL-R, NF-RVFL-K, AND NF-RVFL-C MODELS ALONG WITH RVFL, ELM, IFRVFL, GEELM-LDA, GEELM-LFDA, TOTAL-VAR-RVFL, AND MCVELM ON BINARY UCI DATASETS; TABLES IX AND X OF SUPPLEMENTARY MATERIALS CONTAIN THE STANDARD DEVIATIONS AND BEST HYPERPARAMETER SETTING W.R.T. EACH MODEL AND DATASET

Dataset ↓ Model →	RVFL [6]	ELM [32]	IFRVFL [33]	GEELM-LDA [34]	GEELM-LFDA [34]	Total-Var-RVFL [35]	MCVELM [36]	NF-RVFL-R [†]	NF-RVFL-K [†]	NF-RVFL-C [†]
adult	84.0342	83.8725	83.5438	83.422	83.2136	84.0178	83.9011	83.2992	84.5879	84.6218
bank	89.4051	89.4934	89.1173	89.6158	46.4966	89.6705	89.6703	89.4269	89.4492	90.9391
blood	76.5065	76.9056	77.4398	67.1732	76.2398	76.9065	77.4065	77.838	77.8398	77.4398
breast_cancer	70.1956	70.1754	71.9298	89.8246	84.5796	70.1754	70.5263	72.3351	72.6921	70.8953
breast_cancer_wisc_diag	93.8503	92.2683	89.2719	93.1455	86.988	94.1997	93.4995	94.4753	94.3782	95.6063
chess_krvkp	72.0313	71.936	74.2589	76.7814	69.2351	77.8963	75.8278	82.6045	81.7908	85.2322
congressional_voting	63.6782	63.2184	58.8506	59.7701	54.2529	63.908	63.4483	63.908	63.926	64.1379
connect_4	75.4518	75.4059	75.3407	76.7316	75.4281	75.4992	75.4459	75.4844	75.5022	76.2156
credit_approval	85.5674	85.3623	86.5217	85.5072	85.5072	85.5072	85.5072	85.942	85.942	87.8581
cylinder_bands	66.4154	65.8081	66.8785	67.5405	67.5385	67.7727	66.0156	68.9606	69.724	71.2941
haberman_survival	73.4902	73.8181	75.1348	55.5632	52.2898	74.6398	73.4902	75.2396	75.1296	75.4574
heart_hungarian	73.8223	74.4886	76.8849	83.7288	85.0847	75.8738	74.1613	76.9024	76.5576	78.2466
horse_colic	86.0369	85.424	85.5979	75.8053	74.7501	86.4124	86.1533	86.6938	86.4272	86.6901
miniboone	81.4984	79.2741	84.3698	82.4076	83.3566	81.8659	81.2939	84.3969	84.0217	89.9858
molec_bio_promoter	72.7706	74.632	78.3983	71.8182	77.4892	79.5236	72.684	82.0346	82.12	82.0346
monks_1	83.7934	82.2685	84.6895	73.5811	71.5476	84.1538	84.8697	81.4527	82.3584	86.6992
monks_2	80.0138	81.6804	84.5898	60.7273	57.23	85.8485	82.8499	85.8485	87.3774	88.8894
monks_3	91.1564	90.7912	90.0704	82.8419	77.69	91.6986	91.8821	91.8436	91.6486	92.1744
mushroom	96.3931	96.0735	93.9684	96.3071	98.7561	97.2919	96.7382	94.2369	95.9132	99.3351
mus_k1	72.0614	69.7675	71.864	72.4496	65.9518	72.4846	70.1776	75.2149	76.0482	74.1645
mus_k2	84.5909	84.5909	84.5909	85.1573	84.5909	85.1818	84.909	83.9394	85.501	85.9545
oocytes_trisopterus_nucleus_2f	78.9419	77.5182	75.2219	78.0694	79.8889	79.7036	79.6025	79.3863	79.3779	79.9345
ozone	97.1217	97.1319	97.1612	96.0615	96.0615	97.1612	97.1358	97.1612	97.1612	97.1612
parkinsons	82.2569	80.5128	78.4615	84.1026	83.5897	82.0513	83.5897	83.0769	83.0769	85.5856
pittsburg_bridges_T_OR_D	87.1905	87.3685	89.3025	82.1429	80.1905	89.1429	90.1905	90.1429	88.2381	90.1905
spambase	88.546	87.0472	85.0883	88.9582	88.546	88.6116	87.5819	88.9674	88.9583	89.3931
statlog_german_credit	76.9	75.7	76.9	66.7	53.9	77	76.8	77.1	77.7	77.7
titanic	77.9168	77.9168	79.0532	65.5896	61.0231	78.6901	79.0532	78.4623	79.0532	79.0532
vertebral_column_2classes	71.2903	70.6452	85.4839	94.8387	92.2581	78.806	74.5161	79.0323	82.366	80.3226
Average accuracy	80.4458	80.0378	81.0339	78.8766	75.6489	81.7826	80.9959	82.2584	82.5816	83.5591
Average rank	6.931	7.9483	6.2069	6.3966	7.431	4.8103	5.8793	4.069	3.5172	1.8103
Average standard deviation	9.4944	10.2999	7.9884	11.0356	11.2942	9.116	8.7296	8.0372	6.9893	7.3014

The boldface in each row denotes the performance of the best model corresponding to the datasets. † represents the proposed models.

test, and the Nemenyi post hoc test. These tests allowed us to comprehensively assess model performance, avoiding biasness, and enabling us to make broad conclusions about their effectiveness. In the ranking approach, models are ranked based on their performance across individual datasets, assigning higher ranks to poorer performers and lower ranks to top performers. Let us consider \mathcal{M} models being assessed across \mathcal{D} datasets. The rank of the m th model on the d th dataset is represented as $\rho(m, d)$. The average rank of the m th model is calculated as follows: $\rho(m, *) = (\sum_{d=1}^{\mathcal{D}} \rho(m, d)) / \mathcal{D}$. The average ranks of the models are noted in the last row of Table I. With average lower ranks 1.8103, 3.5172, and 4.069, all the proposed NF-RVFL-C, NF-RVFL-K, and NF-RVFL-R models, respectively, secure the top three positions and demonstrate superiority among the baseline models. The Friedman test [39] is employed to compare the average ranks of models and ascertain whether significant differences exist among them based on their rankings. This test utilizes a chi-squared distribution (χ_F^2) with $\mathcal{M} - 1$ degrees of freedom (dof). The formula for Friedman's test is given as: $\chi_F^2 = \frac{12\mathcal{D}}{\mathcal{M}(\mathcal{M}+1)} (\sum_{m=1}^{\mathcal{M}} (\rho(m, *))^2 - \frac{\mathcal{M}(\mathcal{M}+1)^2}{4})$. Iman and Davenport [40] exhibited that Friedman's χ_F^2 statistic might exhibit an overly cautious behavior. As a remedy, they introduced an improved statistic, denoted as F_F statistic, which is computed as follows: $F_F = \chi_F^2 (\frac{\mathcal{D}-1}{\mathcal{D}(\mathcal{M}-1)-\chi_F^2})$. The distribution of F_F is characterized by $(\mathcal{M} - 1)$ and $(\mathcal{D} - 1)(\mathcal{M} - 1)$ dof. For $\mathcal{M} = 10$ and $\mathcal{D} = 29$, we obtain $\chi_F^2 = 105.3045$ and $F_F = 18.9378$. Referring to the statistical table for the F -distribution, we find that $F_F(9, 252) = 1.9171$ at a significance level of 5%. The null hypothesis is rejected because $18.9378 > 1.9171$, indicating significant differences among the models. Consequently, the Nemenyi post hoc test [38] can be employed to explore the pairwise statistical difference between the models. Under this

test, if the average rank of the model m_1 is lower by at least the critical difference ($C.D.$) compared to the model m_2 ; it is concluded that the performance of the model m_1 is statistically superior to that of the model m_2 . The $C.D.$ is calculated as:

$C.D. = q_\alpha (\sqrt{\frac{\mathcal{M}(\mathcal{M}+1)}{6\mathcal{D}}})$. Here, q_α represents the critical value for the two-tailed Nemenyi test, sourced from the precalculated distribution table in [38]. Following the computation, we arrive at $C.D. = 2.5157$ at a significance level of 5% (i.e., at $\alpha = 0.05$). Table II showcases the statistical analysis of the proposed models with all the models based on the Nemenyi post hoc test. Table II confirms that the proposed NF-RVFL-C is statistically better than all the existing models. The proposed NF-RVFL-K is statistically superior to all the existing models except Total-Var-RVFL and MCVELM. The proposed NF-RVFL-R model is statistically better than the existing RVFL, ELM, and GEELM-LFDA models. Although the proposed NF-RVFL-R is not significantly superior to the existing IFRVFL, GEELM-LDA, Total-Var-RVFL, and MCVELM, however, the lower ranks of the proposed NF-RVFL-R model is a clear indicator that the proposed models have stronger generalization capability compared to all the existing baseline models. The combination of higher average accuracy and consistent results from various statistical tests strongly supports the assertion that the proposed NF-RVFL models possess superior generalization compared to the existing baseline models.

C. For Multiclass Classification: Experimental Results and Statistical Analysis on UCI Datasets

Table III showcases the results related to multiclass classification datasets. The IFRVFL model is designed exclusively for binary classification, making it ineligible for inclusion in

TABLE II
STATISTICAL ANALYSIS OF THE PROPOSED MODELS (LISTED IN THE FIRST COLUMN) WITH ALL THE MODELS (LISTED IN THE FIRST ROW) FOR THE BINARY UCI DATASETS BASED ON THE NEMENYI POST HOC TEST

	RVFL [6]	ELM [32]	IFRVFL [33]	GEELM-LDA [34]	GEELM-LFDA [34]	Total-Var-RVFL [35]	MCVELM [36]	NF-RVFL-R [†]	NF-RVFL-K [†]	NF-RVFL-C [†]
NF-RVFL-R [†]	✓	✓	0	0	✓	0	0	N/A	0	0
NF-RVFL-K [†]	✓	✓	✓	✓	✓	0	0	0	N/A	0
NF-RVFL-C [†]	✓	✓	✓	✓	✓	✓	✓	0	0	N/A

Here, † indicates the proposed models. ✓ signifies that the model in the respective row exhibits statistical superiority over the model in the corresponding column. ✗ signifies that the model in the respective row is statistically inferior to the model in the corresponding column. 0 means there is no statistical difference between the row and column models.

TABLE III
CLASSIFICATION ACCURACIES OF THE PROPOSED NF-RVFL-R, NF-RVFL-K, AND NF-RVFL-C MODELS ALONG WITH THE COMPARED BASELINE MODELS, I.E., RVFL, ELM, IFRVFL, TOTAL-VAR-RVFL, MCVELM ON MULTICLASS UCI DATASETS; SUPPLEMENTARY TABLES XI AND XII CONTAIN EACH MODEL'S STANDARD DEVIATIONS AND BEST HYPERPARAMETER SETTING ON DATASETS

Dataset ↓ Model →	RVFL [6]	ELM [32]	Total-Var-RVFL [35]	MCVELM [36]	NF-RVFL-R [†]	NF-RVFL-K [†]	NF-RVFL-C [†]
abalone	63.4665	63.4419	63.754	63.6578	63.8253	65.4586	64.0887
annealing	89.6381	88.3018	89.1937	89.0381	86.6313	87.0782	90.8591
cardiotocography_10classes	66.2324	65.2894	66.1351	66.2307	64.8151	62.8425	69.2395
ecoli	60.9175	61.2116	51.0667	51.0667	60.619	60.6277	62.5405
energy_y1	88.6716	88.2777	89.7106	89.0604	87.979	88.5434	89.0595
flags	53.6302	51.5789	53.1174	54.197	55.2687	55.1957	55.2092
hayes_roth	61.875	60.625	63.5	61.875	61.875	64.075	65
heart_cleveland	59.7268	59.377	61.0328	59.3607	61.0383	61.3388	60.0437
led_display	72.6	72.5	73.5	73	72.7	72.2	75.6
low_res_spect	87.7605	86.6285	73.9852	74.1851	87.3849	87.5666	87.7605
lymphography	86.4138	85.4745	86.4138	86.4368	85.7701	87.1034	88.5057
molec_biol_splice	56.0949	51.8809	54.2633	51.8809	69.2163	68.652	68.3072
nursery	70.3935	70.1775	65.693	62.8627	66.7824	66.3889	71.9444
oocytes_merluccius_states_2f	91.2846	90.8948	91.8737	91.633	91.3845	91.1894	92.067
optical	95.4626	94.3594	95.4626	94.8754	93.3808	94.0214	97.2317
seeds	89.0476	87.1429	90	89.0476	90.4762	90.3653	90.4762
semeion	87.0673	85.4363	87.2562	83.0492	88.0726	88.7725	87.3187
soybean	87.6954	86.9809	88.4264	88.306	87.3991	88.1387	90.4723
statlog_shuttle	98.6483	98.6241	98.681	98.6793	98.6345	98.6638	98.8764
statlog_vehicle	82.0327	81.5593	82.0313	81.3227	79.1974	79.5475	82.2666
thyroid	95.625	95.4722	95.9917	95.5833	94.7617	94.7639	96.0278
wall_following	75.6057	75.239	75.734	75.1845	71.518	72.1223	80.683
waveform	86.54	86.08	86.56	86.26	86.88	86.68	87.22
waveform_noise	86.28	85.02	86.42	85.42	86.28	86.3	86.58
wine	97.2063	96.0952	96.6508	96.7896	97.2063	97.2363	97.2063
wine_quality_red	59.4769	59.4755	59.9134	60.4759	60.5106	60.8533	60.5388
wine_quality_white	52.3989	52.5332	52.472	52.4103	52.4106	52.6349	53.4925
zoo	95	96	97	96	97	95	97.6
Average accuracy	78.4569	77.7028	77.7085	77.0675	78.5363	78.6914	80.222
Average rank	4.1786	5.7143	3.6429	4.75	4.375	3.8036	1.5357
Average standard deviation	5.4311	7.0908	6.8053	8.7373	4.8882	5.1418	4.5872

The boldface in each row denotes the performance of the best model corresponding to the datasets. † represents the proposed models.

the multiclass comparisons. The GEELM-LDA and GEELM-LFDA are the worst-performing models in binary classification as per Section IV-B, so they are excluded from the multiclass classification comparison. Tables XI and XII of supplementary materials contain the standard deviations and best hyperparameter setting, respectively, *w.r.t.* each model and dataset.

The proposed NF-RVFL-C, NF-RVFL-K, and NF-RVFL-R secured the first, second, and third spots with an average accuracy of 80.222%, 78.6914%, and 78.5363%, respectively. With a standard deviation of 4.5872, 4.8882, and 5.1418, the proposed NF-RVFL-C, NF-RVFL-R, and NF-RVFL-K, respectively, have the lowest standard deviations among all the models. This showcases that the proposed NF-RVFL models have the highest prediction capabilities with high certainty. Among all models, the proposed NF-RVFL-C has the lowest rank of 1.5357, while NF-RVFL-K ranks third, highlighting their superiority over baseline models. With an average rank of 4.375, NF-RVFL-R demonstrates strong competitiveness. We get $\chi^2_F = 59.4811$ and

TABLE IV
STATISTICAL ANALYSIS OF THE PROPOSED (LISTED IN THE FIRST COLUMN) WITH ALL THE MODELS (LISTED IN THE FIRST ROW) FOR MULTICLASS UCI DATASETS BASED ON THE NEMENYI POST HOC TEST

	RVFL [6]	ELM [32]	Total-var-RVFL [35]	MCVELM [36]	NF-RVFL-R [†]	NF-RVFL-K [†]	NF-RVFL-C [†]
NF-RVFL-R [†]	0	0	0	0	N/A	0	✗
NF-RVFL-K [†]	0	✓	0	0	0	N/A	✗
NF-RVFL-C [†]	✓	✓	✓	✓	✓	✓	N/A

Here, † indicates the proposed models. ✓ signifies that the model in the respective row exhibits superiority over the model in the corresponding column. ✗ signifies that the model in the respective row is statistically inferior to the model in the respective column. 0 means no statistical difference between the row and column models.

$F_F = 14.7992$ for $\mathcal{M} = 7$ and $\mathcal{D} = 28$. Also, $F_F(6, 252) = 2.1549$ at a 5% significance level. $F_F = 14.7992 > F_F(6, 252)$ demonstrates substantial differences across the compared models; therefore, the null hypothesis is rejected. Table IV presents the Nemenyi post hoc test results for all proposed and baseline models. The proposed NF-RVFL-C is statistically superior to all existing models, as well as the proposed NF-RVFL-R and NF-RVFL-K models. NF-RVFL-K is statistically better than ELM.

TABLE V
COMPARISON OF NEURO-FUZZY BASED RVFL (PROPOSED) AND BLS MODELS IN TERMS OF ACCURACY, STANDARD DEVIATION, RANK, NUMBER OF FUZZY RULES FOR BINARY AND MULTICLASS CLASSIFICATION

	Model →	RVFL [6]	BLS [37]	Fuzzy-BLS [16]	NF-RVFL-R [†]	NF-RVFL-K [†]	NF-RVFL-C [†]
Binary	Avg. Accuracy	80.4458	81.9585	82.4748	82.2584	82.5816	83.5591
	Avg. Standard Deviation	9.4944	8.2629	8.8137	8.0372	6.9893	7.3014
	Avg. Rank	5.4138	4.3448	3.1552	3.3621	3.069	1.6552
	# Avg. Fuzzy Rules	N/A	N/A	330.52	23.28	27.76	26.38
Multiclass	Avg. Accuracy	78.4569	79.183	79.3606	78.5363	78.6914	80.222
	Avg. Standard Deviation	5.4311	4.9593	6.6522	4.8882	5.1418	4.5872
	Avg. Rank	4.2679	3.6429	3.0179	4.3393	3.8036	1.9286
	# Avg. Fuzzy Rules	N/A	N/A	450.36	30.18	36.25	30

[†] represents the proposed models, and Avg. is used as an abbreviation for average.

The boldface in each row denotes the performance of the best model corresponding to the datasets.

MCVELM and standard ELM get the last and second-last spots in the ranking scheme, highlighting the RVFL-based family's superior generalization performance. Statistical tests confirm the proposed NF-RVFL models' top performance, attributed to enhanced generalization performance via neuro-fuzzy system integration.

D. Comparison of RVFL and BLS-Based Neuro-Fuzzy Models

Now, we conduct a comparative analysis of two neuro-fuzzy-based models: fuzzy-BLS and our proposed NF-RVFL models. The evaluation is performed on both binary and multiclass UCI datasets, and the comparison is based on the outcomes outlined in Table V. Further detailed experimental results, standard deviations, and best parameter settings are presented in supplementary Tables XIII to XVIII.

Analysis:

- 1) The standard BLS outperforms RVFL in terms of average accuracy in both binary and multiclass classification.
- 2) Neuro-fuzzy integration significantly enhances model performance. In binary classification, RVFL and BLS exhibit average accuracies of 80.4458% and 81.9585%, respectively. On the other hand, NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C achieve average accuracies of 82.2584%, 82.5816%, and 83.5591%, respectively. Similarly, fuzzy-BLS attains an average accuracy of 82.4748%, surpassing the accuracy of BLS at 81.9585%. This pattern holds true for multiclass classification as well.
- 3) While standard RVFL models may not match the performance of BLS, notable advancements are observed with NF-RVFL models. NF-RVFL-K and NF-RVFL-C surpass both BLS and fuzzy-BLS in binary classification. NF-RVFL-R outperforms BLS and competes closely with fuzzy-BLS in binary classification. Remarkably, the proposed NF-RVFL-C model outshines all baseline and neuro-fuzzy-based models for multiclass classification. This underscores the effectiveness of our proposed NF-RVFL-C in incorporating a fuzzy inference system within its structure.
- 4) In binary classification, all proposed NF-RVFL models exhibit the lowest standard deviation, and in multiclass classification, NF-RVFL-C, NF-RVFL-R, and NF-RVFL-K secure the first, second, and fourth positions with the lowest standard deviation, respectively. This emphasizes

that the proposed NF-RVFL models not only demonstrate superior prediction capabilities but also achieve heightened certainty.

- 5) NF-RVFL-C secures the top rank with 1.6552 and 1.9286 in binary and multiclass classification, respectively, showcasing a substantial lead over other models. NF-RVFL-K outperforms both BLS and fuzzy-BLS in binary classification.
- 6) The crucial factor of the number of fuzzy rules significantly influences the simplicity and interpretability of neuro-fuzzy models. Notably, the optimal number of fuzzy rules (average) for NF-RVFL models consistently falls within the range of 20–35, whereas Fuzzy-BLS utilizes 330.52 and 450.36 fuzzy rules for binary and multiclass classification, respectively—roughly 15–20 times more than that of NF-RVFL. This highlights the superior structural fuzzy adaptability of our proposed NF-RVFL models, indicating a less complex yet high-performing alternative than fuzzy-BLS.

Main takeaways from the analysis are as follows.

- 1) While RVFL's generalization performance may not rival that of BLS, the proposed NF-RVFL consistently outperforms BLS and fuzzy-BLS and sometimes demonstrates competitive results.
- 2) NF-RVFL exhibits better interpretability than fuzzy-BLS due to its reduced complexity and enhanced structural adaptability, particularly in fuzzy rule utilization.

E. For Regression Task: Experimental Results and Statistical Analysis on UCI Datasets

The significance of regression tasks becomes evident within the RVFL domain, where current research is relatively constrained. Here, we compare the proposed NF-RVFL models with the standard RVFL and ELM models. We fix the value of the regularization parameter $\mathcal{C} = 10^5$. The RMSE value is calculated to measure the performance of the proposed models against the baseline models. $RMSE = \sqrt{\frac{1}{M} \sum_{r=1}^M (t_r - t_r^*)^2}$, where M , t_r , and t_r^* are the total number of samples, real output, and predicted output of the r th sample, respectively. A lower RMSE value signifies superior model performance. Table VI provides an overview of the RMSE values associated with each model across the respective datasets. After analysis, it becomes evident that the proposed NF-RVFL-R model exhibits

TABLE VI

THE *RMSE* VALUES AND THE CORRESPONDING BEST HYPERPARAMETERS OF THE PROPOSED NF-RVFL-R, NF-RVFL-K, AND NF-RVFL-C MODELS ALONG WITH THE COMPARED BASELINE MODELS, I.E., RVFL, ELM ON REGRESSION UCI DATASETS

Model	RVFL [6]	ELM [32]	NF-RVFL-R [†]	NF-RVFL-K [†]	NF-RVFL-C [†]
Dataset (#Samples, #Features)	<i>RMSE</i> (h, Act.)	<i>RMSE</i> (h, Act.)	<i>RMSE</i> (J, h, Act.)	<i>RMSE</i> (J, h, Act.)	<i>RMSE</i> (J, h, Act.)
2D_Planes (40768, 11)	0 (3, 3)	7.3×10^{-2} (183, 1)	0 (10, 3, 6)	0 (10, 3, 3)	1×10^{-6} (50, 203, 4)
Abalone (4177, 8)	6×10^{-6} (203, 4)	7×10^{-6} (203, 5)	2.7×10^{-6} (25, 3, 6)	8.1×10^{-6} (50, 3, 3)	6.1×10^{-6} (40, 3, 3)
Ailerons (13750, 41)	1.5×10^{-5} (43, 6)	6.7×10^{-4} (203, 1)	1.6×10^{-5} (10, 3, 1)	1.6×10^{-5} (50, 3, 2)	1.6×10^{-5} (15, 3, 5)
Airfoil_Self_Noise (1503, 6)	1×10^{-5} (3, 1)	1.7×10^{-5} (203, 2)	3.8×10^{-6} (50, 23, 1)	3.3×10^{-6} (50, 3, 1)	4.1×10^{-6} (25, 23, 6)
California_Housing (20640, 9)	5×10^{-6} (3, 3)	1.7×10^{-5} (203, 2)	1×10^{-6} (45, 3, 3)	1.2×10^{-5} (25, 163, 6)	2.7×10^{-6} (30, 3, 3)
Delta_Ailerons (7129, 6)	0 (3, 4)	8.7×10^{-5} (203, 2)	0 (5, 3, 5)	0 (5, 23, 6)	0 (5, 3, 2)
Elevators (16599, 19)	3×10^{-5} (3, 4)	5.8×10^{-4} (203, 1)	3×10^{-5} (5, 3, 2)	3×10^{-5} (5, 3, 2)	2.8×10^{-5} (5, 3, 6)
Forest_Fires (517, 13)	1.5×10^{-6} (23, 6)	4.7×10^{-3} (183, 1)	7.3×10^{-6} (5, 3, 1)	3.1×10^{-6} (5, 3, 2)	2.7×10^{-6} (5, 3, 1)
Kinematics_Robot_Arm (8192, 9)	0 (3, 2)	2.4×10^{-2} (163, 6)	0 (5, 3, 5)	0 (10, 3, 5)	1×10^{-6} (50, 43, 4)
Parkinsons_Telemonitoring (5875, 22)	2.3×10^{-5} (3, 5)	2.9×10^{-3} (163, 1)	2.1×10^{-5} (10, 23, 6)	2.3×10^{-5} (5, 3, 4)	2.3×10^{-5} (5, 3, 5)
Pole_Telecomm (15000, 49)	0 (3, 3)	9×10^{-2} (203, 1)	0 (10, 23, 3)	0 (10, 3, 3)	0 (10, 3, 6)
Servo (4177, 8)	9×10^{-6} (183, 5)	9×10^{-6} (203, 2)	5.9×10^{-6} (40, 3, 3)	6.2×10^{-6} (30, 3, 6)	6.3×10^{-6} (20, 23, 6)
Stocks_Domain (4177, 8)	6×10^{-6} (163, 6)	7×10^{-6} (163, 4)	5.5×10^{-6} (40, 3, 3)	6.8×10^{-6} (50, 3, 3)	5.6×10^{-6} (45, 3, 3)
Triazines (186, 61)	3.5×10^{-5} (3, 1)	6.9×10^{-1} (203, 6)	2.2×10^{-5} (5, 3, 2)	2.6×10^{-5} (10, 3, 2)	2.2×10^{-5} (5, 3, 6)
Yacht_Hydrodynamics (308, 7)	4×10^{-6} (3, 4)	2×10^{-4} (203, 2)	5.8×10^{-6} (15, 23, 6)	8×10^{-6} (5, 3, 5)	7×10^{-6} (5, 3, 1)
Avg. <i>RMSE</i>	9.6×10^{-6}	5.8×10^{-2}	8.0×10^{-6}	9.4×10^{-6}	8.4×10^{-6}

Here, Avg. and *RMSE* are used as abbreviations for average and root mean square error, respectively. Act. is for the activation function where 1 denotes *Sigmoid*, 2 denotes *Sine*, 3 denotes *Tribas*, 4 denotes *Radbas*, 5 denotes *Tansig* and 6 denotes *Relu*. † represents the proposed models. The boldface in each row denotes the performance of the best model corresponding to the datasets.

TABLE VII

COMPARISON OF WIN-TIE-LOSS RESULTS ON UCI REGRESSION DATASETS

	RVFL [6]	ELM [32]	NF-RVFL-R [†]	NF-RVFL-K [†]
ELM [32]	[0, 1, 14]			
NF-RVFL-R [†]	[8, 1, 6]	[15, 0, 0]		
NF-RVFL-K [†]	[4, 1, 10]	[14, 0, 1]	[4, 4, 7]	
NF-RVFL-C [†]	[6, 0, 9]	[15, 0, 0]	[2, 1, 12]	[8, 0, 7]

In the $[x, y, z]$, x , y , and z denote the number of win, tie, and loss of the row model over the corresponding column model.

the most favorable outcome, with an average RMSE value of 8.0×10^{-6} . Following this, the proposed NF-RVFL-C and NF-RVFL-K models demonstrate competitive performance, registering 8.4×10^{-6} and 9.4×10^{-6} RMSE values, respectively. The standard RVFL model occupies the fourth position, whereas the standard ELM model showcases the worst performance, evidenced by the RMSE value of 5.8×10^{-2} . The RMSE value of the standard ELM is approximately 10000 times higher than the RVFL family's model. ELM can be seen as a variant of RVFL without direct links. In this context, ELM can be referred to as RVFLwoDL (RVFL without direct links). Thus, we conclude that the generalization performance of RVFL with direct links is better than without direct links.

Win-tie-loss for individual models is summarized in Table VII. In comparison with the ELM model, the proposed NF-RVFL-R and NF-RVFL-C models secure victories across all datasets, whereas NF-RVFL-K attains victory on 14 instances and experiences defeat in 1 out of a total of 15 datasets. This shows the dominance of the proposed NF-RVFL models over the existing baseline models. Thus, the emergence of the proposed NF-RVFL models introduces a promising avenue, demonstrating its potential to excel in addressing regression challenges effectively.

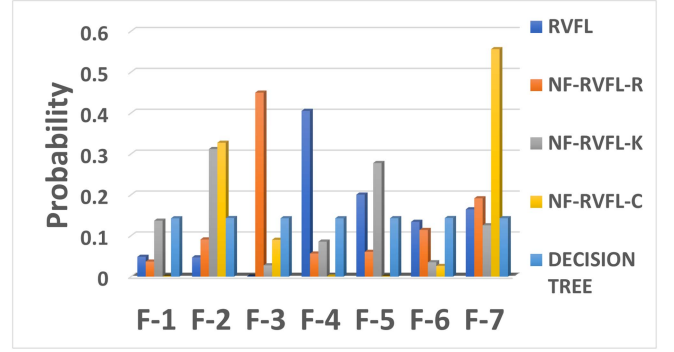


Fig. 3. Partial dependence plot (PDP) of each feature of the dataset pittsburg_bridges_T_OR_D w.r.t. the proposed NF-RVFL models, RVFL and decision tree. F-i in the X-axis denotes the i th feature of the dataset.

F. Interpretability of the Proposed NF-RVFL Models

Interpretability plays a pivotal role in system explanation, aiming to render the internal mechanisms of a system comprehensible to humans. As per [41], two primary categories of interpretability methods exist: intrinsic ante-hoc explanations, centered on transparent and straightforward models such as linear regression and decision trees, and post-hoc explanations, employing methods like PDP [26] and LIME [42] after model training. NF-RVFL, a network model, falls within the local post-hoc interpretability category, extracting fuzzy rules on the RVFL architecture for prediction. Due to the absence of a systematic definition and standardized evaluation methods for interpretability in machine learning, we adopt the established post-hoc method, i.e., PDP and also following the guidance of articles [24], [43], and [44], we assess the interpretability of the proposed NF-RVFL models. The interpretability of NF-RVFL is evaluated based on two factors: 1) PDP-based feature importance interpretability; and 2) feature-importance-based interpretability considering weights assigned to each feature during prediction (following [24], [43]).

1) **Partial dependence plot (PDP)-based feature importance interpretability:** To validate the interpretability of the proposed NF-RVFL models, we employed PDP [26], a standard technique for understanding the feature-prediction relationship. PDP enables the exploration of feature influence on projected outcomes and their importance according to the model. Considering decision trees as one of the best interpretable models [45], we compare the NF-RVFL and standard RVFL models with the decision tree. We depict PDP plots in Fig. 3 for all seven features of the dataset pittsburg_bridges_T_OR_D. For feature ranking, each feature is assigned ranks based on probability scores, with lower ranks indicating higher importance. In this context, NF-RVFL-C assigns rank 1 to feature-7 (F-7), rank 2 to feature-2 (F-2) and so on. Comparing rank errors (in RMSE) between the decision tree and NF-RVFL-R, NF-RVFL-K, NF-RVFL-C, and RVFL, the errors are 2.14, 2.62, 2.39, and 2.93, respectively. The lower rank error of the proposed NF-RVFL models with the decision tree (compared to the standard RVFL model) reveals that the proposed NF-RVFL models align more closely with the decision tree in terms of feature importance ranking. Consequently, this heightened congruence implies that the decision-making process within the proposed NF-RVFL models closely resembles that

of the decision tree (which is one of the best interpretable models). Hence, the interpretability of the proposed NF-RVFL models is enhanced to some extent in comparison to the RVFL model.

2) Feature importance-based interpretability considering weights assigned to each feature during prediction: This study presents the interpretability of the proposed NF-RVFL models using a feature-importance-based approach, wherein weights assigned to each feature during prediction are considered. The step-by-step learning process of the NF-RVFL models is demonstrated using the breast cancer dataset. Following [24], we consider the breast cancer dataset, which aims to predict the likelihood of breast cancer recurrence based on diagnostic measurements (features). Each data point in the dataset is categorized as either class 1 (no recurrence) or class 2 (recurrence). It consists of nine features labeled as x_1 to x_9 . For more information, visit the supplementary Section II-A for a comprehensive description of the dataset breast cancer. To provide a transparent illustration of interpretability, we first illustrate, in Section II-B of the supplementary materials, how our proposed NF-RVFL-K learns fuzzy rules and assigns weights to the original features during the training process at the minimum number of hidden nodes and fuzzy rules. Then, following [43], we show the interpretability of all the proposed NF-RVFL models at the best hyperparameter setting. The learning process of the NF-RVFL models on breast cancer is discussed in detail in supplementary Section II-C using Table IV. Analysis and main takeaways from the learning process and interpretability of NF-RVFL discussed in Section II of the supplementary material: In examining the interpretability of our proposed NF-RVFL models based on supplementary Table V, we analyze the following.

- 1) For NF-RVFL-C, feature emphasis is notably placed on features x_9 (Irradiat), x_6 (Deg Malig), and x_3 (tumor size). These features are associated with radiation therapy, cancer severity, and tumor size. As per biomedical knowledge [46], these are some prominent features in predicting whether breast cancer has no recurrence or recurrence. Align with biomedical knowledge; the proposed NF-RVFL-C gives significance to the above features in predicting breast cancer recurrence.
- 2) For NF-RVFL-R, the model predominantly prioritizes x_3 (tumor size). This concurs with biomedical understanding, highlighting the pivotal role of tumor size in predicting cancer diseases.
- 3) For NF-RVFL-K, the model assigns primary weightage to x_8 (breast-quad) and secondary importance to x_3 (tumor size). NF-RVFL-K adeptly identifies and prioritizes features, aligning with the final output and showcasing its ability to discern meaningful patterns.

In summary, our study demonstrates the learning rules and feature importance-based interpretability of the proposed NF-RVFL models. The models exhibit nuanced prioritization of features, aligning with biomedical knowledge and enhancing the understanding of their predictive capabilities in the context of breast cancer recurrence. In accordance with Friedman [26], “visualization stands as one of the most potent tools for interpretation.” Since the neuro-fuzzy IF-THEN rule is based on the humanlike visualization approach and from the above two

interpretability discussions, the proposed model gets closer to interpretability.

V. CONCLUSION

We proposed neuro-fuzzy RVFL (NF-RVFL) model by integrating RVFL with the neuro-fuzzy systems through a rigorous and comprehensive mathematical framework. The proposed NF-RVFL model enhanced the transparency of RVFL’s decision-making and predictions by embodying humanlike decision-making via an IF-THEN approach. The fuzzification step employed three distinct clustering methods—randomly initialized centers (R-means), K-means clustering centers, and fuzzy C-means clustering centers—resulting in diverse variations of the model (NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C, respectively).

The effectiveness of the proposed NF-RVFL models was demonstrated by rigorous experiments on 29 binary and 28 multiclass classification UCI datasets representing diverse domains and scales. A detailed investigation included benchmarking against nine existing baseline models from the RVFL, ELM, and BLS families. The statistical metrics confirm that the proposed NF-RVFL-C is statistically better than all the compared models. The proposed NF-RVFL-K and NF-RVFL-R also beat existing models (sometimes competitive performance) in terms of average accuracy and rank and secured the second and third positions, respectively. Lower standard deviations were consistently exhibited by the proposed NF-RVFL models compared to the baseline models. In the RVFL family, regression research was limited. We addressed this gap by employing our proposed NF-RVFL models for the regression domain and conducted experiments on 15 UCI benchmark regression datasets. The NF-RVFL models consistently showcased lower RMSE values and maximum wins compared to the existing models.

The interpretability of NF-RVFL was evaluated based on two factors: 1) PDP-based feature importance interpretability and 2) feature-importance-based interpretability considering weights assigned to each feature during prediction. The learning process and the feature importance of the proposed NF-RVFL models were studied based on feature-importance interpretability. It was found that the interpretability of the proposed NF-RVFL models was improved compared to the standard RVFL model. Moreover, NF-RVFL demonstrated superior interpretability compared to fuzzy-BLS due to its reduced complexity and improved structural adaptability, particularly in the utilization of only less fuzzy rules. Furthermore, we conducted a detailed ablation study (with and without key components) on the proposed NF-RVFL models, examining their dependence on key components. The essential elements under investigation included the direct links, fuzzification (fuzzy layer), defuzzification (defuzzified layer), number of fuzzy rules (clusters), and activation functions.

Future endeavors include the development of a neuro-fuzzy-based RVFL model that incorporates simultaneous feature selection and rule extraction. Additionally, exploring the integration of neuron models inspired by biological systems into the NF-RVFL model stands as a promising research avenue to enhance interpretability. This study primarily centered on shallow RVFL, constrained in capturing intricate data

relationships. Subsequent plans include extending this research to encompass deep and ensemble RVFL variants.

ACKNOWLEDGMENT

Open access funding provided by Qatar National Library.

REFERENCES

- [1] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Appl. Sci.*, vol. 12, no. 18, 2022, Art. no. 8972.
- [2] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2018.
- [3] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.
- [4] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Appl. Soft Comput.*, vol. 90, 2020, Art. no. 106181.
- [5] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feed forward neural networks with random weights," in *Proc. Int. Conf. Pattern Recognit., Vol. II. Conf. B: Pattern Recognit. Methodol. Syst.*, 1992, pp. 1–4.
- [6] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.
- [7] P. N. Suganthan, "On non-iterative learning algorithms with closed-form solution," *Appl. Soft Comput.*, vol. 70, pp. 1078–1082, 2018.
- [8] C. R. Rao and S. K. Mitra, "Generalized inverse of a matrix and its applications," in *Proc. 6th Berkeley Symp. Math. Statist. Probability, Volume 1: Theory Statist.*, 1972, pp. 601–621.
- [9] A. K. Malik, R. Gao, M. A. Ganaie, M. Tanveer, and P. N. Suganthan, "Random vector functional link network: Recent developments, applications, and future directions," *Appl. Soft Comput.*, vol. 143, 2023, Art. no. 110377.
- [10] L. Zhang and P. N. Suganthan, "A comprehensive evaluation of random vector functional link networks," *Inf. Sci.*, vol. 367, pp. 1094–1105, 2016.
- [11] N. Vuković, M. Petrović, and Z. Miljković, "A comprehensive experimental evaluation of orthogonal polynomial expanded random vector functional link neural networks for regression," *Appl. Soft Comput.*, vol. 70, pp. 1083–1096, 2018.
- [12] B. Igel'nik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.
- [13] D. Needell, A. A. Nelson, R. Saab, and P. Salanevich, "Random vector functional link networks for function approximation on manifolds," 2020, *arXiv:2007.15776*.
- [14] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Appl. Soft Comput.*, vol. 92, 2020, Art. no. 106275.
- [15] S. Feng, C. P. Chen, and C.-Y. Zhang, "A fuzzy deep model based on fuzzy restricted Boltzmann machines for high-dimensional data classification," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1344–1355, Jul. 2019.
- [16] S. Feng and C. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2020.
- [17] Y. Li, L. Liu, and G. Feng, "Adaptive finite-time controller design for T-S fuzzy systems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2425–2436, Sep. 2017.
- [18] G. Xue, Q. Chang, J. Wang, K. Zhang, and N. R. Pal, "An adaptive neuro-fuzzy system with integrated feature selection and rule extraction for high-dimensional classification problems," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 7, pp. 2167–2181, Jul. 2023.
- [19] G. Xue, J. Wang, B. Yuan, and C. Dai, "DG-ALETSK: A high-dimensional fuzzy approach with simultaneous feature selection and rule extraction," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 11, pp. 3866–3880, Nov. 2023.
- [20] A. M. AlRassas et al., "Advance artificial time series forecasting model for oil production using neuro fuzzy-based slime mould algorithm," *J. Petroleum Exploration Prod. Technol.*, vol. 12, pp. 1–13, 2022.
- [21] A. Vlasenko, N. Vlasenko, O. Vynokurova, Y. Bodyanskiy, and D. Peshko, "A novel ensemble neuro-fuzzy model for financial time series forecasting," *Data*, vol. 4, no. 3, 2019, Art. no. 126.
- [22] S. Han, K. Zhu, M. Zhou, and X. Liu, "Evolutionary weighted broad learning and its application to fault diagnosis in self-organizing cellular networks," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3035–3047, May 2023.
- [23] A. T. Dosdoğru, "Improving weather forecasting using de-noising with maximal overlap discrete wavelet transform and ga based neuro-fuzzy controller," *Int. J. Artif. Intell. Tools*, vol. 28, no. 3, 2019, Art. no. 1950012.
- [24] H. Zhao, Y. Wu, and W. Deng, "An interpretable dynamic inference system based on fuzzy broad learning," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 2527415.
- [25] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: Survey in soft computing framework," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 748–768, May 2000.
- [26] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, 2001.
- [27] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
- [28] C. Ren, S. He, X. Luan, F. Liu, and H. R. Karimi, "Finite-time L_2 -gain asynchronous control for continuous-time positive hidden Markov jump systems via T-S fuzzy model approach," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 77–87, Jan. 2021.
- [29] R. Tabbussum and A. Q. Dar, "Performance evaluation of artificial intelligence paradigms—Artificial neural networks, fuzzy logic, and adaptive neuro-fuzzy inference system for flood prediction," *Environ. Sci. Pollut. Res.*, vol. 28, no. 20, pp. 25265–25282, 2021.
- [30] H. Oufiak and A. Idri, "On the performance and interpretability of Mamdani and Takagi-Sugeno-Kang based neuro-fuzzy systems for medical diagnosis," *Sci. Afr.*, vol. 20, 2023, Art. no. e01610.
- [31] S. Ghosh and S. K. Dubey, "Comparative analysis of K-means and fuzzy C-means algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 4, pp. 35–39, 2013.
- [32] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [33] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, and Alzheimer's Disease Neuroimaging Initiative Initiative, "Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network," *IEEE Trans. Comput. Social Syst.*, to be published, doi: [10.1109/TCSS.2022.3146974](https://doi.org/10.1109/TCSS.2022.3146974).
- [34] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 311–324, Jan. 2016.
- [35] M. A. Ganaie, M. Tanveer, and P. N. Suganthan, "Minimum variance embedded random vector functional link network," in *Proc. 27th Int. Conf. Neural Inf. Process.*, Bangkok, Thailand, 2020, pp. 412–419.
- [36] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum class variance extreme learning machine for human action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 11, pp. 1968–1979, Nov. 2013.
- [37] C. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- [38] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
- [39] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
- [40] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the Fbrietkan statistic," *Commun. Statist.-Theory Methods*, vol. 9, no. 6, pp. 571–595, 1980.
- [41] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2nd edn., 2020. [Online]. Available: <https://christophm.github.io/interpretable-ml-book>
- [42] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.
- [43] R. Xie, C.-M. Vong, and S. Wang, "Doubly interpretable fuzzy apriori classifier by successive stacking and one-step wide calculation," *IEEE Trans. Fuzzy Syst.*, to be published, doi: [10.1109/TFUZZ.2023.3330883](https://doi.org/10.1109/TFUZZ.2023.3330883).
- [44] D. A. Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7786–7795.
- [45] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019, Art. no. 832.
- [46] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1455–1462, Jul. 2016.