Robust Dual-Model Collaborative Random Vector Functional Link Network

A. Quadir
Department of Mathematics
Indian Institute of Technology Indore
mscphd2207141002@iiti.ac.in

A. Rahaman
Department of Mathematics
Indian Institute of Technology Indore
phd2401141001@iiti.ac.in

Mushir Akhtar
Department of Mathematics
Indian Institute of Technology Indore
phd2101241004@iiti.ac.in

M. Tanveer*
Department of Mathematics
Indian Institute of Technology Indore
mtanveer@iiti.ac.in

***Abstract*—Random vector functional link (RVFL) networks are lightweight and fast neural models that offer efficient training and strong generalization through randomized hidden-layer weights and direct input-output connections. However, conventional RVFL models are sensitive to noisy labels, outliers, and imbalanced data, which limits their performance in real-world applications. To address these challenges, we propose the kernel risk-sensitive mean p-power based RVFL (KRPRVFL) model, which integrates the computational efficiency of RVFL with the robustness of the kernel risk-sensitive mean p-power (KRP) criterion. By replacing the standard least-squares objective with a KRP-based loss, KRPRVFL adaptively reduces the influence of corrupted or unreliable samples during training, resulting in improved stability and generalization. Additionally, a collaborative learning mechanism is introduced to enable adaptive interaction among model components, further enhancing robustness in complex and noisy environments. The proposed framework also leverages kernel-induced feature mapping to capture nonlinear relationships without requiring explicit hidden-layer selection, maintaining both efficiency and scalability. Extensive experiments on UCI and KEEL benchmark datasets demonstrate that KRPRVFL consistently outperforms baseline models in terms of accuracy, robustness, and statistical significance, highlighting its effectiveness as a fast, scalable, and reliable solution for challenging classification tasks. The code and supplementary material of the paper can be accessed using the following link: https://github.com/mtanveer1/KRPRVFL.**

***Index Terms*—Kernel risk-sensitive mean p-power (KRP) criterion, Random vector functional link (RVFL) network, dual-model collaborative, label noise.**

I. Introduction

In recent years, deep learning has emerged as a dominant paradigm in artificial intelligence and has achieved remarkable success across a wide range of applications [1, 2]. Despite its strong representation and learning ability, deep learning models typically rely on complex architectures and involve a large number of hyperparameters, which makes the training process computationally expensive and time-consuming. To address these limitations, Pao et al. [3] introduced the random vector functional link (RVFL) network, which offers a lightweight alternative with a simple architecture and a reduced set of tunable parameters, enabling rapid learning without the need for iterative parameter updates [4]. RVFL is a shallow feed-forward randomized neural network in which the hidden-layer weights are randomly generated and kept fixed during training. A distinctive characteristic of RVFL is the presence of direct connections between the input and output layers [5, 6]. These shortcut connections provide an implicit form of regularization, thereby enhancing learning stability and improving generalization performance compared to conventional randomized neural networks [7, 8].

To further enhance the generalization ability of the standard RVFL framework, a number of improved variants have been proposed, aiming to increase robustness and practical effectiveness [9]. RVFL treats all training samples uniformly, which can make the model vulnerable to the presence of noise and outliers [10]. To mitigate this drawback, an intuitionistic fuzzy RVFL (IFRVFL) model is introduced in [11], where fuzzy membership and non-membership functions are employed to assign intuitionistic fuzzy scores to individual samples, thereby reducing the influence of unreliable data. Moreover, in conventional RVFL, the input features are mapped into a randomized feature space, which may introduce instability in the learned representation. To alleviate this issue, Zhang et al. [12] integrated a sparse autoencoder with ℓ_1-norm regularization into the RVFL framework, leading to the development of the SP-RVFL model. By enforcing sparsity in the learned representations, this approach reduces the adverse effects introduced by random feature mapping and promotes more stable and informative feature extraction. Recently, complex-valued extensions of RVFL have been introduced to address limitations of real-valued models in complex signal processing [13]. In particular, CRVFL and its augmented variants effectively exploit complex-valued statistics and correlations between real and imaginary components, achieving improved performance and computational efficiency in complex-valued learning tasks while preserving the fast training advantage of RVFL [14]. Although these extended RVFL variants enhance

*Corresponding author

robustness to a certain extent, they remain inadequate for effectively addressing classification tasks involving heavily contaminated or noisy data commonly encountered in real-world applications.

The kernel risk-sensitive mean (p)-power (KRP) criterion is effective in reproducing kernel Hilbert spaces (RKHS) for robust learning [15]. By combining kernel-based nonlinear mapping with a risk-sensitive formulation, it improves resilience to noise, outliers, and data imbalance [16]. Initially developed for recursive kernel adaptive filtering and later extended to complex-valued learning [17], it has broad applicability. Additionally, a kernel-based RVFL model (KERVFL) avoids explicit hidden-layer size selection [18]. However, kernel methods suffer from high computational cost and scalability issues, and integrating KRP into RVFL remains underexplored.

Motivated by the limitations of conventional RVFL networks in handling noisy, imbalanced, or corrupted data, we propose the kernel risk-sensitive mean p-power based RVFL (KRPRVFL) model, which integrates the efficiency of RVFL with the robustness of the KRP criterion. Unlike standard RVFL, which relies on a least-squares objective and treats all samples equally, KRPRVFL employs a risk-sensitive KRP-based loss function to adaptively suppress the influence of outliers and mislabeled samples during training. This ensures that the model focuses on the underlying data distribution, resulting in improved generalization and stability in noisy environments. Furthermore, KRPRVFL incorporates a collaborative learning mechanism, enabling dynamic interaction among the model components to refine predictions and adapt to complex data patterns. By leveraging kernel-induced feature mapping, KRPRVFL captures nonlinear relationships in the input space without requiring manual selection of hidden-layer size or complex iterative training procedures. The proposed KRPRVFL model provides a fast, scalable, and robust framework for classification tasks, effectively combining the computational simplicity of RVFL with the resilience and adaptability of kernel-based risk-sensitive learning.

The key highlights of this paper can be encapsulated as follows:

1) The proposed KRPRVFL model combines the fast and lightweight architecture of RVFL networks with the robust kernel risk-sensitive mean p-power (KRP) criterion, enabling effective handling of noisy, corrupted, and imbalanced data.
2) By replacing the conventional least-squares objective with a KRP-based loss, the model adaptively suppresses the influence of outliers and mislabeled samples, improving generalization and stability under challenging learning conditions.
3) KRPRVFL uses collaborative learning and kernel mapping to capture nonlinear relationships, improving accuracy and robustness without needing to set hidden-layer size.
4) Extensive experiments on UCI and KEEL benchmark datasets demonstrate that the proposed KRPRVFL consistently outperforms baseline models in terms of accuracy and statistical significance.

II. Related work

In this section, we first establish the notations used throughout the paper and then provide an overview of the RVFL model.

A. Notations

Let the training dataset be represented as $\mathcal{X} = \{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$, where $x_i \in \mathbb{R}^{1\times m}$ denotes the input feature vector and $y_i \in \{+1, -1\}$ is the corresponding target label. n is the total number of training samples and m is the number of features. The transpose operator is denoted by $(\cdot)^T$. The matrices of all input and output samples are defined as $X = [x_1^T, x_2^T, \ldots, x_n^T]^T$ and $Y = [y_1^T, y_2^T, \ldots, y_n^T]^T$, respectively.

B. Random Vector Functional Link (RVFL) Network

The RVFL network is a single-layer feedforward network with random, fixed input-to-hidden weights and shortcut connections from input to output, enhancing generalization. The hidden-layer output matrix $H_1 \in \mathbb{R}^{n\times N}$, with N hidden nodes, is computed as:

$$H_1 = \phi(XW_1 + b_1), \tag{1}$$

where $W_1 \in \mathbb{R}^{m\times N}$ is a randomly initialized weight matrix, $b_1 \in \mathbb{R}^{n\times N}$ is the bias matrix, and ϕ is the activation function.

The combined feature matrix H_2, which concatenates the input and hidden-layer outputs, is defined as $H_2 = [X; H_1]$. The predicted output $\hat{Y}$ is then given by:

$$H_2\beta = \hat{Y}, \tag{2}$$

where $\beta \in \mathbb{R}^{(m+N)\times 1}$ denotes the output weight matrix. Training the RVFL involves solving the following regularized least-squares optimization problem:

$$\beta_{\min} = \arg\min_{\beta} \frac{\mathcal{C}}{2}\|H_2\beta - Y\|^2 + \frac{1}{2}\|\beta\|^2, \tag{3}$$

where $\mathcal{C} > 0$ is a regularization parameter.

The closed-form solution for β depends on the dimensions of H_2 relative to the number of samples n:

$$\beta_{\min} = \begin{cases} H_2^T\left(H_2H_2^T + \frac{1}{\mathcal{C}}I\right)^{-1}Y, & n < m+N, \\ \left(H_2^TH_2 + \frac{1}{\mathcal{C}}I\right)^{-1}H_2^TY, & n \geq m+N, \end{cases} \tag{4}$$

where I is an identity matrix of appropriate size.

III. The Proposed Robust Dual-Model Collaborative Random Vector Functional Link Network

In the presence of label noise, the output weights learned by conventional RVFL networks become highly sensitive to corrupted labels, leading to degraded robustness and generalization performance. To address this issue, kernel risk-sensitive mean p-power based RVFL (KRPRVFL) is proposed, in which the output weights are learned using the KRP criterion. By replacing the conventional least-squares objective, KRP-RVFL

effectively suppresses the influence of mislabeled samples and achieves robust learning under label noise environments.

The KRP metric, defined in RKHS, emphasizes deviations to reduce the impact of outliers, enhancing RVFL robustness in noisy environments. In kernel-induced feature spaces, higher-order statistical relationships in the original input space can be equivalently represented using second-order statistics. Let α and β denote two random variables. Their dependency can be quantified in RKHS via a kernel-based correlation measure [19, 20], which is expressed as:

$$\mathcal{V}(\alpha,\beta) = \mathbb{E}\left[\langle\phi(\alpha),\phi(\beta)\rangle_{\mathcal{H}}\right] = \int \langle\phi(\alpha),\phi(\beta)\rangle_{\mathcal{H}}, dF_{\alpha,\beta}(\alpha,\beta), \tag{5}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator and $F_{\alpha,\beta}(\alpha,\beta)$ represents the joint probability distribution of α and β. The mapping $\phi(\cdot)$ is implicitly defined through a Mercer kernel $\kappa_\sigma(\cdot)$, which projects data from the input space into an RKHS $\mathcal{H}$ equipped with the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. This mapping satisfies:

$$\langle\phi(\alpha),\phi(\beta)\rangle_{\mathcal{H}} = \kappa_\sigma(\alpha,\beta). \tag{6}$$

Based on the above formulation, the KRP criterion is constructed to quantify the similarity between random variables in RKHS while incorporating risk sensitivity, which forms the foundation for robust output-weight learning in the proposed KRPRVFL framework. The KRP criterion is defined as follows:

$$\begin{aligned}\mathcal{L}_{\mu,p,\sigma}(\alpha,\beta) &= \frac{1}{\mu}\mathbb{E}\left[\exp\left(2^{-\frac{1}{p}}\mu\|\phi(\alpha)-\phi(\beta)\|_{\mathcal{H}}^{p}\right)\right] \\ &= \frac{1}{\mu}\mathbb{E}\left[\exp\left(2^{-\frac{1}{p}}\mu\left(\|\phi(\alpha)-\phi(\beta)\|_{\mathcal{H}}^{p}\right)^{\frac{p}{2}}\right)\right] \\ &= \frac{1}{\mu}\int \exp\left(\mu(1-\kappa_\sigma(\alpha,\beta))^{\frac{p}{2}}\right) dF_{\alpha,\beta}(\alpha,\beta),\end{aligned} \tag{7}$$

where α and β denote arbitrary random variables. The parameter $\mu > 0$ regulates the degree of risk sensitivity in the loss function, while $p > 0$ determines the order of the deviation penalty. Throughout this work, the kernel mapping is instantiated using a Gaussian Mercer kernel with a bandwidth parameter $\sigma > 0$, defined as:

$$\kappa_\sigma(\alpha,\beta) = \exp\left(-\frac{(\alpha-\beta)^2}{2\sigma^2}\right). \tag{8}$$

The joint probability distribution of (α,β) is generally inaccessible. Instead, one only observes a finite set of n paired samples $(\alpha_i,\beta_i)_{i=1}^n$. Using the empirical approximation, the KRP criterion is formulated as:

$$\mathcal{L}_{\mu,p,\sigma}(\alpha,\beta) = \frac{1}{n\mu}\sum_{i=1}^{n}\exp\left(\mu\left(1-\kappa_\sigma(\alpha_i,\beta_i)\right)^{\frac{p}{2}}\right). \tag{9}$$

From this perspective, the KRP criterion can be interpreted as a similarity evaluation mechanism between two sample sequences $[\alpha_1,\alpha_2,\ldots,\alpha_n]$ and $[\beta_1,\beta_2,\ldots,\beta_n]$ in the kernel-induced feature space. The learning task aims to minimize the empirical KRP loss between the true label matrix Y and the RVFL output, which can be expressed as:

$$\mathcal{L}_{\mu,p,\sigma}(Y,HW) = \arg\min_{W}\sum_{i=1}^{n}\exp(\mu(1-\kappa_\sigma(y_i,\hat{y}_i))^{\frac{p}{2}}), \tag{10}$$

where y_i and $\hat{y}_i$ denote the true label and the predicted output of the i^{th} sample x_i, respectively. The predicted output $\hat{y}_i$ is obtained from the RVFL model as:

$$\hat{y}_i = h_i W, \tag{11}$$

where $h_i \in \mathbb{R}^N$ represents the i^{th} row of the RVFL hidden-layer feature matrix and W denotes the output weight matrix.

To further control model complexity and prevent overfitting, a regularization term is incorporated into the objective function. Consequently, the final optimization problem of the proposed KRPRVFL model is formulated as:

$$\arg\min_{W}\left(\mathcal{D}\|W\|^2 + \frac{1}{n\mu}\sum_{i=1}^{n}\left(\mu(1-\kappa_\sigma(y_i,h_iW))^{\frac{p}{2}}\right)\right). \tag{12}$$

To simplify the subsequent derivations, the overall objective function is denoted by $\psi(W)$, defined as

$$\psi(W) = \mathcal{D}\|W\|^2 + \frac{1}{n\mu}\sum_{i=1}^{n}\exp\left(\mu\left(1-\kappa_\sigma(y_i,h_iW)\right)^{[p/2]}\right). \tag{13}$$

For notational convenience, we further introduce the auxiliary variable $\upsilon_i = 1-\kappa_\sigma(y_i,h_iW)$, which characterizes the kernel-based deviation between the true label and the predicted output of the i^{th} sample. Based on this definition, the gradient of $\psi(W)$ with respect to the output weight matrix W can be derived as follows:

$$\frac{\partial\psi(W)}{\partial W} = 2\mathcal{D}W - \frac{p}{2n\sigma^2}H^T\Omega(Y-HW), \tag{14}$$

where $\Omega \in \mathbb{R}^{n\times n}$ denotes a diagonal weighting matrix defined in Eqs. (15) and (16). The elements on its diagonal are determined by the contribution of each training sample to the overall loss function, thereby reflecting the relative influence of individual samples during the optimization process. Specifically, the i^{th} diagonal entry is computed as:

$$\Omega_{ii} = \kappa_\sigma(y_i,h_iW)\left(\upsilon_i\right)^{\frac{p-2}{2}}\exp\left(\mu\left(\upsilon_i\right)^{\frac{p}{2}}\right), \tag{15}$$

and

$$\Omega = \begin{bmatrix} \kappa_\sigma(y_1,h_1W)\cdot\left(\upsilon_1\right)^{\frac{p-2}{2}} & & \\ \exp\left(\mu(\upsilon_1)^{\frac{p}{2}}\right) & & \\ & \ddots & \\ & & \kappa_\sigma(y_N,h_NW)\cdot\left(\upsilon_n\right)^{\frac{p-2}{2}} \\ & & \cdot\exp\left(\mu(\upsilon_n)^{\frac{p}{2}}\right) \end{bmatrix}. \tag{16}$$

By equating the gradient in (14) to zero, a closed-form solution

for the output weight matrix W can be derived as:

$$W = (\mu' I + H^T \Omega H)^{-1} H^T \Omega Y, \tag{17}$$

where $\mu' = \frac{4\mathcal{D}n\sigma^2}{p}$ serves as the regularization-related scalar. To improve efficiency, an equivalent form of 17 is derived using matrix inversion identities, given by

$$W = H^T(\mu' I + \Omega H H^T)^{-1} \Omega Y. \tag{18}$$

The preferred formulation depends on the sample size n and feature dimension $N + L$: use (17) if $n > L + N$, and ((18)) if $n < L + N$. The update rule in (18) depends on the current W, allowing it to be viewed as a fixed-point mapping:

$$W = \mathcal{F}(W, H, Y) = (\mu' I + H^T \Omega H)^{-1} H^T \Omega Y. \tag{19}$$

The output weight matrix at the (t^{th} iteration is obtained by applying the mapping $\mathcal{F}(\cdot)$ to the estimate from the previous iteration, which yields:

$$W_t = \mathcal{F}(W_{t-1}, H, Y). \tag{20}$$

The complete procedural steps for the KRPRVFL model are summarized in Algorithm 1.

Algorithm 1 KRPRVFL Algorithm

1: **Input:** Training data X with labels Y; regularization parameters $\mathcal{D}$; kernel bandwidth σ; convergence threshold τ; maximum number of iterations T; risk-sensitivity parameter μ; power parameter p.
2: **Output:** Output weights W.
3: Generate the hidden-layer feature matrix $\mathcal{B} = \gamma(X\mathcal{O} + b)$, where $\mathcal{O}$ is the randomly initialized weights matrix and b is the bias vector, and γ is an activation function.
4: Find the enhanced features using $H = [X\ \mathcal{B}]$.
5: **for** t = 1, 2, …, T **do**
6: Compute the sample-adaptive weighting matrix using Eqs. (15) and (16).
7: Update the output weight matrix W using Eq. (17) or Eq. (18).
8: **if** $\|W_t - W_{t-1}\|^2 < \tau$ **then**
9: **break**
10: **end if**
11: **end for**

IV. Experimental Results

To validate the performance of the proposed KRPRVFL model, extensive experiments are carried out on a collection of widely used benchmark datasets obtained from the UCI [22] and KEEL [23] repositories. The proposed approach is thoroughly evaluated through comparative studies with several representative learning models, including RVFL [3], ELM [21], GB-RVFL [4], GE-GB-RVFL [4], CRVFL [13], and ACRVFL [13], to assess its effectiveness comprehensively. Experiments with added label noise are detailed in Section S.I of the supplementary material. Sensitivity analyses of the proposed models are presented in Section S.II of the supplementary material.

A. Experimental Setup

Experiments are run on a Windows 11 workstation with an Intel Xeon Gold 6226R (2.90 GHz) and 256GB RAM using Python 3.11. Datasets are split 70:30 for training and testing, with hyperparameters tuned via grid search and five-fold cross-validation. The regularization coefficients are explored over the set $\mathcal{D} = \{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. The risk-sensitive parameter $\mu \in [1, 10]$ and the power parameter $p \in [20, 21, ..., 210]$. Hidden nodes N range from 3 to 203, and nine activation functions are tested: SELU, ReLU, Sigmoid, Sine, Hardlim, Tribas, Radbas, Sign, and Leaky ReLU.

B. Evaluation on UCI and KEEL Datasets

This section presents a comprehensive experimental study evaluating the proposed KRPRVFL model against multiple baseline models on 37 benchmark datasets from the UCI and KEEL repositories. Performance is assessed using classification accuracy (Acc), with results reported in Table I. The proposed KRPRVFL model achieves an average Acc of 85.20%, while the baseline RVFL, ELM, GB-RVFL, GE-GB-RVFL, CRVFL, and ACRVFL models obtained an average Acc of 81.55%, 80.81%, 79.62%, 78.17%, 75.86%, and 76.62%, respectively. In terms of average Acc, the proposed KRPRVFL model consistently outperforms competing models, indicating strong predictive performance. However, average Acc alone may mask variability across datasets, as high performance on some can offset weaker results on others. To address this issue and to rigorously examine whether the observed performance variations are statistically meaningful, a series of nonparametric statistical tests is employed in accordance with the recommendations of Demšar [24]. These statistical methods are suitable for comparing multiple classification algorithms across diverse datasets, particularly when parametric test assumptions are violated. Accordingly, nonparametric techniques such as ranking-based analysis, the Friedman test, and the Nemenyi post hoc test are used. In ranking-based evaluation, each model is ranked based on its performance on individual datasets, and the ranks are aggregated to determine overall performance. Under this ranking strategy, models with inferior performance are given larger rank scores, while more effective models obtain smaller ranks. This mechanism captures performance trade-offs across datasets, ensuring that strong results on some datasets can compensate for weaker outcomes on others. Consider the evaluation of g models over n datasets, where $\mathcal{R}_i^j$ denotes the rank assigned to the j^{th} model on the i^{th} dataset. The overall performance of the j^{th} model is given by its mean rank, which is obtained by averaging its ranks across all datasets, i.e., $\mathcal{R}^j = \frac{1}{n}\sum_{i=1}^{n} \mathcal{R}_i^j$. The proposed KRPRVFL model achieves an average rank of 2.03, while the corresponding average ranks for the baseline RVFL, ELM, GB-RVFL, GE-GB-RVFL, CRVFL, and ACRVFL models are 3.88, 4.34, 4.07, 4.38, 4.95, and 4.36, respectively. Among all evaluated models, the proposed KRPRVFL attains the lowest average rank. Given that a lower rank indicates better performance, this result confirms that the proposed KRPRVFL model outperforms the baseline

TABLE I: Performance comparison of the proposed KRPRVFL model along with the baseline models for UCI and KEEL datasets.

Dataset	RVFL [3]	ELM [21]	GB-RVFL [4]	GE-GB-RVFL [4]	CRVFL [13]	ACRVFL [13]	KRPRVFL†
bank	89.17	87.31	88.43	88.95	88.28	88.8	89.31
blood	76.44	72.14	76	77.33	75.56	77.33	77.33
breast_cancer	72.09	74.42	77.91	65.12	79.07	74.42	74.42
breast_cancer_wisc_prog	75	73.33	73.33	73.33	66.67	66.67	63.33
bupa or liver-disorders.csv	65.38	65.38	54.81	56.73	94.95	77.24	64.42
checkerboard_Data.csv	85.94	85.98	87.02	87.02	62.48	71.5	87.02
chess_krvkp	90.41	90.2	91.45	89.16	93.95	94.06	97.81
cleve.csv	81.11	80	75.56	82.22	73.79	79.79	84.44
conn_bench_sonar_mines_rocks	74.6	71.43	74.6	68.25	65.08	66.67	84.13
credit_approval	84.62	82.3	77.88	80.77	83.65	84.13	83.17
crossplane150.csv	81.11	81.11	86.67	73.33	84.68	73.79	95.56
cylinder_bands	72.73	70.67	59.74	64.29	70.78	66.23	74.03
ecoli-0-1-4-6_vs_5.csv	98.81	98.81	98.81	98.81	87.14	86.74	95.83
ecoli-0-1-4-7_vs_5-6.csv	90	96	94	94	72.9	70.66	96
fertility	90	80	86.67	90	90	90	90
haber.csv	76.09	78.26	78.26	77.17	75.46	85.85	78.26
haberman.csv	76.09	76.09	78.26	77.17	71.58	83.33	78.26
haberman_survival	76.09	78.26	78.26	77.17	82.61	82.61	77.17
heart_hungarian	72.78	72.65	74.16	74.16	75.28	76.4	78.65
hepatitis	72.34	70.11	72.34	72.34	76.6	82.98	82.98
hill_valley	68.41	67.31	59.07	60.71	53.3	54.12	72.25
horse_colic	83.78	80.18	72.97	76.58	76.58	77.48	85.59
ionosphere	82.79	82.45	83.96	83.02	85.85	83.96	87.74
monks_3	90.41	90.41	85.63	91.02	74.85	85.03	95.81
new-thyroid1.csv	86	86	90.77	96.92	73.15	67.62	100
oocytes_merluccius_nucleus_4d	80.71	79.74	80.78	78.5	67.43	77.2	84.69
oocytes_trisopterus_nucleus_2f	80.12	80.94	75.55	80.29	69.71	71.9	83.21
statlog_australian_credit	68.75	68.75	62.02	62.02	70.19	70.19	69.71
statlog_german_credit	70.33	70.67	70	75.33	69	66.33	76.67
vehicle2.csv	90.85	90.03	91.73	92.52	90.36	85.9	98.43
vertebral_column_2clases	81.4	72.25	74.19	74.19	66.67	77.42	88.17
votes.csv	90.47	90.47	92.37	87.02	67.22	67.78	96.18
vowel.csv	99.33	98.99	85.19	63.97	70.71	67.03	99.66
wpbc.csv	69.49	70.97	76.27	69.49	75.69	84.96	81.36
yeast-0-2-5-6_vs_3-7-8-9.csv	89.71	89.71	93.38	90.73	77.16	67.02	93.05
yeast-0-2-5-7-9_vs_3-6-8.csv	95.35	95.68	98.01	97.35	78.44	70.71	98.34
yeast-0-3-5-9_vs_7-8.csv	88.82	90.79	69.74	45.39	70.01	81.01	89.47
Average Acc	<u>81.55</u>	80.81	79.62	78.17	75.86	76.62	**85.2**
Average Rank	3.88	4.34	4.07	4.38	4.95	4.36	2.03

The proposed model is denoted by †.
The top and second-best models in terms of Acc are denoted by boldface and underline, respectively.

models. The Friedman test [25] is a nonparametric statistical test designed to examine whether meaningful performance differences exist among multiple models by evaluating their average rankings over a set of datasets. It offers a structured framework for comparing several models in multi-dataset experiments. The null hypothesis assumes that all models exhibit equivalent performance, implying that their average ranks are identical. The Friedman test statistic is calculated using a chi-square measure, denoted as χ_F^2, which follows a chi-squared distribution with $g-1$ degrees of freedom, and is defined as: $\chi_F^2 = \frac{12n}{g(g+1)}\left[\sum_j \mathcal{R}_j^2 - \frac{g(g+1)^2}{4}\right]$. The Friedman statistic can be transformed into the F_F statistic, given by $F_F = \frac{(n-1)\chi_F^2}{n(g-1)-\chi_F^2}$, which follows an F-distribution with $(g-1)$ and $(n-1)(g-1)$ degrees of freedom. For $n = 37$ datasets and $g = 7$ competing models, the Friedman analysis produces a test statistic of $\chi_F^2 = 41.807$, and the F_F value is 8.35. At the 5% significance level, the corresponding critical value from the F-distribution for $F_F(6, 216)$ is 2.1407. As the obtained F_F value is substantially larger than this threshold, the null hypothesis of equal performance is rejected, confirming the presence of statistically significant performance differences among the evaluated models. Now, the Nemenyi post hoc test is employed to analyze pairwise performance disparities between the competing models. The critical difference (C.D.) is determined as: C.D. $= q_\alpha\sqrt{\frac{g(g+1)}{6n}}$, where q_α denotes the critical value obtained from the two-sided Nemenyi distribution table. Based on the F-distribution table, the critical value q_α at a 5% significance level is 2.949, which corresponds to a computed C.D. of 1.4811. The average rank differences between KRPRVFL and RVFL, ELM, GB-RVFL, GE-GB-RVFL, CRVFL, and ACRVFL are 1.85, 2.31, 2.04, 2.35, 2.92, and 2.33, respectively. The Nemenyi test shows KRPRVFL achieves statistically significant improvements over all baselines.

V. Conclusion

In this paper, we introduced the KRPRVFL model, a robust extension of the random vector functional link network that integrates the kernel risk-sensitive mean p-power criterion with a collaborative learning mechanism. Our experiments on UCI and KEEL benchmark datasets demonstrated that KRPRVFL consistently achieves superior classification performance compared to existing RVFL variants and baseline models, particularly in the presence of noisy labels, outliers, and imbalanced data. The results highlight the effectiveness of

risk-sensitive learning in improving robustness and the value of collaborative feature interaction for enhancing adaptability in complex data scenarios. KRPRVFL is computationally efficient and scalable, but its performance depends on kernel and risk-sensitive parameters and is best suited for small to medium datasets. Future work will target large-scale and streaming data, adaptive kernel selection, multi-task and multi-view learning, and deep RVFL integration for hierarchical features.

REFERENCES

[1] R. Yao, H. Zhao, Z. Zhao, C. Guo, and W. Deng, "Parallel convolutional transfer network for bearing fault diagnosis under varying operation states," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024.

[2] A. Quadir and M. Tanveer, "Hypergraph neural network with state space models for node classification," *Engineering Applications of Artificial Intelligence*, vol. 163, p. 112922, 2026.

[3] 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.

[4] M. Sajid, A. Quadir, M. Tanveer, and Alzheimer's Disease Neuroimaging Initiative, "GB-RVFL: Fusion of randomized neural network and granular ball computing," *Pattern Recognition*, vol. 159, p. 111142, 2025.

[5] M. Akhtar, A. Kumari, M. Sajid, A. Quadir, M. Arshad, P. Suganthan, and M. Tanveer, "Towards robust and inversion-free randomized neural networks: The XG-RVFL framework," *Pattern Recognition*, p. 112711, 2025.

[6] L. Zhang and P. N. Suganthan, "A comprehensive evaluation of random vector functional link networks," *Information sciences*, vol. 367, pp. 1094–1105, 2016.

[7] A. Quadir and M. Tanveer, "Randomized based restricted kernel machine for hyperspectral image classification," *arXiv preprint arXiv:2503.05837*, 2025.

[8] M. Sajid, A. Quadir, and M. Tanveer, "Wave-RVFL: A randomized neural network based on wave loss function," in *International Conference on Neural Information Processing*. Springer, 2025, pp. 242–257.

[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," *Applied Soft Computing*, vol. 143, p. 110377, 2023.

[10] A. Quadir, M. Sajid, and M. Tanveer, "Multiview random vector functional link network for predicting DNA-binding proteins," *arXiv preprint arXiv:2409.02588*, 2024.

[11] A. K. Malik, M. A. Ganaie, M. Tanveer, and P. N. Suganthan, "Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4754–4765, 2022.

[12] Y. Zhang, J. Wu, Z. Cai, B. Du, and P. S. Yu, "An unsupervised parameter learning model for RVFL neural network," *Neural Networks*, vol. 112, pp. 85–97, 2019.

[13] C. Liu, H. Zhang, L. Chen, and F. Li, "Complex-valued random vector functional link neural network based on real augmented representation and its applications," *Applied Soft Computing*, vol. 170, p. 112682, 2025.

[14] M. Sajid, M. Akhtar, A. Quadir, and M. Tanveer, "RVFL-X: A novel randomized network based on complex transformed real-valued tabular datasets," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[15] S. Perla, R. Bisoi, P. Dash, and A. Rout, "Short-term forecasting of electricity price using ensemble deep kernel based random vector functional link network," *Applied Soft Computing*, vol. 174, p. 113012, 2025.

[16] T. Zhang, S. Wang, H. Zhang, K. Xiong, and L. Wang, "Kernel risk-sensitive mean p-power error algorithms for robust learning," *Entropy*, vol. 21, no. 6, p. 588, 2019.

[17] G. Huang, M. Shen, T. Zhang, F. He, and S. Wang, "Complex multi-kernel random fourier adaptive algorithms under the complex kernel risk-sensitive p-power loss," *Digital Signal Processing*, vol. 115, p. 103087, 2021.

[18] T. Chakravorti and P. Satyanarayana, "Non linear system identification using kernel based exponentially extended random vector functional link network," *Applied Soft Computing*, vol. 89, p. 106117, 2020.

[19] L. Yi, R. Min, C. Kunjie, L. Dan, Z. Ziqiang, L. Fan, and Y. Bo, "Identifying and managing risks of AI-driven operations: A case study of automatic speech recognition for improving air traffic safety," *Chinese Journal of Aeronautics*, vol. 36, no. 4, pp. 366–386, 2023.

[20] C. Zhang, L. Gao, X. Li, W. Shen, J. Zhou, and K. C. Tan, "Resetting weight vectors in MOEA/D for multiobjective optimization problems with discontinuous pareto front," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9770–9783, 2021.

[21] 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.

[22] D. Dua and C. Graff, "UCI machine learning repository." *Available: http://archive.ics.uci.edu/ml*, 2017.

[23] J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult. Valued Log. Soft Comput*, vol. 17, pp. 255–287, 2015.

[24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[25] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.