Learning with Randomized Neural Networks for Classification Problems

Ph.D. Thesis

By
Ashwani Kumar Malik

under the supervision of
Dr. M. Tanveer

Department of Mathematics
INDIAN INSTITUTE OF TECHNOLOGY INDORE

September 2023

Learning with Randomized Neural Networks for Classification Problems

A THESIS

submitted in partial fulfillment of the
requirements for the award of the degree
of
DOCTOR OF PHILOSOPHY

By
Ashwani Kumar Malik

under the supervision of
Dr. M. Tanveer

Department of Mathematics
INDIAN INSTITUTE OF TECHNOLOGY INDORE
September 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

I hereby certify that the work which is being presented in the thesis entitled **Learning with Randomized Neural Networks for Classification Problems** in the partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** and submitted in the **Department of Mathematics, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from **December 2018** to **September 2023** under the supervision of **Dr. M. Tanveer, Associate Professor, Indian Institute of Technology Indore**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Ashwani Kumar
September 14, 2023

Signature of the student with date

(Ashwani Kumar Malik)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

14/09/2023

Signature of thesis supervisor with date

(Dr. M. Tanveer)

Ashwani Kumar Malik has successfully given his Ph.D. Oral Examination held on **January 22, 2024**.

22/01/2024

Signature of thesis supervisor with date

(Dr. M. Tanveer)

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my heartfelt gratitude to a number of persons who in one or the other way contributed by making this time as learnable, enjoyable, and bearable. At First, I would like to thank my supervisor **Dr. M. Tanveer**, who was a constant source of inspiration during my work. Without his constant guidance and research directions, this research work could not be completed. His continuous support and encouragement has motivated me to remain streamlined in my research work.

I am also thankful to Prof. Swadesh Kumar Sahoo and Prof. Aruna Tiwari, my research progress committee members for their valuable feedback and comments which made significant improvements in my research work. I am also thankful to the Head, Department of Mathematics, for his continuous support and encouragement.

My sincere gratitude and respect to the worthy Director, Indian Institute of Technology Indore for providing the research facilities and support.

I would like to acknowledge the Council of Scientific and Industrial research (CSIR), India, for the fellowship during my Ph.D. programme.

I am thankful to Dr. M.A. Ganaie for his continuous support and assistance, whenever required. I am also thankful to my lab members, Anuradha, Sajid, Mushir, Abdul, and Rahul sir for the healthy discussion. Moreover, I would like to thank the Department staff, especially Jitendra sir for his support and assistance whenever required.

I am also thankful to administrative and other staff members of the institute for their support and assistance during the Ph.D. programme. I would like to express my heartfelt respect to my parents for their love, care and support they have provided to me throughout my life. Special thanks to my parents, sisters (Reenu and Meenu) and friends (Kulbhushan, Bikash, Pradeep) as this thesis would not have been possible without their support and encouragements. Finally, I am thankful to all who directly or indirectly contributed, helped and supported me.

Ashwani Kumar Malik

Dedicated

to

My Family and Friends

ABSTRACT

Randomized neural networks (RdNNs) have shown their strength in classification and regression problems. RdNNs with less computational cost and good generalization performance are highly desirable machine learning models. In RdNNs such as random vector functional link (RVFL) neural network, some parameters are kept fixed (during training), either in a stochastic or a deterministic way, and rest parameters are optimized via closed form or iterative methods. RVFL with direct links is a special randomized network. By using a closed form solution approach, RVFL avoids concerns that back-propagation trained networks experience, such as the local minima problem, sluggish convergence, and sensitivity to learning rate setting. This thesis aims to contribute to the evolution of RVFL by developing novel variants of RVFL for classification problems. We give an extensive review of the progress of RVFL, which is useful for beginners as well as professionals. RVFL assumes that all the samples are equally important, however, this may not be true in real world scenarios. To handle this issue, we employ fuzzy and intuitionistic fuzzy theory to reduce the negative influence of the noise/outliers over RVFL's performance and develop intuitionistic fuzzy RVFL (IFRVFL) and class probability-based fuzzy RVFL (CP-FRVFL) models. However, IFRVFL ignores the geometrical information of the data while calculating the final parameters. Therefore, to further improve it, we develop graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) that uses graph embedding framework to handle the geometrical relationship of the data. The least square method used in standard RVFL may struggle to separate possibly overlapping patterns, whereas nonparallel hyperplane-based classifiers such as twin support vector machine (TWSVM) and nonparallel support vector machine (NPSVM) perform well in such situations. While SVMs have their strengths, they are not as flexible as neural networks/deep learning models when it comes to automatically learning complex features from raw data. Therefore, we combine hyperplane-based classifiers with RdNNs and propose novel models, namely, nonparallel RVFL (NPRVFL), extended least squares twin SVM (ext-LSTSVM) and extended robust energy-based least squares twin SVM (ext-RELS-TSVM) to benefit from both hyperplane-based classifiers and neural networks, providing a more flexible solution for classification problems.

Single RVFL may struggle to capture complex and intricate patterns present in the data. Ensembles usually improve generalization performance by leveraging the advantages of many models. They have the ability to reduce overfitting and capture a wide range of patterns in the data. Ensembles often outperform individual models, achieving higher accuracy and reliability in predictions. Therefore, to improve the RVFL's stability and generalization performance, we develop extended feature RVFL (efRVFL), an ensemble of efRVFL (en-efRVFL), and rotated RVFL (RoF-RVFL) models. Shallow RVFLs have limited capacity to automatically learn deep hierarchical representations from raw data. Deep architectures, with multiple hidden layers, are better suited for learning such representations. Deep RdNNs with non iterative learning have fast training speed, less tunable parameters, and good generalization performance. Ensemble deep RVFL (edRVFL) with direct links and closed form solution method has good performance. However, the edRVFL model doesn't consider the geometrical relationship of the data while calculating the final output parameters. We propose an extended graph embedded RVFL (EGERVFL) model that employs both intrinsic and penalty subspace learning criteria under the graph embedded framework in its optimization process to calculate the output parameters. The proposed shallow EGERVFL model has only a single hidden layer and hence, has less representation learning. We further develop an ensemble deep EGERVFL (edEGERVFL) that can be considered a variant of edRVFL model. Finally, we present a brief summary of this thesis and potential future research directions.

Keywords: Random vector functional link network, intuitionistic fuzzy theory, classification, ensemble learning, ensemble deep learning.

List of Publications

Publications from the thesis

In Refereed Journals

1. **A. K. Malik**, R. Gao, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Random vector functional link network: recent developments, applications, and future directions, Applied Soft Computing (2023) 110377. doi:https: //doi.org/10.1016/j.asoc.2023.110377.

2. **A. K. Malik**, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network, IEEE Transactions on Computational Social Systems (2022). doi:10.1109/TCSS.2022.3146974.

3. **A. K. Malik**, M. Tanveer, Graph embedded ensemble deep randomized network for diagnosis of Alzheimer's disease, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2022). doi:10.1109/TCBB.2022.3202707.

4. **A. K. Malik**, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Extended features based random vector functional link network for classification problem, IEEE Transactions on Computational Social Systems (2022). doi:10.1109/TCSS.2022.3187461.

In Refereed Conferences

1. **A. K. Malik**, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Support vector machine based models with sparse auto-encoder based features for classification problem, in: International Conference on Neural Information Processing, Springer, 2022, pp. 248–259. doi:10.1007/978-3-031-30105-6_21.

2. **A. K. Malik**, M. A. Ganaie, M. Tanveer, Graph embedded intuitionistic fuzzy weighted random vector functional link network, in: 2022 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2022, pp. 293–299. doi:10.1109/SSCI51031.2022.10022212.

3. **A. K. Malik**, M. A. Ganaie, M. Tanveer, P. N. Suganthan, A novel ensemble method of RVFL for classification problem, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8. doi:10.1109/IJCNN52387.2021.9533836

Under submission

1. **A. K. Malik**, M. A. Ganaie, M. Tanveer, Class probability based fuzzy RVFL for classification problems.

2. **A. K. Malik**, M. A. Ganaie, M. Tanveer, Non-parallel RVFL for classification problems.

Publications other than thesis

In Refereed Journals

1. M. A. Ganaie, M. Hu, **A. K. Malik**, M. Tanveer, P. N. Suganthan, Ensemble deep learning: A review, Engineering Applications of Artificial Intelligence 115 (2022) 105151. doi:https://doi.org/10.1016/j.engappai.2022.105151.

2. M. Tanveer, A. Rastogi, V. Paliwal, M. A. Ganaie, **A. K. Malik**, J. Del Ser, C. T. Lin, Ensemble deep learning in speech signal tasks: a review, Neurocomputing (2023) 126436. doi:https://doi.org/10.1016/j.neucom.2023.126436.

3. M. A. Ganaie, A. Kumari, **A. K. Malik**, M. Tanveer, EEG signal classification using improved intuitionistic fuzzy twin support vector machines, Neural Computing and Applications (2022) 1–17. doi:10.1007/s00521-022-07655-x.

4. M. A. Ganaie, M. Sajid, **A. K. Malik**, M. Tanveer, Graph embedded intuitionistic fuzzy RVFL for class imbalance learning, IEEE Transactions on Neural Networks and Learning Systems (2024). doi:10.1109/TNNLS.2024.3353531.

5. M. Sajid, **A. K. Malik**, M. Tanveer, P. N. Suganthan, Neuro-Fuzzy Random Vector Functional Link Neural Network for Classification and Regression Problems, IEEE Transactions on Fuzzy Systems (2024). doi:10.1109/TFUZZ.2024.3359652.

6. M. Sajid, **A. K. Malik**, M. Tanveer, Intuitionistic fuzzy broad learning system: Enhancing robustness against noise and outliers, IEEE Transactions on Fuzzy Systems. doi:https://doi.org/10.48550/arXiv.2307.08713. [Under revision]

In Refereed Conferences

1. M. A. Ganaie, M. Tanveer, **A. K. Malik**, P. N. Suganthan, Minimum variance embedded random vector functional link network with privileged information, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–8. doi:10.1109/IJCNN55064.2022.9891930.

2. N. Ahmad, M. A. Ganaie, **A. K. Malik**, K. T. Lai, M. Tanveer, Minimum variance embedded intuitionistic fuzzy weighted random vector functional link network, in: International Conference on Neural Information Processing, Springer, 2022, pp. 600–611. doi:10.1007/978-3-031-30105-6_50.

Contents

List of Figures

List of Tables

List of Abbreviations

DT	Decision tree
RaF	Random forest
RoF	Rotation forest
SRM	Structural risk minimization
ELM	Extreme learning machine
RVFL	Random vector functional link network
RdNN	Randomized neural network
DNN	Deep neural network
SVM	Support vector machine
TWSVM	Twin support vector machine
LSTSVM	Least squares twin support vector machine
RELSTSVM	Robust energy based least squares twin support vector machine
KRR	Kernel ridge regression
IFKRR	Intuitionistic fuzzy kernel ridge regression
PCA	Principal component analysis
LDA	Linear discriminant analysis
LFDA	Local fisher discriminant analysis
NN	Neural network
dRVFL	Deep random vector functional link network
edRVFL	Ensemble deep random vector functional link network
BP	Back propagation
QPP	Quadratic programming problem
ML	Machine learning
k-NN	k-nearest neighbour
GE	Graph embedded
SL	Subspace learning

Chapter 1

Introduction

Artificial intelligence (AI) endeavors to create machines capable of emulating human-like intelligence. AI is a rapidly growing field that has the potential to revolutionize many aspects of the world [1]. The rapid growth of AI as a field, driven by the advancement of machine learning (ML) algorithms over recent decades, has ushered in transformative changes in human lives. In ML, several models, such as artificial neural networks (ANNs), support vector machines (SVMs), randomized neural networks (RdNNs), and so on, have been developed for classification problems. In classification problem, an input variable (x) is mapped to a discrete output (y) via a learning algorithm (f), i.e., $y = f(x, \Theta)$, where Θ is the parameter of the learning algorithm and $y \in \mathcal{Y}$, where $\mathcal{Y}$ contains the class labels. Depending on the cardinality of $\mathcal{Y}$, the classification problem may be binary or multiclass in nature. Therefore, developing advanced ML algorithms for classification problems holds substantial value.

In this thesis, our objective is to develop novel RdNNs for classification problems and their application in the diagnosis of Alzheimer's disease. This section gives a brief overview of the ML algorithms discussed in this thesis, followed by the research motivations, objectives, major contribution, and overall outline of the thesis.

1.1 Research background

The long history of evolution has endowed the human brain with numerous desirable characteristics, including huge parallelism, generalization capability, adaptivity, distributed representation and computing, and others. ANNs or neural networks (NNs), which have been designed following biological NNs, are expected to possess some of these appealing characteristics [2]. Researchers and scientists have employed ANNs to address a broad variety of problems arising in mathematics (such as function approximation), physics, medicine, economics, and other fields [3]. Typically, ANNs comprise an input layer, one or more hidden layers, and one output layer. These layers are interconnected by weighted connections, representing vital network parameters. Within these layers, neurons, often referred to as nodes, execute intricate mathematical operations encompassing both linear or nonlinear transformations. Generally, gradient descent (GD) methods are a widely used iterative process in ANN that aims to find the best settings for the model's parameters by comparing the predicted outputs with the expected ones. However, GD based approaches have some drawbacks, such as sluggish convergence, failure in obtaining global minima and being very sensitive to learning rate [4]. In contrast to traditional GD-trained NNs, RdNNs with closed-form solutions avoid the aforementioned problems [5]. These networks are faster to train and have good generalization performance [6, 7]. RdNNs have simple architecture, strong mathematical foundations [8, 9], and effective data modeling capacity [10]. Random vector functional link (RVFL) [11] network with direct links and closed form solution approach is a special single hidden layer feed forward neural network (SLFN) among existing RdNNs. Research demonstrates that direct links have a significant impact on RVFL's performance for classification [12] and regression problems [13, 14]. These direct connections separate RVFL from other RdNNs such as radial basis function (RBF) [15], extreme learning

machine (ELM) [16], and so on.

On the other hand, SVM [17] is an effective maximum margin classifier rooted in statistical learning theory [18], which offers a planes-based classification approach. SVM and its variants have been successfully employed in several domains [19, 20, 21]. However, SVM has a considerable degree of computational complexity, i.e., $O(N^3)$, where N represents the number of samples. As a result, twin SVM (TWSVM) [22], a framework designed to reduce the time complexity of SVM, obtains two nonparallel proximal hyperplanes to classify the data. TWSVM solves two smaller quadratic programming problems (QPPs), unlike SVM, which solves a single large QPP. TWSVM performs well with cross-plane datasets due to its nonparallel nature [22]. Over the last decades, numerous advanced TWSVM-based algorithms have been proposed, such as least square twin SVM (LSTSVM) [23], twin bound SVM (TBSVM) [24], energy-based LSTSVM (ELSTSVM) [25], Robust energy based LSTSVM (RELSTSVM) [26], intuitionistic fuzzy TWSVM (IFTWSVM) [27], and many others [28, 29]. According to the study [30], RELSTSVM is the most efficient classifier among twin SVM-based models. Inspired by non parallel hyperplanes based learning, several ANNs such as twin ELM (TWELM) [31] and twin RVFL [32] have been developed under this approach. TWSVM is successful in classifying data; however, it has several challenges, such as lack of sparsity, matrix invertibility on large datasets, sensitive to feature noise near decision boundaries [33], and it needs to reconstruct optimization problems for nonlinear cases by taking kernel-generated surfaces into account [34]. In order to improve TWSVM performance for classification tasks and to address the aforementioned issues, Tian et al. [35] developed a nonparallel SVM (NPSVM), which employs ϵ-insensitive and hinge loss functions, avoids matrix inversion operations (as in TWSVMs) and directly inherit kernel trick in itself. NPSVM constructs only two convex QPPs for both linear and nonlinear cases, and it doesn't need to compute the inverse of matrices prior to training (as

required in TWSVM) and efficiently uses the sequential minimization optimization (SMO) algorithm to solve its QPPs. Moreover, its sparsity is similar to standard SVMs, and when the parameters are properly chosen, NPSVM degenerates to the TWSVMs.

Single models often struggle to strike the right balance between capturing complex patterns in the training data while avoiding overly specific and noisy details. This can lead to poor generalization when applied to unseen data [36]. The ML community has paid significant attention in designing the ensemble learning approach to mitigate the aforementioned challenges of single models. Ensemble methods combine the predictions of multiple individual models (known as base models), each trained on a different subset or perspective of the data. By doing so, they harness the collective wisdom of these models, effectively reducing the risk of overfitting or underfitting [37]. Ensemble learning techniques have been successfully used in classification and regression problems [38]. Combining the diverse and unstable base models [36], the ensemble model performs better than its constituent models. Both decision trees (DTs) and NNs are unstable classifiers (low bias, high variance) and have been extensively studied in the ensemble learning framework [38, 39]. Extensive research has been done in developing ensemble algorithms such as random forest (RaF) [40], random subspace method [41], rotation forest (RoF) [42], and so on. Few research combines DTs and NNs to develop an efficient model: an ensemble of DT and RVFL [43] was proposed for multi-class classification problems; Katuwal and Suganthan [44] proposed an ensemble of RVFL classifiers, wherein dropout and dropConnect regularization techniques [45, 46] are employed; Ling et al. [47] proposed an improved ensemble of RVFLs based on particle swarm optimization (PSO) with a double optimization strategy (DOS); and in [48], an adaptive ensemble model combined boosting technique with RVFL network. Ensemble models have shown excellent performance in various fields such as malaria parasite classification [49], short-term electric load forecasting [50], Alzheimer's disease diagnosis [51],

and so on.

Typically, conventional ML models frequently encounter challenges when attempting to uncover intricate latent representations within data. This limitation can subsequently hinder the resultant ensemble model's ability to achieve high performance across a diverse range of datasets and tasks [52]. A branch of ML called deep learning (DL) aims to use hierarchical architectures to learn high-level representations from data. Recent research has shown the superiority of deep architectures over shallow architectures in terms of accuracy [53]. Deep neural networks (DNNs) have proven effective at identifying complex patterns in high-dimensional data and have been successfully applied to many fields in research, business, and government [54]. Conventional DNNs often pose challenges due to their extensive parameter count, typically in the millions, which necessitates meticulous tuning. Moreover, these networks may encounter sub-optimal solutions when employing the backpropagation (BP) optimization techniques [55]. As a remedy, Deep RdNNs have been put forth to avoid the pitfalls of the above-mentioned challenges of the DNNs [56]. Deep RdNNs offer several advantages, such as rapid training speeds and fewer tunable parameters, which make them capable of mitigating the complexity associated with conventional DNNs [56]. Recently, cutting-edge research has been conducted within the ensemble deep learning framework, strategically capitalizing on the synergistic potential of both deep and ensemble learning frameworks. Shi et al. [57] proposed deep RVFL (dRVFL) and ensemble deep RVFL (edRVFL) models, which seamlessly combine the inherent strengths of RdNNs, DNNs, and the effectiveness of ensemble learning methods. Research [58, 59] shows that deep and ensemble deep variants of RVFL model perform better than several deep models [60]. The ensemble deep RVFL models have shown their effectiveness and superior performance in several domains such as Alzheimer's disease diagnosis [61], time series classification problem [62], and so on.

The research outlined above deepens our fundamental comprehension of RdNNs and serves as a compelling catalyst, inspiring us to embark on a more profound exploration within the domain of RdNNs.

1.2 Motivation of the study

- In 2016, Zhang and Suganthan [12] conducted a comprehensive examination of RVFL architecture in the context of classification problems. Their investigation yielded several significant findings, including insights into the impact of output layer bias, the presence of direct connections from the input layer to the output layer, the choice of activation functions in the hidden layer, the scaling of parameter randomization, and the performance assessment of methods like the Moore–Penrose pseudoinverse and ridge regression (or regularized least square solutions) in the computation of output weights. Consequently, the RVFL model attracted the interest of researchers from various fields due to its fast training speed, direct links, simple architecture, and universal approximation capability, which make it a viable RNN. RVFL has undergone significant improvements over the years and has shown its strength in diverse domains, such as biomedical [63], forecasting [50, 64, 65], industries [66], non-linear system identification [67], driving fatigue detection [68], and many more. The absence of a systematic literature review on the progression of RVFL presents a challenge for both newcomers and practitioners seeking suitable models for diverse applications. Therefore, Chapter 2 presents a detailed review of the RVFL models to bridge this information gap and provides a consolidated and accessible resource to the research fraternity.

- Standard RVFL assigns uniform weights to each sample and hence considers all the

data samples to be equally significant while calculating the output parameters. However, in real-world scenarios, this uniform weighting assumption may not be ideal in the presence of outliers and noise in the datasets. Hence, RVFL shows lower generalization performance when confronted with outliers and noise within the datasets. Moreover, RVFL ignores the geometric and discriminative information of the data while training. Therefore, in Chapter 3, we propose novel RVFL models addressing the aforementioned issues.

- RVFL and ELM are two popular RdNNs. There are several research that combine plane-based techniques (i.e. SVM, TWSVM and so on) with ANNs architectures to benefit from both approaches. Ganaie and Tanveer [69] integrated LSTSVM with RVFL's features (original features and randomized features) and developed an improved model. TELM [31] learns two non-parallel separating hyperplanes in the randomized feature space. TELM performs better in terms of classification accuracy and has fewer optimization constraint variables than TWSVM. Borah and Gupta [32] developed unconstrained convex minimization-based implicit Lagrangian twin RVFL networks for binary classification (ULTRVFLC), and experimental results demonstrate that ULTRVFLC has superior performance than existing baseline models (SVM, ELM, TWSVM and RVFL). According to study [30], RELSTSVM is the most efficient classifier among TWSVM-based models. Moreover, NPSVM has superior theoretical properties and better generalization performance than TWSVM and SVM models. RVFL employs a least square classifier, which may encounter difficulty in distinguishing potentially overlapping patterns in the dataset. While hyperplane-based classifiers are suitable for classifying such data. On the other hand, SVMs do not have the inherent capacity to learn hierarchical or deep representations from raw data. Thus, inspired by the aforementioned studies, Chapter 4 introduces integrated

models that amalgamate hyperplane-based learning techniques with the RVFL network.

- The performance of a single RVFL model is often unstable due to the random nature of its hidden features [70], and it has less feature extraction capability that prevents it from capturing the complex hidden relationship within the data. Ensemble techniques are often regarded as the cutting-edge answer to such problems. By training multiple models and pooling their predictions, the ensemble method increases the predictive performance of a single model [37]. The ensemble of randomized models, such as random forests and RdNNs, effectively reduces the variance resulting from the random feature space. Ensemble RVFL models are more stable, robust and have better generalization performance compared to single RVFL, and have been successfully applied in various domains such as crude oil price forecasting [71], medical domain [72] and so on. Moreover, extensive research [42, 73] has been done in designing ensemble classifiers, wherein several base models are trained over different feature subsets or data subsets to get diverse base models. To improve the RVFL's stability and generalization performance, Chapter 5 presents novel ensemble RVFL models.

- Shallow RVFL models have limited capacity to automatically learn hierarchical and abstract features from raw data. The single hidden layer in a shallow architecture may struggle to capture complex relationships present in the data. Currently, DL architectures have shown their strength in classification and regression problems. Deep RdNNs such as edRVFL with non-iterative learning have fast training speed, fewer unknown parameters, and good generalization performance [57, 59]. However, edRVFL, in which each layer constitutes a base model, doesn't consider the geometrical relationship of the data while calculating the output parameters of the model.

Therefore, in Chapter 6, we propose an ensemble deep RVFL-based model to maintain the inherent topological properties of the data.

1.3 Research objectives

The following are the objectives of this thesis:

[1] To present a comprehensive review of the evolution of RVFL model.

[2] To develop robust shallow RVFL models using fuzzy theory and graph embedded approach.

[3] To propose novel hybrid models amalgamating hyperplane-based learning and randomized neural networks.

[4] To propose ensemble learning based RVFLs to improve its stability and performance.

[5] To propose an enhanced ensemble deep RVFL model incorporating graph embedding framework.

1.4 Major contributions

In this section, we provide a brief summary of our proposed works. The contributions to this thesis are listed below.

[1] The lack of a systematic literature review on RVFL's progression poses a challenge for researchers. Therefore, in Chapter 2, we contribute to the research community by providing a comprehensive review of the development of RVFL. In this review, we discuss the shallow RVFLs, ensemble RVFLs, deep RVFLs and ensemble deep RVFL models. The variations, improvements and applications of RVFL models are

discussed in detail. Moreover, we discuss the different hyperparameter optimization techniques that improve the generalization performance of RVFL. Finally, we give potential future research directions and opportunities that can inspire the researchers to improve the RVFL's architecture and learning algorithm further.

[2] During the training process, the standard RVFL assigns equal weights to each sample in order to determine their respective contributions. Nevertheless, it is important to acknowledge that in practical situations, this assumption may not be valid due to the existence of outliers and noise within the datasets. To address these challenges, in Chapter 3, we propose three novel RVFL-based models by employing the fuzzy weighting scheme. In the first approach, we propose an intuitionistic fuzzy RVFL (IFRVFL) model which assigns each sample an intuitionistic fuzzy (IF) number based on its membership and non-membership values. The membership function considers the sample's distance from the centroid of its corresponding class, and the non-membership function considers the sample's distance from the centroid as well as takes the neighbourhood information of each sample. Hence, by allocating distinct weights to each sample, the proposed IFRVFL model exhibits enhanced intelligence compared to the standard RVFL. Consequently, it acquires the ability to effectively manage noise and outliers present within the datasets. As an application, we effectively employ the proposed IFRVFL model for the diagnosis of AD. In the second approach, we propose a class probability based fuzzy RVFL (CP-FRVFL) model. The proposed CP-FRVFL uses a non-linear membership function to determine the membership value of each sample. Moreover, the k-nearest neighbour (k-NN) technique is used to determine the class probability of each sample to take its surrounding information. Finally, a score value is assigned to each sample based on a score function by combining the membership function and class probability. Although the proposed

IFRVFL model handles the issues related to noise and outliers, but the randomization process distorts the original structure of the data. As per the literature [74], the graph embedding approach can be used to preserve the geometrical relationships in the datasets. Therefore, in the third approach, we propose a graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) model. The proposed GE-IFWRVFL model has two major benefits: (i) it leverages graph embedding to maintain the inherent geometrical linkages within the datasets, and (ii) it uses IF theory to handle the noise and outliers within the datasets.

[3] In Chapter 4, we propose integrated approaches that combine RVFL networks with hyperplane-based models, leveraging the strengths of both methodologies for enhanced model performance. In the first approach, we propose a novel non-parallel RVFL (NPRVFL), that combines NPSVM model with RVFL network. The proposed NPRVFL encompasses two non-parallel separating hyperplanes and is designed to acquire knowledge from the RVFL's feature space (original features combined with randomized features) for data classification. In the proposed NPRVFL model, each hyperplane is subject to joint optimization, minimizing its proximity to one class while simultaneously ensuring a substantial separation from the other class. On the other hand, the sparse pre-trained RVFL (SP-RVFL), an advanced variant of RVFL, employs a sparse autoencoder with $l1$-norm regularization to acquire enhanced hidden layer parameters. In the second approach, we propose novel models, namely extended LSTSVM (ext-LSTSVM) and extended RELS-TSVM (ext-RELS-TSVM), wherein we use SP-RVFL's feature space (original features and auto-encoder based hidden features) to learn the hyperplanes for data classification.

[4] RVFL is an unstable classifier due to the randomization process and has less feature

representation due to the presence of only one hidden layer. To address these issues, we study RVFL with ensemble learning approaches in Chapter 5. Firstly, we propose an extended feature RVFL (efRVFL) trained over extended feature space generated analytically from the original feature space. The proposed efRVFL has three types of features, i.e., original features, supervised randomized (newly generated) features and unsupervised randomized features, in its feature matrix. The proposed efRVFL with additional features has the capability to capture non-linear hidden relationships within the dataset. Ensemble models are more stable and have better generalization performance than single models. Therefore, we propose an ensemble of extended feature RVFL (en-efRVFL). Each base model (efRVFL) of en-efRVFL is trained over different feature spaces so that more accurate and diverse base models can be generated. The outcomes of base models are integrated via the average voting scheme. Secondly, we propose a novel ensemble method, namely, a rotated random vector functional link neural network (RoF-RVFL), which combines rotation forest (RoF) and RVFL classifier.

[5] Randomized shallow and deep NNs with closed form solutions avoid the shortcomings of the BP-based trained NNs. The edRVFL network utilizes the strength of two growing fields, i.e., deep learning and ensemble learning. However, the edRVFL model harms the geometrical relationship of the data in the training process. Graph embedded frameworks have been successfully employed to maintain the geometrical relationship within data. In Chapter 6, we propose an extended graph embedded RVFL (EGERVFL) that, unlike standard RVFL, employs both intrinsic and penalty subspace learning (SL) criteria under the graph embedded framework in its optimization process to calculate the model's output parameters. The proposed shallow EGERVFL has only a single hidden layer and, hence, has less representation learn-

ing. Therefore, we further develop an ensemble deep EGERVFL (edEGERVFL) that can be considered as a variant of the edRVFL model. Unlike edRVFL, the proposed edEGERVFL model solves graph embedded based optimization problems in each layer and, hence, has better generalization performance than the edRVFL model.

1.5 Outline of the thesis

There are seven chapters in this thesis. Figure 1.1 shows the overall visual organization of the thesis. The following is a brief overview of the chapters:

[1] In chapter 2, we present a comprehensive review [75] of the RVFL network's development, followed by a brief discussion of the existing literature related to this thesis work.

[2] In chapter 3, we propose three models, i.e., intuitionistic fuzzy RVFL (IFRVFL) [76], class probability based fuzzy RVFL (CP-FRVFL) and graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) [77] for classification problems.

[3] In chapter 4, we combine hyperplane-based learning techniques with RdNNs approaches and propose non-parallel RVFL (NPRVFL), extended LSTSVM (ext-LSTSVM) [78], and extended RELS-TSVM (ext-RELS-TSVM) models [78].

[4] In chapter 5, we study RVFL with ensemble learning approaches and propose the following models: extended feature RVFL (efRVFL) [79], ensemble of efRVFL (en-efRVFL) [79], and rotated RVFL (RoF-RVFL) [80].

[5] In chapter 6, we develop an extended graph embedded RVFL (EGERVFL) [81] and its ensemble deep variant, i.e. ensemble deep EGERVFL (edEGERVFL) [81].

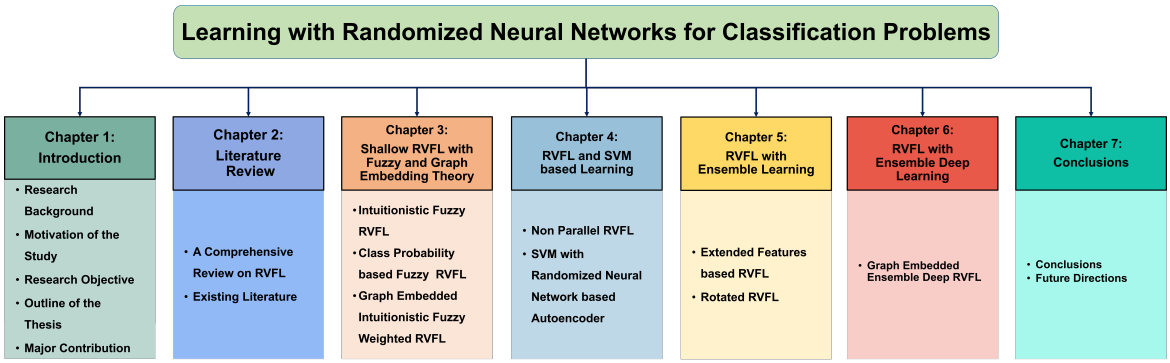


Figure 1.1: The overall outline of the thesis.

[6] In chapter 7, we present a summary of the thesis along with potential future research recommendations.

Chapter 2

Literature Review

In this chapter, we present the literature related to the research work of this thesis. We give a brief introduction of the artificial neural network in section 2.1 and randomized neural in section 2.2. We present a comprehensive review [75] of the RVFL network in section 2.3 and ELM is discussed in section 2.4. Plane-based learning methods such as support vector machines, twin support vector machines and its variants are presented in section 2.5. Section 2.6 discusses some miscellaneous theories based on intelligent machine learning models that have been developed in the literature. Decision trees and their ensemble models are presented in section 2.7. Section 2.8 gives a brief overview of Alzheimer's diseases and statistical tests that are used to evaluate the performance of machine learning models are presented in section 2.9.

2.1 Artificial neural networks

Artificial intelligence (AI) is a rapidly growing field that has the potential to transform many aspects of the world [1]. It refers to the development of computer systems that can perform tasks that typically require human intelligence, such as learning, decision making, and problem-solving [82]. AI has the potential to revolutionize industries and improve

efficiencies in a wide range of fields, including healthcare [83], transportation [84], finance [85], and energy [86]. Machine learning algorithms are the engine of AI. Hence, developing advanced machine learning algorithms for various tasks is of real value.

Among machine learning algorithms, artificial neural networks (ANNs) have received considerable attention due to their success in diverse domains such as medicine [87], chemistry [88], robotics [89], control systems [90], industrial applications and function approximation [91, 92], and so on. The architecture of ANN is inspired by the topology of biological neurons [93]. The ANN consists of neurons which are simple processing units and these neurons are connected via weighted links. The neurons do mathematical operations that are either linear or nonlinear, and they carry out some task that enables the ANN to approximate the unknown function (rule) that generates the data [94]. In general, the training phase of an ANN is an iterative process, and all the parameters are tuned via the backpropagation (BP) method [95]. However, the traditional iterative techniques based on the BP algorithm have some challenges, i.e., slow convergence [96], not getting global minima that leads to sub-optimal parameters [97], and very sensitive to learning rate [98].

2.2 Randomized neural networks

The behavior of neural systems with the majority of fixed connections is investigated through randomized neural networks (RdNNs), either in a stochastic or deterministic manner. Such systems often take the form of shallow/multi-layered neural network architectures with initialized connections to one or more hidden layers that are not trained. The class of RdNNs naturally possesses a number of intriguing aspects due to the training algorithms' restriction to working with a small number of weights [56]. The advantages of randomization have been extensively investigated for shallow networks [4, 99]. In addition, they are very appealing from a theoretical perspective [100, 101] because of their randomization, and

universal approximation properties [9, 94]. Neural networks trained using randomization-based algorithms exhibit their attractiveness in terms of accuracy and processing speed [102]. Numerous studies demonstrate that on short datasets, RdNNs trained via closed-form solutions outperform neural networks trained using gradient-based methods [62]. More recently, RdNNs have also emerged as multi-layer and ensemble architectural ideas [103]. As a result, a number of fresh deep and/or ensemble-based randomization-based methods have been developed that enhance the modeling effectiveness of earlier single-layer RdNNs architectures while retaining the computational effectiveness of randomization-based training [104]. Recently, a number of experimental investigations examined the potential of RdNNs in a number of application fields, including wave height forecasting [105], intelligent fault diagnosis [106], biomedical domain [61], renewable energy prediction [107] and so on.

2.3 Random vector functional link network: a comprehensive survey

To overcome the aforementioned issues with ANNs (see Section 2.1), randomization-based techniques have been proposed [4, 103] with fast convergence and universal approximation properties. Randomization-based neural network such as the RVFL model has several characteristics such as fast training speed, direct links, simple architecture, and universal approximation capability, that make it a viable randomized neural network. This section presents a comprehensive review of the evolution of the RVFL model, which can serve as an extensive summary for beginners as well as practitioners. We discuss the shallow RVFLs, ensemble RVFLs, deep RVFLs, and ensemble deep RVFL models. The variations, improvements, and applications of RVFL models are discussed in detail. Moreover, we discuss the different hyperparameter optimization techniques followed in the literature to improve the

generalization performance of the RVFL model. Finally, we give potential future research directions/opportunities that can inspire the researchers to improve the RVFL's architecture and learning algorithm further.

The single hidden layer feed-forward neural (SLFN) network architectures have been extensively studied in the last twenty years and employed in various domains, e.g., classification and regression problems, due to their universal approximation capability [8, 9, 94, 108, 109]. In most of the SLFN networks, the learning process is done in the output layer while weights and biases are generated randomly in the hidden layer. The output layer weights are calculated either via closed-form solution [110] or iterative process [111]. The origin of randomized feedforward networks can be traced to the late 20th century [103]. In 1988, Broomhead and Lowe [15] discussed universal approximation property using radial basis function (RBF) network with random centers [6]. There are several other architectures [112] like RBF network, and recurrent neural network (RNN) which have randomization-based training algorithms [4].

Moreover, Schmidt et al. [113] proposed a feed-forward neural network with random weights based on a randomization technique. At the same time, connecting the input layer to the output layer via direct link, Pao et al. [11, 114] proposed a random vector functional link (RVFL) neural network in 1992, wherein the parameters (weights and biases) from the input layer to the hidden layer are generated randomly from a fixed domain and the output weights are needed to be computed analytically. The idea of the direct link can be traced back to the pioneering work in fuzzy systems [115] in 1985. The direct links have shown significant improvement in RBF networks' performance [116] in 2002. The intention behind creating a direct link between inputs and outputs is to capture information about how the first derivative of the output with respect to the inputs. In 1994, Igelnik and Pao proved that RVFL is a universal approximator [117]. Recently, Needell et al. [118] proposed RVFL

networks for function approximation on manifolds that fill the theoretical gap lacking in [117]. In 2016, Zhang and Suganthan [12] conducted a comprehensive evaluation of RVFL for classification problems and concluded some remarkable results about this architecture. After that, the RVFL model got the attention of researchers from diverse domains due to its simple architecture, fast training speed, and universal approximation capability. The shallow RVFL model has been employed successfully in several domains, i.e., forecasting [50, 65, 119], non-linear system identification [67], function approximation [117], classification and regression problem [13, 120, 121], etc.

The RVFL model transforms the original feature space into randomized feature space via a random feature map (RFM), and this randomization process makes the RVFL model an unstable classifier. Ensemble learning techniques develop stable, robust, and accurate models integrating the several models known as base models [38]. Combining the diverse and accurate base models [36], the ensemble model performs better than it constitutes models. Broadly speaking, ensemble learning can be divided into three categories, i.e., bagging [122], boosting [123] and stacking framework [124]. Thus, the RVFL model has been improved (developed) in ensemble frameworks, and the more stable and robust RVFL variants have been proposed and employed in various domains, i.e., crude oil price forecasting [125], medical domain [126], and classification problem [127], etc. Deep learning architectures have a high representation learning capability due to several stacked layers for extracting informative features [56] and have been successfully employed in several domains, i.e., computer vision [128], bioinformatics [129], and visual tracking [130], and speech recognition task [131] etc. On the other hand, utilizing the strength of two individual growing fields, i.e., ensemble learning and deep learning, researchers are developing ensemble deep models [52, 132]. The shallow RVFL model has been extended to deep and ensemble deep architectures that improve its generalization performance. The deep RVFL network [57] has

several stacked layers wherein all parameters of hidden layers are generated randomly and kept fixed during the training process, and only output layer parameters are needed to be computed analytically. The deep RVFL model has better representation learning compared to the shallow RVFL model. The deep RVFL model faces memory issues when training data size, the no. of hidden layers, and the feature dimension of the data are considerable. Therefore, to address these issues, an implicit ensemble technique-based ensemble deep RVFL network known as the edRVFL model has been proposed [57].

RVFL has been improved in multiple aspects both in shallow and deep frameworks and has been applied in diverse domains. In this chapter, we present the journey of shallow and deep RVFL along with its applications. We conclude this article with potential future research directions that might inspire researchers to develop this architecture further.

The rest part is organized as follows. In Section 2.3.1, we present the formulation of the standard RVFL model. Section 2.3.2 discusses the research methodology and objectives. The improvements in shallow RVFL and their applications are discussed in Section 2.3.3. In Section 2.3.4 and 2.3.5, we discuss the semi-supervised methods and clustering-based methods that have been developed based on the RVFL model, respectively. We present the ensemble learning-based RVFL model in Section 2.3.6 and Section 2.3.7 discusses the deep architectures based on the RVFL model. Section 2.3.8 discusses the hyper-parameters optimization and experimental setup details and Section 2.3.9 discusses the applications of the RVFL model. In Section 2.3.10, a comparison of RVFL with other machine learning models is given. Finally, the potential future directions with conclusions are given in Section 2.3.11. Figure 2.3 shows the layout of the review work.

2.3.1 Mathematical formulation of RVFL network

In this section, we discuss the formulation of the standard RVFL model. Let $X = [x_1, x_2, \cdots, x_N]^T, \ \ x_i \in \mathbb{R}^d$ be the training dataset and $Y = [y_1, y_2, \cdots, y_N]^T, \ \ y_i \in \mathbb{R}^c$, be the target matrix. Here, d represents the number of features in each sample (x_i), and c denotes the number of classes. RVFL is a randomized version of a single hidden layer feedforward neural (SLFN) network, with three layers known as the input, hidden, and output layers. All three layers consist of neurons that are connected via weights. To avoid the implementation of the backpropagation algorithm, the weights from the input layer to the hidden layer are generated randomly from a domain and kept fixed during the training process. Only the output weights are analytically computed by the least square method. In RVFL, original features are also used to link the input and output layers. The direct links improve the generalization performance of RVFL [12]. The architecture of RVFL is given in Fig.2.1 (a) and Figure 2.1 (b) shows the different types of architectures of the RVFL model. Mathematically, the RVFL model, i.e., $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^c$, can be written as:

$$\mathbf{f}(x_i) = \sum_{k=1}^{d} \beta_k x_{ik} + \sum_{k=d+1}^{L} \beta_k \theta(\langle \mu_k, x_i \rangle + \sigma_k), \ i = 1, 2, \cdots, N. \tag{2.1}$$

In particular, $\langle \mu_k, x_i \rangle = \mu_k \cdot x_i$ is the standard inner product defined on Euclidean space $(\mathbb{R}^d)$. The objective function of the standard RVFL model with L hidden nodes can be written as:

$$\begin{aligned} &\min \ \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\|\xi\|^2 \\ &\text{subject to } H\beta - Y = \xi. \end{aligned} \tag{2.2}$$

where $\|.\|$ represents the Frobenius norm and $\xi = [\xi_1, \xi_2, \ldots, \xi_N]^T$ is the error term

corresponding to N samples. This is the quadratic optimization problem with linear constraints. β and H are the output weight matrix and the concatenation matrix consists of input data and outputs from the hidden layer, respectively and Y is the target matrix.

The optimization problem (2.2) can be rewritten as:

$$\min_{\beta \in \mathbb{R}^{(d+L)\times c}} \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \|H\beta - Y\|^2, \tag{2.3}$$

here,

$$H = [H_1 \; H_2]_{N\times(d+L)},$$

where

$$H_1 = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}_{N\times d}, \tag{2.4}$$

and

$$H_2 = \begin{bmatrix} \theta(\mu_1 \cdot x_1 + \sigma_1) & \theta(\mu_2 \cdot x_1 + \sigma_2) & \dots & \theta(\mu_L \cdot x_1 + \sigma_L) \\ \vdots & \ddots & \vdots & \vdots \\ \theta(\mu_1 \cdot x_N + \sigma_1) & \theta(\mu_2 \cdot x_N + \sigma_2) & \dots & \theta(\mu_L \cdot x_N + \sigma_L) \end{bmatrix}_{N\times L}, \tag{2.5}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{(d+L)} \end{bmatrix}_{(d+L)\times c} \quad \text{and} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N\times c}.$$

Here, $\beta_k = [\beta_{k1}, \beta_{k2}, \cdots, \beta_{kc}]$ is the output weight vector connecting the k^{th} input (hidden) node and the output nodes, where $1 \leq k \leq d + L$, and $\mu_j = [\mu_{j1}, \mu_{j2}, \cdots, \mu_{jd}]$ is

the weight vector connecting the j^{th} hidden node and the input nodes, $1 \leq j \leq L$. Also, $x_i = [x_{i1}, x_{i2}, \cdots, x_{id}]$ is the i^{th} sample. For the target matrix, $y_i = [y_{i1}, y_{i2}, \cdots, y_{ic}]$, $1 \leq i \leq N$. Moreover, $\theta(\cdot)$ and σ_i are the non-constant activation function and the bias term of i^{th} hidden node, respectively.

The optimal solution of the problem (2.2) when $\delta = \frac{1}{C} = 0$ is as follows:

$$\beta = H^{+}Y, \tag{2.6}$$

where H^{+} represents the Moore-penrose generalized inverse of the matrix H [133]. The regularization term is employed to avoid the over-fitting issue.

Therefore, the optimization problem with the regularization term is solved. Let the Lagrangian be

$$L(\beta, \xi, \alpha) = C\frac{1}{2}\|\xi\|^2 + \frac{1}{2}\|\beta\|^2 - \alpha^T(H\beta - Y - \xi), \tag{2.7}$$

and obtain the partial derivatives of L w.r.t β, ξ and α and set them to zero.

$$\frac{\partial L}{\partial \beta} = 0 \implies \beta = H^T\alpha, \tag{2.8}$$

$$\frac{\partial L}{\partial \xi} = 0 \implies \alpha = -C\xi, \tag{2.9}$$

$$\frac{\partial L}{\partial \alpha} = 0 \implies H\beta - Y - \xi = 0. \tag{2.10}$$

When the number of features is less than number of samples, from (3.19) and (3.20), we

obtain $\alpha = -C(H\beta - Y)$. By substituting the value of α in (3.18):

$$\beta = (H^T H + \frac{1}{C} I)^{-1} H^T Y \tag{2.11}$$

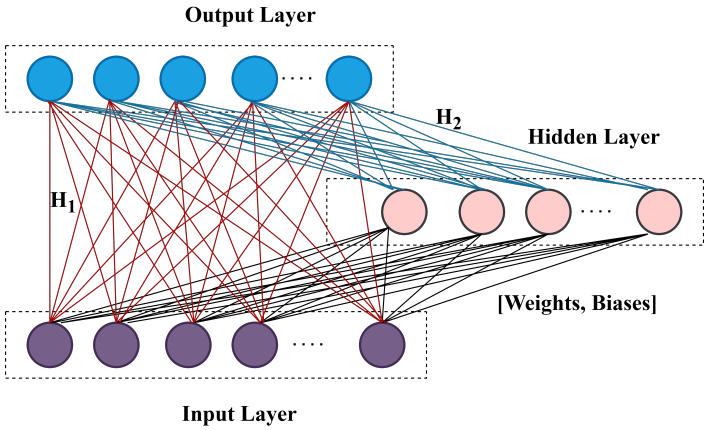
When the number of samples is less than the number of features, after substituting (3.18) and (3.19) into (3.20), we obtain $\alpha = (HH^T + \frac{1}{C} I)^{-1} Y$. By substituting the value of α in (3.18):

$$\beta = H^T (HH^T + \frac{1}{C} I)^{-1} Y. \tag{2.12}$$

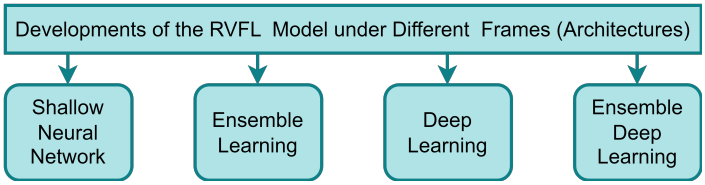
Therefore, in this case the optimal solution of (2.2) is given by,

$$\beta = \begin{cases} (H^T H + \frac{1}{C} I)^{-1} H^T Y, & (d + L) \leq N, \\ H^T (HH^T + \frac{1}{C} I)^{-1} Y, & N < (d + L), \end{cases} \tag{2.13}$$

where C is the regularization parameter to be tuned and I is an identity matrix of appropriate dimension. Both matrices $H^T H$ and HH^T are symmetric positive semidefinite matrix and $C > 0$, so both matrices in (3.23) is positive definite, therefore, $(HH^T + \frac{1}{C} I)$ and $(H^T H + \frac{1}{C} I)$ are non singular matrix.



(a) Shallow RVFL: Red lines show the direct links between the input layer to output layer, black lines show the connection between the input layer to hidden layer and blue lines show the links between the hidden layer to output layer (Best viewed in color).



(b) Different types of architectures of RVFL model.

Figure 2.1: The architectures of RVFL model.

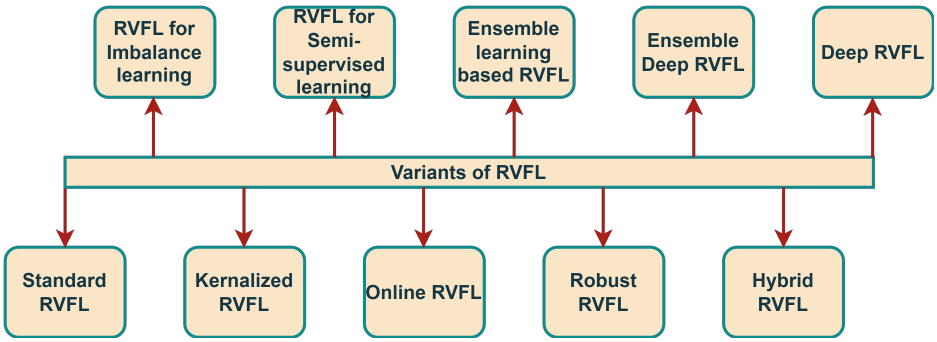


Figure 2.2: Algorithmic variants of the RVFL model.

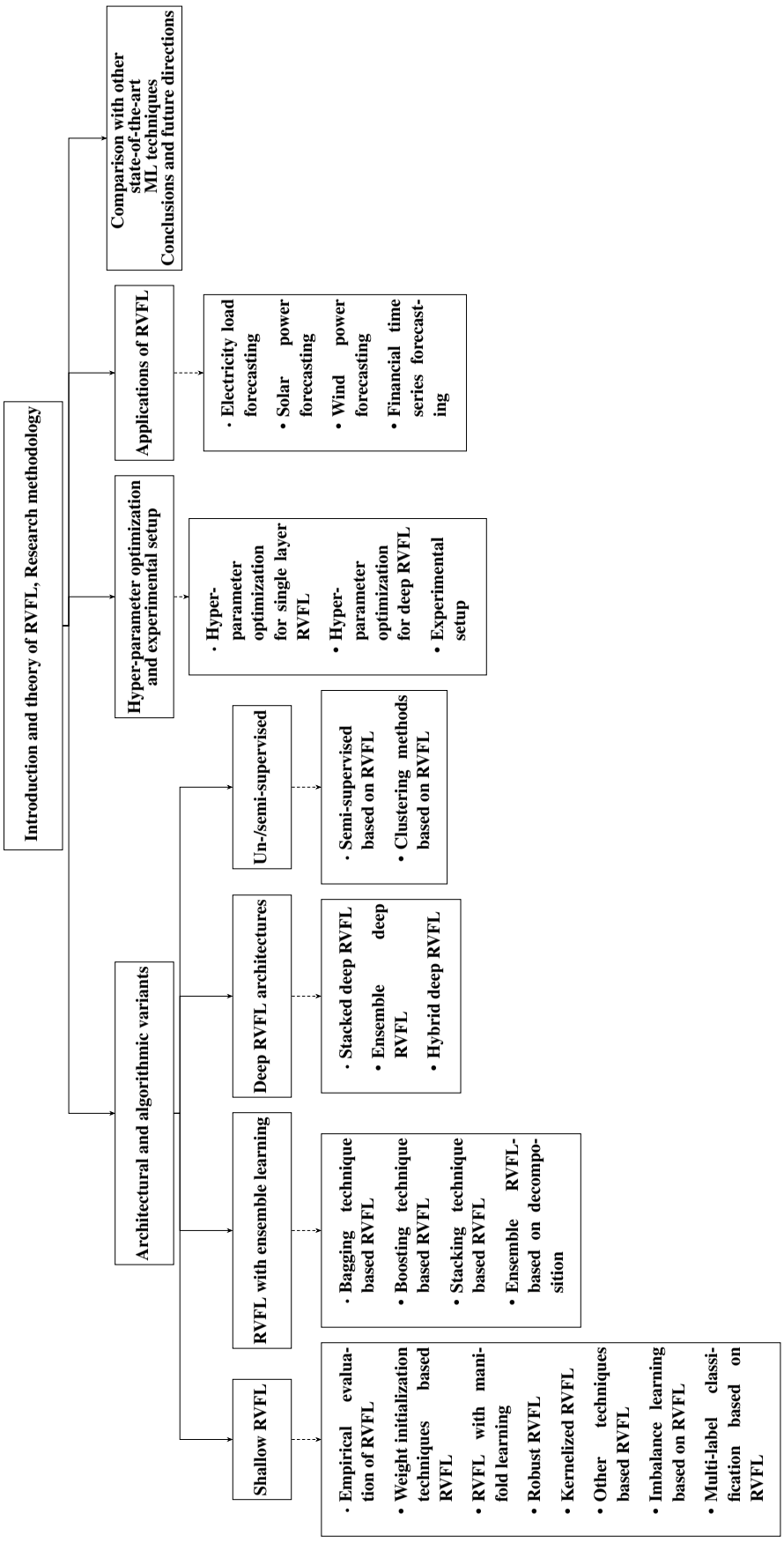


Figure 2.3: Layout of the RVFL's review.

2.3.2 Research methodology

The studies included in this review were obtained by searching the Google Scholar and Scopus search engines. The papers are the result of keywords random vector functional link and deep RVFL. The articles are screened based on the title and abstract, followed by the screening of the full-text version. The focus of this article is to represent the developments of shallow RVFL, ensembles of RVFL models, deep RVFL, ensembles of deep RVFL-based models, and their applications. Having these search strategies, we included fundamental papers on the RVFL model proposed in the late 20th century and after that, we included all RVFL papers from 2016 onward. This review discusses the following issues:

[1] How are the weights initialized in RVFL-based models?

[2] Techniques used to improve the robustness of the models in the presence of noise/outliers and class imbalance problems.

[3] Techniques followed for selection of hidden nodes like kernelized approach for RVFL.

[4] How do different approaches of ensemble learning like bagging, boosting, and stacking improve the performance of the RVFL-based ensemble models?

[5] Techniques followed for the development of deep RVFL-based models.

[6] Development of RVFL-based models for different scenarios like semi-supervised learning, unsupervised learning and regression problems.

[7] How to select the different parameters of the RVFL-based models, followed by the techniques for the optimization of these models?

[8] Approaches followed for analyzing the effect of hyper-parameters on the generalization performance of the RVFL-based models.

[9] The loopholes in the current literature of the RVFL-based models and possible future research directions.

2.3.3 Developments and applications of RVFL model

In this section, we discuss the journey of improvements in the shallow RVFL architecture.

2.3.3.1 Empirical evaluation of RVFL for classification and regression problems

Few scientific questions regarding the RVFL model have been addressed rigorously in the literature. Zhang and Suganthan [12] conducted a comprehensive evaluation of the RVFL model to answer the following questions, i.e., the impact of direct links, the effect of bias term in the output layer, and different types of activation functions in the hidden layer, domain from where random parameters are generated and the output weights calculation techniques over 121 UCI classification datasets. The experimental results demonstrate that RVFL with direct links from the input layer to the output layer has better generalization performance than RVFL without direct links, and output weights calculated via the ridge regression method are better than Moore-Penrose pseudoinverse-based output weights. Moreover, generating the random weights and biases within a suitable domain from the input layer to the hidden layer has a significant impact on the performance of RVFL, and biases term in the output layer may or may not have any impact on the performance of RVFL. Similarly, a comprehensive evaluation of orthogonal polynomial expanded RVFL (OPE-RVFL) [13] method was conducted in which the input patterns are expanded non-linearly using four orthogonal polynomials, i.e., Chebyshev, Hermite, Laguerre, and Legendre with three activation functions tansig, logsig, and tribas. The results demonstrate that direct links in the OPE-RVFL model also have a significant impact on regression problems, and the ridge regression-based approach is better than the Moore-Penrose pseudoinverse-based approach. Basically, this study also supports direct links in RVFL (as in [12]). Moreover, the results demonstrate that the OPE-RVFL model with the Chebyshev polynomial performs better than other compared polynomials, and tansig is the best-performing activation function followed by tribas and logsig. Table 2.1 shows the commonly used activation functions with the RVFL model in the literature.

The random feature mapping (RFM) mechanism has a significant impact on the per-

formance of RVFL and plays a vital role in the success of RVFL, but there is very little research to explore this topic. Cao et al. [134] conducted an experiment to study the relationship between the rank of input data and the performance of RVFL and introduced a concept to measure the quality of RFM via dispersion degree of matrix information distribution (DDMID). Moreover, in [134], several scientific questions were addressed, such as the relationship between the performance of the RVFL model and the rank of the input dataset, the sensitivity of this relationship with the different types of activation function, and the number of hidden nodes, respectively and many more. Rasheed et al. [135] implemented standard RVFL with different activation functions in the respiratory motion prediction and compared RVFL with direct and without direct links. The authors find out that the results with hardlim activation function are better than results with sigmoid, sine, tribas, radbas and sign functions, and direct links prevent RVFL from overfitting. Also, RVFL with direct links has better generalization performance than RVFL without direct links. Dudek [14] also conducted an experiment to answer a few scientific questions related to the RVFL model, such as whether direct links and bias terms in the output layer are necessary for modeling the data in regression problem and what should be the optimal domain (or optimal process) for choosing the random parameters for the hidden layer. In this study, three methods for selecting the random parameters are considered, and the author concludes that direct links seem helpful in modeling the target function with linear regions. The target function with a non-linear nature can be modeled with RVFL having no direct connections and bias terms.

2.3.3.2 RVFL with different weight initialization techniques

The RVFL model has a random initialization process from the input layer to the hidden layer, wherein the original feature space is transformed into randomized feature space, and hence, RVFL has a breakneck training speed and less tunable parameters. It is also capable of modeling complex (linear/non-linear) data and has universal approximation property [117]. On the other hand, the randomization mechanism makes RVFL an unstable model. The quality of random weights and hidden biases significantly affect the RVFL model's performance. Traditionally, the random weights and hidden biases are randomly generated with some probability distribution from a fixed domain, i.e. [-1, 1] and [0, 1], respectively. Study

[12] reveals that the aforementioned range is not the optimal domain for choosing the random parameters. Zhang et al. [120] proposed a sparse pre-trained random vector functional link (SP-RVFL) network in which a sparse autoencoder with L_1 norm is employed to generate weights and hidden biases from the input layer to the hidden layer. Sparse auto-encoder learned superior network parameters than the randomization process. The experimental evaluation on different datasets demonstrates that SP-RVFL outperforms standard RVFL. SP-RVFL model generates sparse weights between the input layer to hidden layer that is better as compared to randomly generated weights.

Cao et al. [136] studied the relationship between the probability distribution of the features (variables) in the datasets and the probability distribution of input weights and hidden biases in a neural network with random weights (NNRW) (in particular with RVFL and ELM). This study generated seven regression datasets with known distributions (Gaussian, Gamma, and Uniform). The input weights and hidden biases are initialized with different distributions, and the models, i.e., RVFL and ELM, are trained, respectively. The experimental results conclude several phenomena, such as the model having input weights and hidden biases from the Gaussian distribution can have a faster convergence rate than ones with the Gamma and Uniform distributions. Suppose one or more features follow the Gamma distribution, and input weights and hidden biases follow the same distribution. In that case, the corresponding model has a slow convergence rate and faces an over-fitting issue, etc. In another similar study [121], experiments were conducted to study the impact of Gaussian, Uniform, and Gamma distributions on the performance of the RVFL model, and the results suggest that input weights and hidden biases' probability distribution have a significant impact on the performance of RVFL model. This study also confirms that RVFL with direct links performs better than RVFL without direct links.

Tanveer et al. [137] proposed several ways to generate input weights and hidden biases in RVFL, wherein twin bounded support vector machine (SVM), least-square twin SVM, twin k-class SVM, least-square twin k-class SVM, and robust energy-based least square twin SVM models were employed to initialize input weights and hidden biases. The experiments illustrate that the twin bounded SVM-based approach has better generalization performance with a lower model rank (Friedman rank) than other proposed approaches.

In [138], the random subspace Fisher linear discriminant (FLD) method generates the random weights and hidden biases wherein important features are assigned higher weights than weights assigned to more minor essential features. Therefore, this approach improves the performance of RVFL. Pan et al. [139] proposed a novel method, Jaya-RVFL, where a new emerging intuitive optimization technique- Jaya algorithm [140] is employed to optimize the randomization range of input weights and employed to transient stability assessment in power system. Lu et al. [141] proposed three classification methods, i.e., MobileNet-RVFL-CBA, MobileNet-ELM-CBA, and MobileNet-SNN-CBA, respectively, to classify the brain MRI image. In each scenario, first, the MobileNetV2 [142] trained over ImageNet data is employed to extract the features from brain MRI images and then to improve the generalization performance of these randomized neural networks (RVFL, ELM, and SNN [113]), chaotic bat algorithm (CBA) is utilized to optimize the random weights and biases. The empirical evaluation of these models for classifying brain MRI data demonstrates that MobileNet-RVFL-CBA performs better than MobileNet-ELM-CBA and MobileNet-SNN-CBA in terms of sensitivity and overall accuracy. In the literature, it can be seen that several approaches such as different distributions [136], autoencoder [120], and SVM models [137] have been used to calculate the weights from input layer to hidden layer. There is no such method which is always adaptable. Therefore, further research is required to develop efficient techniques to initialize the weights from the input layer to the hidden layer in RVFL.

2.3.3.3 RVFL with manifold learning theory

Maintaining the data's global and local geometric structure while data is processed via several randomization processes is a challenging task. Manifold learning theory helps to overcome this issue. The standard RVFL model does not consider the geometrical aspect of the data and hence, loses information that leads to lower generalization performance. Li et al. [143] proposed a discriminative manifold RVFL neural network (DM-RVFLNN), wherein a soft label matrix is used to enlarge the margin between inter-class samples that makes the DM-RVFLNN model more discriminative, and manifold learning based with-in class similarity graph is used to enhance the compactness and the similarity with-in class

samples. Experiments on the rolling bearing fault diagnosis process demonstrate that the DM-RVFLNN model is superior and effective compared to standard RVFL. Considering the topological relationship of samples and to improve the robustness of the RVFL model, a sparse Laplacian regularized RVFL neural network with $L_{2,1}$-norm (SLapRVFL) [144] was proposed to assess the dry weight of hemodialysis patients and the experiments demonstrate that SLapRVFL is more robust than standard RVFL. Here, $L_{2,1}$-norm was used with regularization term to obtain sparse output parameters. The standard RVFL model doesn't consider the within-class/total variance of training data while obtaining the final output parameters; therefore, Ganaie et al. [145] proposed two variants of the RVFL model known as total variance minimization-based RVFL (Total-Var-RVFL) that employs the complete variance information of the training data in the objective function of standard RVFL, and intraclass variance minimization based RVFL (Class-Var-RVFL) wherein the variance of each sample from its respective class is considered, and hence, both models show better generalization performance than standard RVFL model.

Parija et al. [146] proposed minimum variance-based kernel RVFL (MVKRVFL) to identify the seizure and non-seizure epileptic EEG signal. In MVKRVFL, both the total variance of the training data and within class variance are minimized to improve the generalization performance of the RVFL model. The kernel trick is also employed to avoid the hidden layer nodes and activation function (as these parameters are chosen in RVFL). In MVKRVFL, deep long short-term memory (DLSTM) [147] is employed to extract the features from epileptic EEG signals. The empirical evaluation of the DLSTM-MVKRVFL model over EEG data demonstrates that it efficiently classifies seizure and non-seizure movement. In [148], co-trained RVFL (coRVFL) model is proposed in which two feature spaces i.e., randomly projected features and sparse-l_1 norm autoencoder-based features are employed. The coRVFL model utilizes the strength of different feature spaces and improves the generalization performance of the RVFL model. Alzheimer's disease (AD) typically affects the brain's cognitive functions and damages the cells and memory in the brain. Having heterogeneous medical data, it is a challenging task to diagnose it at an early stage. Dai et al. [149] proposed a hybrid model combining features extracted from different modalities and introduced the manifold concept in the RVFL model to enhance the diagnosis process

of AD. Adopting the manifold theory in its optimization process, the RVFL model has the capability of maintaining the geometrical properties of the data while calculating the final output parameters. Literature indicates that maintaining the local and global geometrical (or statistical) properties of the data, manifold leaning-based RVFL models have better generalization performance than standard RVFL.

2.3.3.4 Robust RVFL models

The standard RVFL model employs L_2 norm-based loss function that is sensitive to outliers and hence, affects the generalization performance of the model. The standard RVFL doesn't perform well over noisy datasets, and therefore, one needs to handle the noisy datasets with extra attention. Managing noisy data is a challenging task. Thus, several approaches have been introduced to address such problems. To deal with datasets having noise or outliers and to reduce the complexity of the model so that the generalization performance of the model can be improved, Cui et al. [150] proposed RVFL-based approach, wherein a novel feature selection method is introduced to make the RVFL model more efficient and robust based on the augmented Lagrangian method. The proposed RVFL-FS method can be fitted into a parallel or distributed computing environment. The RVFL-FS method is employed for the indoor positioning system (IPS) as a regression problem to illustrate the proposed idea. RVFL-FS model works on the idea that instead of using all hidden nodes, one should select the hidden nodes to generate robust features. Therefore, the RVFL-FS model is computationally efficient and robust as compared to the standard RVFL model. Samal et al. [151] proposed a robust non-iterative RVFL, i.e., Added activation function based exponentially enhanced robust RVFL (AAERVFL), wherein trigonometric function based exponentially expanded input vector is connected by a weighted direct link from the input layer to the output layer and a new activation function using a weighted linear combination of two activation functions, i.e., the local sigmoid and global Morlet wavelet function, is introduced. The AAERVFL model is examined with five different non-linear systems and three real-world datasets like electricity price prediction, currency exchange rate prediction, and stock price prediction. The experiments demonstrate the superiority of the AAERVFL model over the standard RVFL.

In industrial processes like the mineral grinding process, noisy data with outliers is acquired due to unavoidable circumstances. Dai et al. [152] proposed robust regularised RVFL (RR-RVFL) and its online version as well. Weight calculated from a non-parametric kernel density estimation method is assigned to the empirical error corresponding to each sample. Therefore, the weighting mechanism reduces the negative impact of the outliers over the RR-RVFL model. The experiments over the mineral grinding process demonstrate that the RR-RVFL model is more robust and has better generalization performance than standard RVFL. In RR-RVFL model, weights are assigned using the Gaussian kernel function which can be replaced via other efficient functions such as piece-wise continuous function to develop more robust model. Predicting the stock market movement is a significant task for future investment. Chen et al. [153] fused the two different algorithms, i.e., RVFL and group method of data handling (GMDH) [154], and proposed RVFL-GMDH model consist of many nice properties such as resist noise/outlier effectively, avoiding the over-fitting problem and has better generalization performance compared to standard RVFL model. The RVFL-GMDH model effectively predicts the turning point of the stock price. Iron and steel-making-based industries are famous industries in the modern world, and now data-driven models are being employed for the estimation of molten iron quality (MIQ) in these industries.

Cauchy distribution weighted M-estimation-based robust RVFL [155] model was developed to estimate the molten iron quality. The training data having outliers affects the modeling capability of standard RVFL, so extra care is needed to handle such data. Therefore, weights are assigned to outliers in the data using Cauchy distribution so that their (outliers) contribution to the modeling process can be identified and the negative influence of these outliers can be reduced. Several techniques such as the weighting method [152], and different loss functions [155], have been used with RVFL to improve its robustness and generalization performance. The standard RVFL model with the assumption that all the samples are equally important gives equal weights to each sample for calculating the final parameters but in the real world, noisy datasets with outliers are acquired that might have negative impact on the model performance. Therefore, one needs to take care of these datasets. In [156], intuitionistic fuzzy theory is employed to define membership and non-membership

function to address the above issues with the RVFL model and proposed intuitionistic fuzzy RVFL (IFRVFL) and to check the applicability of IFRVFL, it has been employed for diagnosis of Alzheimer's disease. IFRVFL is a robust and binary classifier. Similar works can be seen in [157, 158]. Therefore, it should be extended to multiclass problems. Hazarika and Gupta [159] proposed robust 1 norm RVFL (1N RVFL) model wherein the optimization problem is solved via newton technique. The 1N RVFL model produces spare outputs and hence, has less number of hidden neurons as compared to the standard RVFL model.

Table 2.1: The summary of activation functions commonly used in RVFL model

S. No.	Activation function	Mathematical formulation
1	Sigmoid	$\theta(x) = \frac{1}{1+e^{-x}}$
2	Sign (Signum)	$\theta(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0 \end{cases}$
3	Rectified linear unit (Relu)	$\theta(x) = \max\{0, x\}$
4	Sine	$\theta(x) = sin(x)$
5	Radbas	$\theta(x) = e^{-x^2}$
6	Hard limit (Hardlim)	$\theta(x) = \begin{cases} 1, & x \leq 0 \\ 0, & \text{otherwise}, \end{cases}$
8	Tribas	$\theta(x) = \max\{1 - \lvert x \rvert,\ 0\}$
9	Hyperbolic tangent (Tanh)	$\theta(x) = \frac{1-e^{-x}}{1+e^{-x}}$
10	Radial basis function (RBF)	$\theta(\mu, \sigma, x) = e^{-\sigma\|x-\mu\|^2}$
11	Multiquadratic	$\theta(\mu, \sigma, x) = \sqrt{\|x-\mu\|^2 + \sigma^2}$
12	Scaled exponential linear units (Selu)	$\theta(x) = \gamma(\max(0, x) + \min(0, \alpha(e^x - 1)))$

2.3.3.5 Kernelized RVFL models

For training the standard RVFL, one needs to determine the number of enhancement nodes and activation functions in advance. Manually determining the optimal range of hidden nodes and the optimal activation function is challenging task. Kernel methods can be used to address the aforementioned issues. Chakraavorti et al. [67] proposed kernel exponentially extended random vector functional link network (KERVFLN) for non-linear system identification. Expanding the dimension of input vector via trigonometric function and utilizing the expended vector in the learning process increase the generalization performance of the RVFL model. Here, the kernel function is used to increase the stability of

standard RVFL, and the inputs are extended using a trigonometric function that improves the generalization performance of the KERVFLN model. Based on the analogy of the teacher-student interaction mechanism, Zhang and Yang [160] proposed RVFL+ and kernel RVFL+ (KRVFL+) model that utilizes the learning using privileged information (LUPI) paradigm in the training process of standard RVFL, and the experiments demonstrate that RVFL+ has better generalization performance compared to standard RVFL model. Moreover, a tight generalization error bound based on the Rademacher complexity is derived for the RVFL+ model and proves the efficiency and effectiveness of the RVFL+ and KRVFL+ models. Several machine-learning approaches have been developed to classify brain images according to brain abnormalities in the medical domain. Machine learning tools help physicians to make decisions. Nayak et al. [161] proposed kernel RVFL (KRVFL) model with a new feature descriptor based on Tsallis entropy and fast curvelet transform to classify brain abnormalities such as brain stroke, degenerative disease, infectious, brain tumor, and normal brain. An efficient hybrid model [162] consists of weighted multi-kernel RVFL network (WMKRVFLN), empirical mode decomposition (EMD) based features, and water cycle algorithm (WCA) was proposed to diagnose and classify the epileptic electroencephalogram (EEG) signals. When number of samples are large enough then Kernel-based RVFL models are not applicable to large-scale datasets.

2.3.3.6 RVFL with Bayesian inference and other techniques

Scardapane and Wang [163] proposed several alternatives to train standard RVFL by exploiting the Bayesian Inference (BI) framework. In the standard RVFL model, the optimal output weights are generally calculated via (regularized) least square method but here (in [163]), the probability distribution of the output weights is derived. The Bayesian approach has several advantages, i.e., additional prior information can be introduced in the training process of standard RVFL and the capability of automatically inferring hyper-parameters from given data, etc. Experimental results demonstrate that the Bayesian Inference (BI) based approaches are better than least square approaches (as in standard RVFL). The BI approach to train the RVFL model gives a new area of research to develop robust RVFL model. Introducing hybrid regularization term with L_2 and L_1 norm into standard RVFL,

Ye et al. [164] proposed $L_2 - L_1$ RVFL model that overcomes the stability and sparsity issue of the standard RVFL and gives an iterative algorithm to train $L_2 - L_1$ RVFL model. Alalimi et al. [165] employed the Spherical Search Optimizer (SSO) algorithm to optimize the RVFL model and named it the SSO-RVFL model. The SSO algorithm improves the parameters of the standard RVFL model. Dai et al. [166] incorporated the LUPI paradigm into the incremental RVFL (IRVFL) model and proposed IRVFL+ that has strong theoretical foundation. IRVFL+ model has been trained via two approaches. The first one is named IRVFL-I+ which focuses on the speed of the model and another one is IRVFL-II+ which focuses on accuracy. Incremental learning-based RVFL model solves the problem of constructing an appropriate RVFL model. Here, IRVFL-II+ is computationally expensive as compared to standard RVFL and IRVFL-I+ models. In [167], a model for artificially intelligent diagnosis that uses privileged information to learn was proposed to help with ELN differential diagnosis when dealing with single- or dual-modal picture data. In order to create a more effective unmanned aerial vehicle automatic target recognition system, [168] research suggests two unique machine learning methods, namely Random Vector Functional Link Forests and Extreme Learning Forests. For EEG-based driving fatigue detection, in [68], an auto-weighting incremental random vector functional link (AWIRVFL) network model that combines incremental learning and online regression prediction. Although AWIRVFL outperformed several deep learning models, its network topology is still shallow, which restricts its feature learning capacity in describing the underlying characteristics of EEG data.

In another approach, the RVFL-MO method [169] optimizes the RVFL model via the mayfly optimization (MO) algorithm to predict the performance of solar photovoltaic thermal collector combined with the electrolytic hydrogen production system. Experiments demonstrate that the RVFL-MO model performs better than standard RVFL. Elsheikh et al. [170] proposed an enhanced RVFL model with equilibrium optimizer (EO), i.e., RVFL-EO, to predict kerf quality indices during CO_2 laser cutting of polymethylmethacrylate (PMMA) sheets. The equilibrium optimizer (EO) algorithm is employed to obtain the parameters of the RVFL model that enhance its generalization performance. Several statistical tests were used to compare RVFL-EO with the standard RVFL model, and the results indicate the superior performance of the RVFL-EO model. A novel classification method [141] based

on MobileNet and three feed-forward neural networks with random weights, i.e., extreme learning machine (ELM), Schmidt neural network (SNN), and RVFL network, to classify brain magnetic resonance image (MRI) image was proposed. Here, MobileNetv2 is employed to extract features from the input brain image, and then the classification task is executed via ELM, SNN, and RVFL models, respectively. The experimental results reveal that the MobileNet-RVFL-CBA method performs better than other proposed MobileNet-ELM-CBA, MobileNet-SNN-CBA methods and compared state-of-the-art methods.

Naive Bayes classifier has the capability to handle the mixed data containing categorical and numerical attributes. On the other hand, the one-hot encoding technique is used to deal with categorical features in neural networks. To avoid the use of one-hot encoding with neural network, Ruz and Henriquez [171] proposed a two-stage learning approach based on the RVFL model and Naive Bayes classifier, i.e., RVFL-NB, to handle the mixed data. In the first stage, the Naive Bayes classifier is employed to compute the posterior probabilities for each class, and in the second stage, the RVFL model is trained using as inputs the continuous features and including as additional hidden units the posterior probabilities obtained in the previous step. Single hidden layer feed-forward neural networks face a challenge, i.e., how to choose a number of hidden neurons in the hidden layer because this choice leads to underfitting and overfitting phenomena. To address this issue, a non-iterative method [172] for pruning hidden neurons with random weights was proposed. The pruning method is based on Garson's algorithm and was employed on three neural networks, i.e., single hidden layer neural network with random weights (RWSLFN), RVFL, and ELM, to increase their generalization performance.

Parsimonious RVFL (pRVFL) [173] model was proposed for the data stream and hence, performs better than the standard RVFL model. pRVFL model has flexible and adaptive working principle wherein its structure is automatically generated and pruned. In the optical fiber pre-warning system (OFPS), most of the feature extraction methods are quested from the view of the time domain. To address this issue, using multi-level wavelet decomposition, Wang et al. [174] extract intrusion signal features of the running, digging, and pick mattock in the frequency domain and then for considering the feature of each intrusion type, the average energy ratio of different frequency bands is obtained. Finally, the RVFL model

is trained for the classification and identification of the signal. The results demonstrate that RVFL correctly classifies the different intrusion signals. El-Said et al. [175] conducted experiments with four machine learning algorithms, i.e., support vector machine (SVM) [176], K-nearest neighbour (K-NN) [177], sequential minimal optimization regression [178, 179] and RVFL model, to predict the air injection effect on the thermohydraulic performance of shell and tube heat exchanger. The experimental analysis reveals that the RVFL model outperforms compared models with excellent accuracy and better generalization performance.

Borah and Gupta [32] proposed unconstrained convex minimization based implicit Lagrangian twin RVFL for binary classification (ULTRVFLC) for addressing the overfitting issue in the standard RVFL and hence, has better generalization capability as compared to standard RVFL model. Unlike TWSVM and twin ELM (TELM), in ULTRVFLC, three iterative convergent schemes are employed to make the model computationally efficient. The least-square twin SVM (LSTSVM) [23] has been a successful classifier, and it works on the original feature space. On the other hand, the RVFL model works on both original and randomized features. Ganaie and Tanveer [69] proposed a novel LSTSVM model with enhanced features obtained from the pre-trained RVFL model and hence, improved the generalization performance of the baseline model. Prediction of international oil prices has become a hot topic in the field of energy system modeling and analysis. Tang et al. [180] proposed a new technique introducing a multi-scale forecasting methodology with multi-factor search engine data (SED). Incorporating the informative SED, the multi-scale relationship with oil price is explored, and four machine learning models, i.e., ELM, RVFL, linear regression (LR), and backpropagation neural network (BPNN), are employed in this task. Mary et al. [181] employed standard RVFL in the image retrieval (IR) framework for better performance. To address the instability issue in the sliding mode control system, Zhou and Wu [182] proposed an adaptive fuzzy RVFL (FRVFL), wherein self-mapping between fuzzy rules and hidden layers is employed and adaptive rules are also employed to achieve self-adjustment for the output weights. FRVFL model combines RVFL with the dynamic fuzzy system to improve its generalization performance. To address the threats issues in android malware detection tools, in [183], a novel technique using RVFL model with artificial jellyfish search (JS) optimizer algorithm for selecting the optimal features of

android malware datasets, i.e., RVFL+JS, has been proposed. The JS algorithm reduces the redundant and irrelevant features from the data that handle the storage and time complexity issue and hence, improves the generalization performance of the RVFL+JS model. In [184], ResNet18 was used to extract wrinkle image features, and an RVFL algorithm optimized by the TSA algorithm based on logistic maps and opposition-based learning was proposed for evaluating fabric wrinkle level. This was done to address the issues of low accuracy and low efficiency in evaluating the wrinkle degree by visual perception and the shortcomings of the current artificial neural networks in evaluating the wrinkle level. However, the amount of fabric samples at some levels was insufficient, and just a few different fabric types were included in the study's fabric samples. Label distribution learning (LDL), as opposed to multi-label learning (MLL), can reflect the importance of pertinent labels in samples, which is why many LDL studies have lately been appearing. In [185], a unique LDL framework based on RVFL is proposed, which can efficiently and precisely handle the live data stream.

2.3.3.7 Imbalance learning based on RVFL model

The class imbalance problem occurs when one class has small samples compared to other classes. Standard RVFL is not capable of handling imbalanced data. Cao et al. [186] proposed improved fuzziness-based RVFL (IF-RVFL) where synthetic minority over-sampling technique (SMOTE) [187] is combined with fuzziness-based RVFL model [188]. The experiments on the real-life liver disease dataset demonstrate that the IF-RVFL model performs better than the standard RVFL and F-RVFL models. To diagnose the power quality disturbance (PQD), Sahani and Dash [189] proposed class-specific weighted RVFL (CSWRVFL). Here, a novel signal decomposition technique- reduced sample empirical mode decomposition- is proposed to extract the highly correlated monocomponent mode of oscillations. Hilbert transforms (HT) extracted the two effective power quality indices are extracted from Hilbert transforms (HT). Finally, the combined framework RSHHT-CSWRVFL is applied for online monitoring of the power quality disturbances (PQDs) with better classification accuracy.

2.3.3.8 Multi-label classification based on RVFL model

In a traditional classification problem, each sample is associated with only one target label from a set of labels. However, each sample can be related to more than one label in multi-label classification problems. There are several fields, i.e., medical diagnosis, music categorization, etc., wherein multi-label data is produced. Chauhan and Tiwari [190] extended the standard RVFL for the multi-label task. In [190], randomization-based neural networks, i.e., multi-label RVFL (ML-RVFL), multi-label kernelized RVFL (ML-KRVFL), multi-label broad learning system (ML-BLS), and multi-label fuzzy BLS (ML-FBLS) were proposed to handle the multi-label classification problems. Here, the ML-KRVFL model performs better than other compared models.

Table 2.2 shows the summary of the shallow RVFL model and its variants. The table gives a highlight of the journey of shallow RVFL in tabular form with variants name, activation functions, hyper-parameter optimization methods/solutions, and finally their applications. In summary, researchers have developed several variants of RVFL model using various methods such as different kind of initialization techniques [12, 120], kernel methods [67, 160], manifold learning [144, 145], fuzzy theory [156] and so on. Kernel-based RVFL models [161, 162] are robust, stable, and have better generalization performance than standard RVFL model. There is no need to choose activation functions and hidden nodes in kernel-based RVFL models, however, these models are not suitable for large-scale datasets (when N is large enough). Standard RVFL doesn't consider geometric information of the data while calculating the output weights whereas, using this kind of information, RVFL variants with manifold learning theory have better generalization than standard RVFL.

Table 2.2: The summary of shallow RVFL models

Year	Literature	Model description	Activation function	Hyper-parameter optimization or Solution	Application
2022	Ganaie et al. [191]	MVRVFL+	Relu	Closed form	Classification
2022	Malik et al. [156]	Intuitionistic fuzzy RVFL (IFRVFL)	Selu, relu, sigmoid, sin, hardlim, tribas, radbas, sign function	Closed form	Alzheimer's disease diagnosis
2022	Hazarika and Gupta [159]	1-norm RVFL (1N RVFL)	Sine, relu	Newton technique	Classification problem
2021	Samal et al. [151]	A non-iterative robust AAERVFL	Sigmoid, global morlet wavelet functions	Closed form	Nonlinear system identification
2021	Parija et al. [146]	Minimum variance based kernel RVFL (MVKRVFL)	-	Closed form	Epileptic EEG signal classification
2021	Chen et al. [153]	Group method of data handling (GMDH) based RVFL (RVFL-GMDH)	Sigmoid	Iterative method	Stock price prediction
2021	Zhou and Wu [182]	Adaptive fuzzy RVFL (FRVFL)	-	Iterative method	Slide mode control for manipulators
2021	Zhang et al. [192]	Reinforced fuzzy clustering-based rule model (RFCRM)	-	Iterative method	Regression problem
2021	Elkabbash et al. [183]	RVFL+JS	-	Jellyfish search optimizer	Android malware classification
2021	Guo et al. [144]	Sparse Laplacian regularized RVFL (SLapRVFL)	-	Iterative method	Assessing dry weight of hemodialysis patients
2021	Alalimi et al. [165]	Spherical search optimizer based RVFL (SSO-RVFL)	Tribas, sign, hardlim, radbas, sin, sig	Spherical search optimization algorithm	Prediction of oil production in China
2021	Gao et al. [64]	Walk-forward EWT based RVFL (WFEWT-RVFL)	-	Closed form	Time series forecasting
2021	Tanveer et al. [137]	TBSVM-FL, TWKSVC-FL, LSTWKSVC-FL, RELSTSVM-FL, LSTSVM-FL	Radbas	Closed form (LS)	Classification problem
2021	Abd Elaziz et al. [169]	Mayfly optimization (MO) algorithm based RVFL (RVFL-MO)	Sign, hardlim, sig, tribas, radbas	Mayfly optimization algorithm	solar photovoltaic thermal collector combined with electrolytic hydrogen production system
2021	Elsheikh et al. [170]	Equilibrium optimizer based RVFL (RVFL-EO)	Sign, hardlim, sig, tribas, radbas	Equilibrium optimization algorithm	Laser cutting parameters for polymethylmethacrylate sheets
2021	Zayed et al. [193]	RVFL-CHOA	Sign, tribas, sigmoid, hardlim, radbas	Chimp optimization algorithm (CHOA)	Solar power forecasting
2021	Ganaie et al. [148]	Co-trained RVFL (coRVFL)	-	Closed form	Classification problem
2021	Dash et al. [194]	Empirical wavelet transform based robust minimum variance RVFL (EWT-RRVFLN)	Local sigmoid function, global morlet wavelet	Closed form (LS)	Short term solar power forecasting
2021	Gao et al. [64]	Walk forward empirical wavelet transformation based RVFL (WFEWT-RVFL)	-	-	Time series forecasting
2021	Dai et al. [166]	Incremental learning paradigm with privileged information for random vector functional-link networks: IRVFL+	Sigmoid, sine, triangular, radial	Incremental learning	Classification and regression problems
2020	Ganaie and Tanveer [69]	LSTSVM classifier with enhanced features from pre-trained RVFL	Radbas	Closed form	Classification problem
2020	Ganaie et al. [145]	Total-Var-RVFL, Class-Var-RVFL	Relu	Closed form	Classification problem
2020	Hazarika and Gupta [195]	Wavelet-coupled RVFL (WCRVFL) network	Relu, sigmoid	Closed form (LS)	COVID-19 cases forecasting
2020	Dudek [14]	Standard RVFL	Sigmoid	Closed form (LS)	Regression problem
2020	Chakravorti and Satyanarayana [67]	Exponentially extended RVFL network (ERVFLN), Kernel ERVFLN (KERVFLN)	Tanh(.)	Closed form (LS)	Nonlinear system identification

2020	Cao et al. [134]	-	Sigmoid, radial basis function (RBF), sine		Classification problem
2020	Lu et al. [141]	MobileNet-RVFL-CBA, MobileNet-ELM-CBA and MobileNet-SNN-CBA	-	Closed form (LS)	Brain MRI image classification problem
2020	Rasheed et al. [135]	Standard RVFL	Sine, hardlim, sigmoid, tribas, radbas, sign	Closed form (LS)	Respiratory motion prediction
2020	Zhang and Yang [160]	RVFL+, kernel RVFL+	Sigmoid, sine, hardlim, triangular basis function (TBF), radial basis function (RBF)	Closed form (LS)	Classification and regression problems
2020	Ye et al. [164]	$L_2 - L_1$-RVFL	Sigmoid	Iterative method	Classification Problem
2020	Abd Elaziz et al. [196]	MPA-RVFL	Sigmoid, sine, hardlim, tribas, radbas	Marine predators algorithm (MPA)	Tensile behavior prediction
2020	Essa et al. [197]	RVFL-AEO	Sign, tribas, sigmoid, hardlim, radbas	Artificial ecosystem-based optimization (AEO) algorithm	Forecasting power consumption and water productivity of seawater
2020	Sharshir et al. [198]	Firefly algorithm based RVFL (FA-RVFL)	Sigmoid, radbas, hardlim, sine, sign, tribas	Firefly algorithm	Thermal performance and modeling prediction of developed pyramid solar
2020	Pan et al. [139]	Jaya-RVFL	-	Jaya algorithm	Transient stability assessment of power systems
2020	Parija et al. [162]	Empirical mode decomposition (EMD) and water cycle algorithm (WCA) based weighted multi-kernel RVFL network (WMKRVFLN) (WCA-EMD-WMKRVFLN)	-	Closed form (LS)	Epileptic EEG signal classification
2019	Hussein et al. [199]	Moth search algorithm based RVFL (MSA-RVFL)	-	Closed form (LS)	Water quality analysis
2019	Zhang et al. [120]	Sparse autoencoder with l_1 norm based RVFL (SP-RVFL)	Sigmoid	Closed form (LS)	Classification problem
2019	Nayak et al. [161]	kernel RVFL (KRVFL)	-	Closed form (LS)	Brain abnormalities detection
2019	Zhou et al. [200]	Improved orthogonal incremental RVFL (I-OI-RVFL)	Sigmoid	Iterative method	Data modeling
2019	Ruz and Henríquez [171]	Naive Bayes classifier based RVFL (RVFL-NB)	Sigmoid	closed form	Classification problem of mixed data
2019	Cao et al. [136]	Standard RVFL	Sigmoid	Closed form (LS)	Classification problem
2019	Bisoi et al. [119]	Variational mode decomposition based RVFL (VMD-RVFL)	Tanh	Closed form (LS)	Crude oil price forecasting
2019	Borah and Gupta [32]	Unconstrained convex minimization based implicit Lagrangian twin RVFL for binary classification (ULTRVFLC)	Multiquadratic	Newton-Armijo stepsize method	Classification problem
2019	Sahani and Dash [189]	Class-specific weighted RVFL (CSWRVFL)	Tanh	Closed form (LS)	Power quality disturbances
2019	Fan et al. [138]	Random subspace fisher linear discriminant (FLD) based RVFL	-	Closed form (LS)	Image steganalysis
2018	Wang et al. [174]	Standard RVFL	-	Closed form (LS)	Optical fiber pre-warning system
2018	Pratama et al. [173]	Parsimonious RVFL (pRVFL)	-	FWGRLS method	Data stream
2018	Henríquez and Ruz [172]	Neural networks with random weights (NNRW)	Sigmoid	Closed form (LS)	Regression and classification problem
2018	Vuković et al. [13]	Orthogonal polynomial expanded RVFL (OPE-RVFL)	Tansig, logsig, tribas	Closed form (ridge regression/ Moore-Penrose pseudoinverse)	Regression problem
2018	Dash et al. [201]	Standard RVFL, regularized online sequential network (ROS-RVFL)	Radbas	Closed form (LS)	Indian summer monsoon rainfall prediction
2018	Nhabangue and Pillai [202]	Empirical mode decomposition based improved RVFL (EMD-IRVFL)	-	Closed form (LS)	Wind speed forecasting

2017	Cui et al. [150]	Feature selection based RVFL (RVFL-FS)	-	Alternative direction method of multiplier (ADMM) algorithm	Fingerprinting based indoor positioning system
2017	Scardapane et al. [163]	Bayesian inference RVFL (B-RVFL)	-	Bayesian inference algorithm	Classification problem
2017	Cao et al. [121]	Standard RVFL	Sigmoid	Closed form (LS)	Regression problem
2017	Dai et al. [152]	Robust regularized RVFLN (RR-RVFLN), online robust regularized RVFL (ORR-RVFLN)	-	Closed form (LS)/ Iterative method	Industrial application
2017	Zhou et al. [155]	Cauchy distribution weighted M-estimation-based robust RVFL (Cauchy-M-RVFLN)	Sigmoid	Iterative method	Blast furnace iron-making process
2017	Xu et al. [203]	Kernel RVFL (K-RVFL)	-	Closed form (LS)	Thermal process
2017	Dai et al. [149]	Manifold learning based RVFL	-	Closed form (LS)	Analysis of alzheimer's disease
2016	Zhang et al. [204]	Multivariable incremental RVFL (M-I-RVFLN)	-	Iterative method	Molten iron quality prediction
2016	Zhang and Suganthan [12]	Standard RVFL	Sigmoid, sine, hardlim, tribas, radbas, sign	Closed form (ridge regression/ Moore-Penrose pseudoinverse)	Classification problem
2015	Zhou et al. [205]	Online sequential RVFL (OS-RVFL)	-	Iterative method	Molten iron quality prediction
2015	Ren et al. [206]	Standard RVFL	Logistic sigmoid	Closed form (LS)	Wind power ramp detection

2.3.4 Semi-supervised methods based on RVFL model

There are datasets in which small number of samples are labeled in many applications. RVFL variants have been successfully employed in diverse domains, i.e., classification and regression, etc. However, there is very little research on solving semi-supervised learning problems with the RVFL model. Table 2.3 summarizes the RVFL models developed for semi-supervised learning. Peng et al. [207] proposed a joint optimized semi-supervised RVFL model, i.e., JOSRVFL, in which a novel approach is used to optimize the objective function of the JOSRVFL model. There are many techniques in machine learning to improve the generalization performance of a model, and fuzzy theory is one of them. In [188], a novel fuzziness-based RVFL model has been proposed for a semi-supervised learning problem. Inspired by transductive SVM [208], Scardapane et al. [209] proposed a transductive RVFL (TR-RVFL) model that defines box-constrained quadratic (BCQ) problem solvable in polynomial time. The TR-RVFL model performs better than many state-of-art algorithms based on the manifold regularization (MR) theory. In [210], two algorithms, i.e., horizontally distributed semi-supervised learning (HDSSL) and vertically DSSL, were proposed. Both algorithms are based on the RVFL model and alternating direction method of multipliers (ADMM) strategy. The HDSSL and VDSSL algorithms solve DSSL problems with horizontally and vertically partitioned data, respectively. Therefore, the RVFL model performs well in semi-supervised problems.

Table 2.3: The summary of semi-supervised RVFL models

Year	Literature	Model description	Activation function	Hyper-parameter optimization or Solution	Application
2020	Peng et al. [207]	JOSRVFL and JOSELM	-	Iterative method	Classification problem
2020	Xie et al. [210]	Horizontally distributed semi-supervised learning (HDSSL), vertically DSSL (VDSSL)	-	Iterative method	Classification problem
2017	Cao et al. [188]	F-RVFL	Sigmoid	Closed form	Classification problem
2016	Scardapane et al. [209]	TR-RVFL	Sigmoid	Closed form	Classification problem

2.3.5 Clustering methods based on RVFL model

Clustering is an unsupervised learning problem where samples are categorized into clusters (groups) based on their similarities. In the literature, there are various clustering techniques, i.e., point-based clustering methods [211] and plane-based clustering [212, 213],

etc. Zhang et al. [214] proposed an unsupervised discriminative RVFL (UDRVFL) model for the clustering problem. To capture the local information within data, the local manifold learning concept has been used while global biased knowledge of the data has also been considered so that data can be clustered in an optimal manner.

2.3.6 Ensemble frameworks based on RVFL model

Ensemble learning utilizes multiple learning algorithms, which are named base learners. The performance of a single RVFL model is often unstable because of the random nature of its hidden features. To improve the model's stability and performance, it can be beneficial to average the outputs from multiple RVFL models that each have different hidden features. This approach is commonly known as ensemble learning. Ensembles of randomized models, such as random forests, are effective in reducing the variance resulting from the random feature space. Thus, researchers working with RVFL models have explored the development of ensemble RVFLs to improve the model's stability and performance. In general, there are two steps in ensemble learning. First, a pool of base learners is constructed parallel or sequential. Second, the base learners are combined for decisions according to some rules. Therefore, the ensemble RVFL method either trains multiple RVFLs or utilizes a meta-RVFL to connect the outputs of the base learners. Table 2.4 summarizes the representative literature about the ensemble RVFL models.

2.3.6.1 Ensemble RVFL-based on bagging

Bagging is short for bootstrap aggregating, which trains the base leaner using a subset of the training data. The subsets are drawn randomly with replacement [215]. For instance, the base learners of a rotated forest are replaced by RVFLs for classification problems [216]. Bagging generates different subsets with different features to train the RVFL with different structures [71].

2.3.6.2 Ensemble RVFL-based on boosting

Boosting constructs the ensemble RVFL incrementally by paying more attention to the samples that are not correctly learned by the base RVFL. In [217], Maximum Relevance Minimum Redundancy is utilized to select features, and the ensemble RVFL is constructed using the Adaboost scheme.

2.3.6.3 Ensemble RVFL-based on stacking

Stacking refers to training a meta-learner that works on the outputs from all base learners. In [218], a meta-RVFL is trained to combine the results from all the base RVFL networks with different activation functions. In [219], an individual RVFL network is trained to forecast each sub-series generated by the decomposition and an incremental RVFL is introduced to aggregate all forecasts. In [220], the short-term load is decomposed by a two-stage decomposition, and an incremental RVFL aggregates the forecasts from each sub-series RVFL.

2.3.6.4 Ensemble RVFL-based on decomposition

Another branch of ensemble RVFL is the decomposition-based ensemble framework shown in Figure 2.4. The time series is decomposed into sub-series carrying different frequencies and each sub-series is modeled by an individual RVFL network. Finally, the aggregation of all forecasts is the output. There are many mature signal decomposition algorithms, such as the empirical mode decomposition (EMD) [221], bi-variate empirical mode decomposition (BEMD) [222], ensemble empirical mode decomposition (EEMD) [223], complete ensemble empirical mode decomposition (CEEMD) [224], hybrid decomposition [225, 226] and other algorithms [227, 228]. For the decomposition block, there are many hybrid ensemble RVFL with signal decomposition algorithms, such as EMD [125, 219, 229], EEMD [230, 231, 232, 233], CEEMD [234, 235, 236] and two-stage decomposition [220]. In [220], the discrete wavelet transform (DWT) is utilized to decompose the modes generated by EMD into sub-series and an incremental RVFL is trained to aggregate the forecasts from all sub-series. Technical indicators are also utilized to augment the decomposed fea-

tures for stock forecasting and stock trend classification [225, 226].

2.3.6.5 Ensemble weights

Determination of the ensemble weights is crucial for the final performance. If large weights are assigned to a bad base RVFL, it is a disaster for the overall performance. Most of the ensemble RVFL employs the equal-weight scheme [125, 229, 230, 231, 232, 233, 234, 235]. Besides the equal-weight scheme, different algorithms are proposed and applied to learn such unequal ensemble weights. For instance, evolutionary algorithms are adopted to learn, and ensemble weights, as well as the base learners' parameters [71, 237, 238]. A negative correlation learning strategy is utilized to learn the ensemble weights in [239, 240].

2.3.6.6 Diverse model pool

Besides the above ensemble RVFL, whose base learners are all RVFL networks, some researchers utilize multiple machine learning models, including RVFL, to construct the ensemble pool, which increases the models' diversity. For instance, ELM, RVFL, and Schmidt neural networks (SNNs) are trained on the same features generated from a DL model in [239]. Finally, the majority voting mechanism combines the outputs from these three neural networks. In [241], the successive projections algorithm is utilized to build ensemble ELM, RVFL, and feedforward networks with random weights. In [230], ELM, BPNN, and RVFL are employed to forecast each sub-series after decomposition. In [242], the model pool consists of RVFL and ELM, which are trained in an offline fashion first, and only a subset of them is randomly selected for online updating. In [243], the fast Fourier transform and Relief algorithms extract features for ensemble ELM and RVFL.

2.3.6.7 Other diversity strategies

Besides the diversity strategies of typical ensemble learning, novel diversity strategies specifically designed for RVFL networks are investigated [71, 244]. In [244], different RVFLs' enhancement features are initialized according to different distributions. Five novel diversity strategies, such as data quantity diversity, sampling interval diversity, parameter

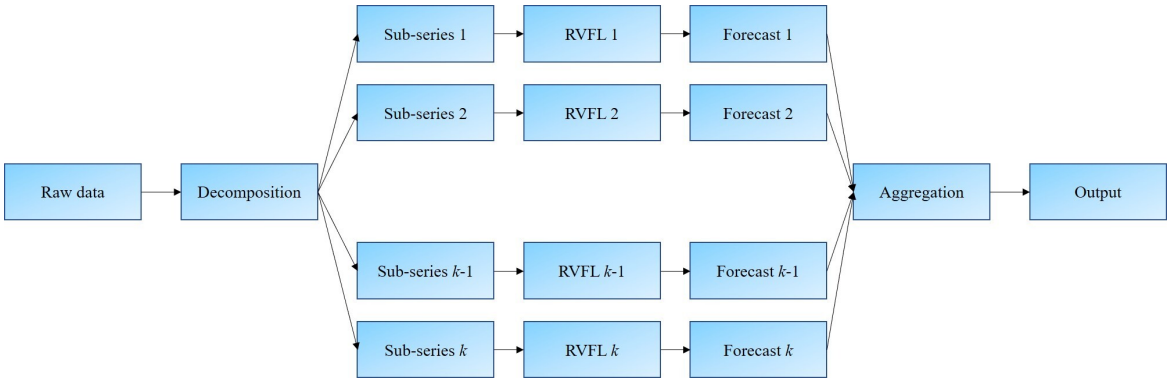


Figure 2.4: The architecture of decomposition-based ensemble RVFL.

diversity, ensemble number diversity, and ensemble method diversity, are proposed and investigated in [71]. Zhang and Suganthan [245] propose a novel and efficient strategy to increase the diversity of ensemble RVFL. A single RVFL is trained, and its hyper-parameters are optimized by cross-validation. Then the other RVFLs' hyper-parameters are generated by adding a noisy deviation to the optimal value.

2.3.6.8 Others

RVFL also helps split the dataset into subsets, and each subset is learned by the oblique decision tree [127]. In [126], a cascaded ensemble RVFL where the shallow layers' RVFL generate predicted values for the successive layers. Malik et al. [246] proposed a novel ensemble model (en-efRVFL) which has extended features based RVFL model as a base model and the output of each base model is integrated by averaging method. The en-efRVF model has three kinds of features, i.e., original features, supervised randomized features (newly generated), and unsupervised randomized features, and therefore generates more accurate and diverse base model in the ensemble. In [51], conv-eRVFL model combines the CNN model with the ensemble RVFL model and implements it for the diagnosis of Alzheimer's Disease. An ensemble of RVFL models is fed with the features that an eight-layer trained CNN derives from multiple layers. The s-membership fuzzy function is incorporated into the RVFL network as an activation function to help deal with outliers. In order to reach a judgment, the outputs of all the bespoke RVFL classifiers are averaged and supplied to the RVFL classifier.

Table 2.4: Summary of ensemble RVFL models.

Year	Literature	Model pool	Diversity strategy	Ensemble strategy	Application
2022	Malik et al. [246]	Extended feature RVFL (efRVFL)	-	Averaging	Classification Problem
2021	Yu et al. [71]	RVFL	Data quantity diversity Sampling interval diversity Parameter diversity Ensemble number diversity Ensemble method diversity	Averaging Adaboost RVFL	Forecasting
2021	Lu et al. [247]	RVFL,ELM and Schmidt NN	-	Majority voting mechanism	Cerebral microbleed diagnosis
2021	Hu et al. [70]	RVFL	Solutions' angle	Evolutionary algorithm	Forecasting
2021	Malik et al. [216]	RVFL	Bagging	Rotated forest	Classification
2020	Liu et al. [244]	RVFL	Different distributions	Majority voting mechanism	Classification
2020	Tahir et al. [218]	RVFL	Stacking	RVFL	Multichannel fall detection
2019	Chen et al. [248]	RVFL	Bootstrap	Game theory	Classification
2019	Musikawan et al. [249]	RVFL	Metaheuristic algorithm	Liear regression	Regression
2019	Xia et al. [243]	RVFL and ELM	Diversity strategy	Ensemble strategy	IGBT open-circuit fault diagnosis
2018	Shi et al. [126]	RVFL	Diversity strategy	Ensemble strategy	Parkinson's disease diagnosis
2018	Mesquita et al. [241]	RVFL, ELM and randomized feedforward NN	Successive Projections Algorithm	Successive Projections Algorithm	Regression
2018	Lu et al. [237]	RVFL	De-correlation	Negative correlation learning	Forecasting
2018	Katuwal and Suganthan [127]	Oblique decision tree	The RVFL splits the data into subsets.		Classification
2018	Li et al. [232] Qiu et al. [229] Sun et al. [233] Tang et al. [230, 231] Zhang et al. [125]	RVFL	Signal decomposition	Summation	Forecasting
2017	Qiu et al. [234]	Kernel ridge regression	Signal decomposition	RVFL	Forecasting
2017	Miskony and Wang [240]	RVFL	De-correlation	Negative correlation learning	Prediction interval

2.3.7 Deep architectures based on RVFL model

The success of deep learning (DL) is based on the hierarchical representations of the raw data [250]. DL stacks multiple hidden layers and optimizes the weights using any variants of the backpropagation algorithm. With the help of deep architecture, the DL can extract multi-scale features automatically. Inspired by the idea of DL, deep RVFL with multiple enhancement layers has been proposed [57].

Table 2.5 summarizes the representative literature about different deep RVFL networks. The main distinction among them is the utilization of direct links. Some literature only utilizes the direct link to connect the input layer to the output layer to assist in learning the linear patterns [251, 252]. Some literature utilizes direct links to connect all hidden layers and output layers [57, 59, 83, 253, 254]. Therefore, the raw features are utilized to provide clean information to each level's representation. Furthermore, Katuwal and Suganthan [58] proposed dense connections of all hidden layers. Another characteristic of the deep RVFLs is the number of output layers. Multiple output layers benefit from the ensemble learning and improve the performance [57, 59, 83, 253, 254].

The main research problem of deep RVFL is the architecture design. This section reviews the state-of-the-art deep RVFL architectures in detail. The state-of-the-art deep RVFL architectures are shown in Figure 2.5. These architectures can be classified into three categories, the stacked deep RVFL, the hybrid deep RVFL, and the ensemble deep RVFL

(edRVFL).

Table 2.5: Summary of deep RVFL models.

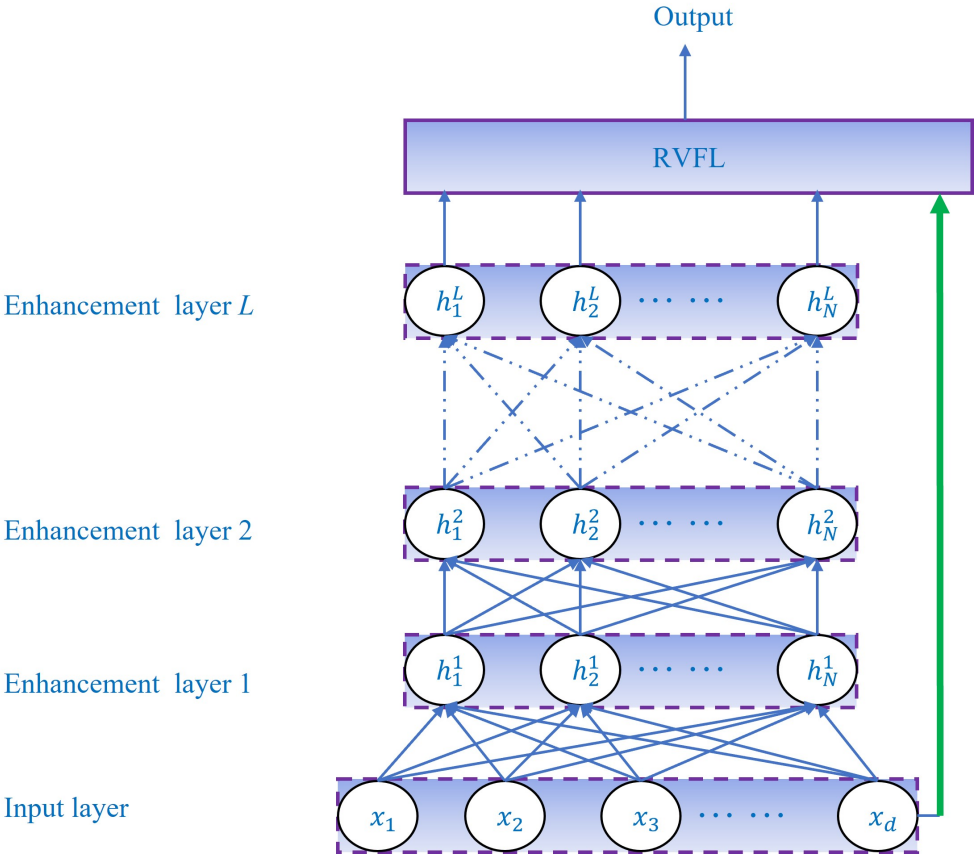
Year	Literature	Direct link	Output layer	Activation function	Application
2022	Hu et al. [255]	Random skip connections	Multiple output layers	Four activation functions	Regression
2022	Du et al. [256]	From input to each hidden and output layer	Dynamic ensemble	*sigmoid*	Forecasting
2022	Yu et al. [257]	From input to each output layer	Multiple output layers	Na	Landslide displacement prediction
2022	Hu et al. [258]	Automatic search	Multiple output layers	Automatic search	Classification
2022	Shi et al. [259]	From input to each hidden and output layer	**Multiple output layers**	*Sigmoid*	Semi-supervised classification
2022	Shi et al. [59]	From input to each hidden and output layer	**Multiple output layers**	*Sigmoid*	Forecasting
2022	Gao et al. [83]	From input to each hidden and output layer	**Multiple output layers**	*Sigmoid*	Forecasting
2022	Ganaie and Tanveer [260]	From input to each hidden and output layer	Multiple output layers	Different activations	Diagnosis of Alzheimer disease
2022	Malik and Tanveer [261]	From input to each hidden and output layer	Multiple output layers	Different activations	Diagnosis of Alzheimer disease
2021	Sharma et al. [61]	-	-	s-fuzzy activation function	Diagnosis of Alzheimer disease
2021	Shi et al. [57]	From input to each hidden and output layer	Multiple output layers	*Sigmoid*	Classification
2021	Cheng et al. [254]	From input to each hidden and output layer	Multiple output layers	Five activation functions	Time series classification
2021	Dai et al. [262]	Without direct link	Last hidden layer's features	Sign	SAR target recognition
2019	Katuwal and Suganthan [58]	Densely connected to hidden layers	Last hidden layer's features	*Sigmoid*	Classification
2019	Katuwal and Suganthan [58]	From input to output layer	Last hidden layer's features	*Sigmoid*	Classification
2018	Henríquez and Ruz [252]	From input to output layer	Last hidden layer's features	*Sigmoid*	Sentiment classification
2017	Zhang and Suganthan [251]	From input to output layer	Last hidden layer's features	*Relu*	Visual tracking

2.3.7.1 Stacked deep RVFL

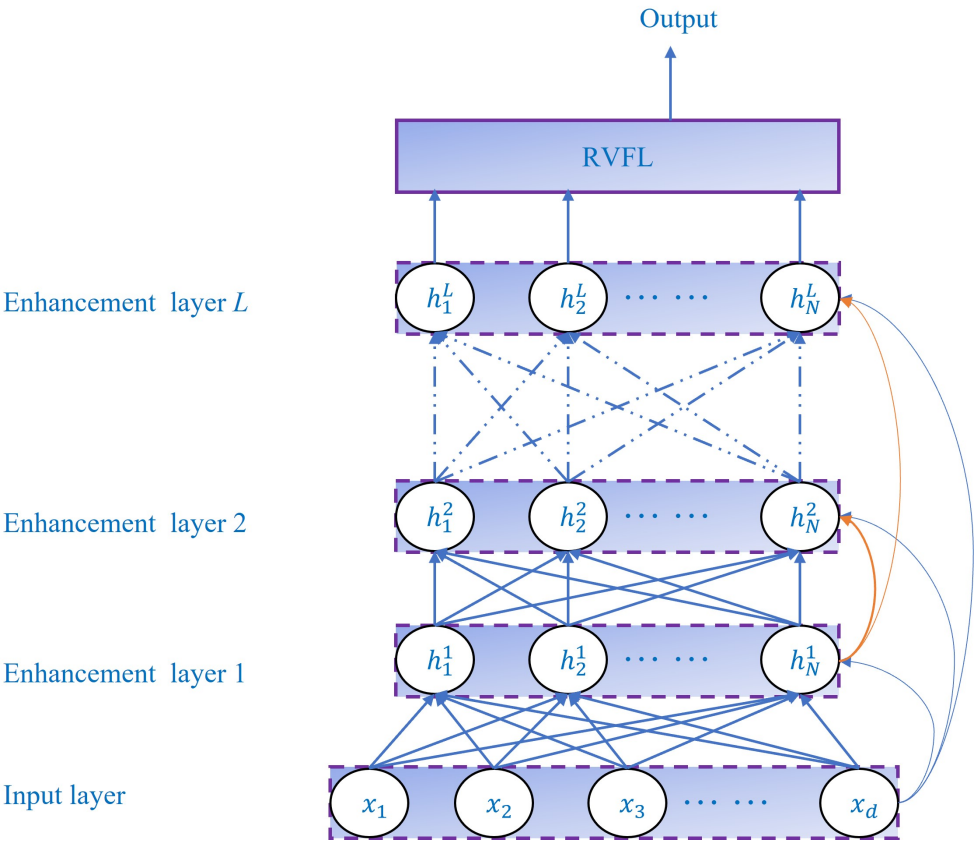
The stacked deep RVFL utilizes multiple enhancement layers to achieve multi-scale feature extraction. The consistent characteristic of the deep RVFL architectures is the multiple stacked enhancement layers. The main difference among them lies in how the direct links are connected. The sdRVFL is the most straightforward deep RVFL architecture, which stacks multiple hidden layers, and the direct link only connects the input layer and output layer [58, 251, 252]. The convolutional deep RVFL also establishes the direct connection in this fashion, but the enhancement layers are convolution layers with random weights [251, 262]. However, the direct links are not equipped with hidden layers, which weakens the unsupervised feature extraction. The direct links are densely connected to the hidden layers to guide the random features' generation, and the sdRVFL(dense) is proposed [58]. The above architectures only utilize the last hidden layer's features for decision, which may lose valuable information from the intermediate layers. A dRVFL is proposed to take advantage of the rich information from all hidden layers in [57]. A reservoir layer is utilized to extract features for the following deep RVFL [263].

2.3.7.2 Ensemble deep RVFL

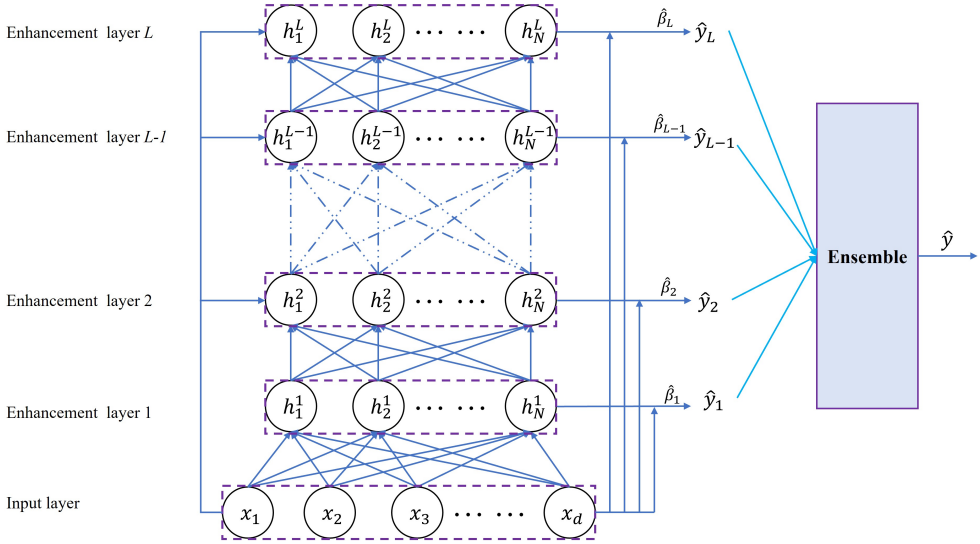
The stacked deep RVFL does not fully use the features from intermediate layers. However, the dRVFL's utilization of all features requires an inversion of a super large matrix



(a) sdRVFL [58]



(b) sdRVFL(dense) [58]



(c) edRVFL [57]

Figure 2.5: Different types of architectures of deep RVFL model.

[57]. Therefore, the edRVFL is proposed to achieve a trade-off between computational efficiency and features utilization [57, 253]. The structure of edRVFL is shown in Figure 2.5. In edRVFL, the direct link connects each enhancement layer with the input layer to guide the random features' generation. An individual output layer with a direct link to the input layer is built for each enhancement layer. Such design splits the inversion of a large matrix into multiple mini-matrix inversion and takes advantage of all features [57]. After training all the output layers, an ensemble block generates the final output. The majority voting mechanism is adopted for classification tasks [57, 254] and mean/median operation is applied for forecasting tasks [253]. Recently, a comparative study shows that edRVFL outperforms ensemble deep ELM on human joint angles prediction [264]. In addition to using all hidden layers for decision-making, Yu et al. [257] proposes to utilize a genetic algorithm for selection. Since the hidden neurons are randomized without optimization, there may be inferior neurons which hampers the generation of high-quality features in deeper layers. Therefore, Shi et al. [59] prune the inferior neurons before generating the next layer's neurons. In addition, a weighting scheme, which assigns different weights to the training samples in each hidden layer, is proposed to improve the performance. The wrongly classified samples are assigned larger weights in the next layer to increase the diversity and accuracy simultane-

ously. The norm features from an edRVFL are concatenated with the privileged information from another edRVFL with different activation [260]. This concatenation is fed into another deep RVFL for classifications. Recently, a strategy of random skip connections is proposed to enhance the representation ability of the edRVFL in [255]. Instead of using all output layers, an edRVFL with a selective ensemble method is designed and succeeds in landslide displacement prediction [257]. Following the principle of determining important output layers, an edRVFL with a dynamic ensemble based on online performance is proposed for time series forecasting [256]. An approach for the automatic design of ensemble deep randomized neural networks is proposed in [258].

In addition to the supervised learning based on edRVFL, Hu and Suganthan [265] propose a clustering algorithm based on edRVFL's features. The unsupervised learning is achieved by the manifold regularization. Then, the k means is developed based on the edRVFL features. The consensus clustering method is related to the ensemble block of the edRVFL. Recently, a novel edRVFL for semi-supervised tasks has been proposed by Shi et al. [259]. The proposed jointly optimized semi-supervised edRVFL (JOSedRVFL) minimizes the loss function consisting of three components, the error term, L_2 norm regularization, and manifold regularization. The L_2 norm regularization aims at reducing the model's complexity, and the manifold regularization ensures the conditional probabilities of similar samples are close. In [261], geometrical information under the graph embedded framework is employed while calculating the output parameters of each hidden layer (base model) and therefore, has better generalization performance than the edRVFl model.

2.3.7.3 Hybrid deep RVFL

Unlike the above deep RVFLs, the hybrid ones utilize other advanced feature extraction techniques, like DL, to generate the input of the decision block. The decision block can be any RVFL's variants, including shallow and deep architectures. For instance, features extracted from a pre-trained ResNet-50 are fed into RVFL whose activation is s-fuzzy membership function [61]. In [254], the ResNet extracts features from time series data, and these features are fed into multiple edRVFL. The convolutional sparse coding deep network extracts features and feeds them into a stacked deep RVFL [266]. In [247], the base model

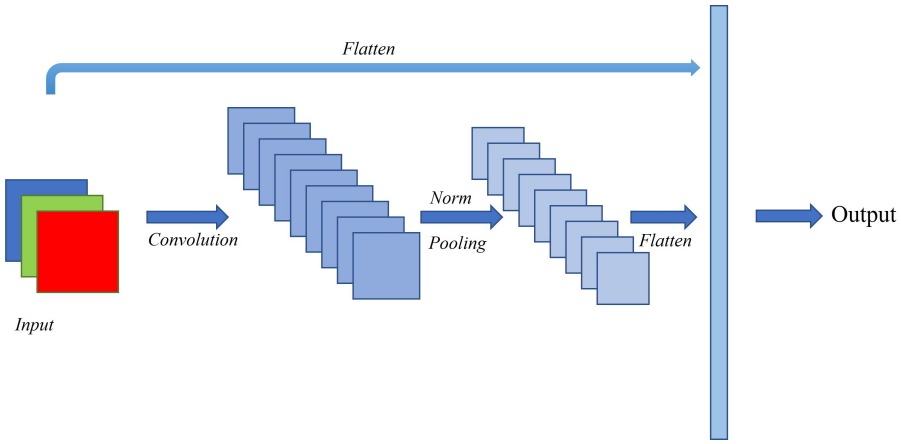


Figure 2.6: The architecture of deep convolutional RVFL.

in the ensemble learning framework is a hybrid deep RVFL whose DL performs feature extraction, and RVFL makes decisions.

2.3.8 Hyper-parameters optimization and experimental setup

The RVFL's performance heavily depends on hyper-parameters optimization. This section first separately summarizes the hyper-parameters optimization of single-layer and deep RVFLs. Finally, experimental setup, including data partitioning, evaluation metrics, and statistical tests, are presented.

2.3.8.1 Hyper-parameters optimization for single-layer RVFL

For the canonical RVFL, these hyper-parameters include input scaling, number of hidden nodes, activation functions, regularization strength, and distribution of random weights. Most literature utilizes a comprehensive grid search to tune these parameters. Grid search is straightforward to implement and succeeds on many tasks, such as classification [12, 190], forecasting [64, 253]. A comprehensive evaluation of RVFL for classification is conducted by Zhang and Suganthan [12]. A grid search based on 4-fold cross-validation is utilized to select the number of hidden nodes and regularization strength for the RVFLs with different configurations. This study achieves some significant findings. First, the results demonstrate the superiority of the direct links. Second, the output layer's bias must be tuned based on the specific task. Third, the *hadlim* and *sign* activation functions usually degenerate the ac-

curacy. Finally, tuning the scaling of randomization weights (input scaling) also increases the performance. Gao et al. [64] divides the time series into training, validation, and test set in chronological order. Then, a grid search is conducted to select the hyper-parameters, number of hidden nodes, and regularization strength, according to the forecasting errors on the validation set. Choosing the optimal activation function and number of hidden neurons is also an challenging job so some researchers adopt incremental learning techniques [166] and kernel trick [67] to avoid these issues. Table 2.1 shows the list of activation functions used in the literature.

However, exhaustive grid search has several drawbacks. Some literature implements evolutionary algorithms for hyper-parameters optimization, such as levy flight based PSO [194] and chimp optimization [193]. The evolutionary algorithms encode the hyper-parameters into an individual, obtaining the optimal configuration after many generations. Each generation selects the individuals whose performance is outstanding. Therefore, generation after generation, the suitable configuration survives.

Based on the above descriptions, there are two main branches of hyper-parameters tuning of single-layer RVFL, the grid search, and evolutionary optimizations. The performance of grid searches heavily relies on the researchers' experience because they have to define the hyper-parameters for selection manually. A detailed discussion and analysis of single-layer RVFL's hyper-parameters are given in [12]. For evolutionary optimization of the RVFL, the researchers can define a large search region to allow the evolutionary algorithm to explore the best configuration. However, the evolutionary algorithms are time-consuming because each generation trains multiple RVFLs.

The following suggestions about tuning single-layer RVFLs are provided for researchers and practitioners. First, large hidden dimensions are preferred for large datasets with huge input dimensions. Second, tuning the input scaling parameter may boost the performance on specific tasks, although the most common practice is to set the scaling factor to one. Third, it is advisable to increase the hidden nodes when tuning the input scaling worsens the performance. Fourth, the regularization strength of ridge regression plays a critical role in improving performance.

2.3.8.2 Hyper-parameters optimization for deep RVFL

As the RVFL becomes deep, the hyper-parameters that wait to be optimized grow exponentially. Whether each layer needs a different set of hyper-parameters is still an open problem. Katuwal and Suganthan [58] utilizes a grid search to optimize the number of hidden nodes and regularization, but the number of layers is fixed as three.

Some hyper-parameters tuning strategy is proposed for the deep RVFLs. For instance, a two-stage tuning strategy is proposed to obtain the best configurations of edRVFL for classification [57]. The optimal number of hidden nodes and regularization parameters are selected for a two-layer network in the first stage. Optimizing the hyper-parameters for a two-layer network saves computational time and also considers the effects of deep representations. Then, the second stage fine-tunes the hyper-parameters within the neighbourhood of the obtained number of hidden nodes and regularization parameter from the first stage optimization. Similar to the two-stage tuning, a three-stage method is proposed by Cheng et al. [254] for time series classification. In [254], the authors tunes the number of hidden nodes and regularization parameters for the first and second hidden layer in the first and second tuning stages. Then, the third stage imposes a random deviation on the optimal hyper-parameter obtained from the previous steps. The random variations can enhance the generalization ability and reduce the computational burden of the hyper-parameters tuning. Gao et al. [253] proposes a layer-wise grid search algorithm to determine the deep RVFL's hyper-parameters layer by layer. The cross-validation is conducted to obtain the best hyper-parameters. Once the hyper-parameters tuning is finished, the hyper-parameters and hidden states for this layer are fixed. The cross-validation is applied to the next layer. This process is repeatedly until the hyper-parameters for the last layer are determined.The layer-wise tuning offers each layer an opportunity to own different configurations. The layer-wise tuning method is also utilized in [83].

When the neural networks become deep, the hyper-parameters tuning becomes more challenging, because there are more hyper-parameters and the computation burden increases. For the deep RVFLs, the tuning of hyper-parameters must take into account both computational efficiency and performance. Hence, to achieve these two goals, stage-based

[57, 254] and layer-wise tuning algorithms are proposed [83, 253]. In summary, the stage-based tuning divides the tuning process into stages to save computation time and ensure efficient hyper-parameter exploration. The first stage of stage-based tuning searches hyper-parameters within a coarse region for saving computation time and exploring a large space. The following stages are the fine tuning to further improve the accuracy. The layer-wise tuning considers each hidden layer and the corresponding output layer as an independent model. Hence, each layer's tuning only works on a single-layer RVFL whose computation is much faster than that of deep architectures. Furthermore, such tuning ensures that each output layer performs exceptionally well. Finally, the layer-wise tuning can offer different layers with different hyper-parameters to increase the diversity.

The preceding discussion offers some insights into tuning hyper-parameters for the deep RVFLs. First, tuning for shallow RVFLs or each layer can save computational time. Second, searching within a coarse region is an efficient way to explore distinctive hyper-parameters. Third, fine-tuning based on the selected hyper-parameters from wide ranges can further improves the performance. Fourth, assigning different hyper-parameters to each layer contributes to the diversification of the edRVFL.

2.3.8.3 Experimental setup

This section presents and summarizes the experimental procedures, including data partitioning, normalization, evaluation metrics, and statistical tests. For classification problems, the researchers usually adopt the k-fold cross-validation for hyper-parameters tuning and evaluate the models on the remaining test set [12]. If the partition schemes of the dataset are available, the researchers must follow the same partitioning for fair comparisons, such as annealing and audiology-std dataset [12]. With a given experimental setup, there is a need to benchmark the models for fair comparison. There have been multiple attempts in the literature to benchmark the performance of the models on a given experimental setup Fernández-Delgado et al. [267]. Recently, self normalizing networks [268] released the publicly available data partitions for reproducibility and the benchmarking of the models. Following self normalizing networks [self normalizing ref], several studies like [57] followed this setup for fair evaluation of the models. Recently, Del Ser et al. [102] presented

a through survey based on randomization based machine learning models with renewable energy prediction problems and compared them. There is still a gap for the benchmarking of the models like lack of evaluation of the models across different hyper-parameters, their range and so on. Thus, benchmarking of the models to ensure the progress of the literature, reproduciblity of the results and fair comparison is needed in randomization based models.

The test set is always located at the end for time series datasets, and the remaining observations are utilized for training and hyper-parameters tuning. There are two approaches to split the training and validation set. The first approach is the same as the cross-validation for regression and forecasting. Some researchers formulate the observations into input patterns and response values. Then, a k-fold cross-validation is conducted to tune the hyper-parameters [229]. The second approach splits the training and validation set in chronological order [64].

All models have limitations, and therefore evaluation and comparison of machine learning models depend on the specific dataset. A fruitful set of evaluation metrics is utilized to evaluate the RVFL's performance. For classification, the classification accuracy is always the first choice [57]. Table 2.6 summarizes the forecasting errors utilized in the literature, where, x_j and $\hat{x}_j$ represent the raw observation and its forecast, L and T represent the size of training and test set, respectively. MAE and RMSE can be utilized when the time series are of the same scale, although RMSE is more sensitive to outliers. MAPE is a popular percentage error with high interpretibility. Finally, MASE is a scaled metric and can be utilized for comparisons on different time series. In addition to the forecasting errors, the direction statistics are utilized for comparison in some literature [230, 231]. The definition of direction statistics (*Dstat*) is

$$Dstat = \frac{1}{L}\sum_{1}^{L} a_i \times 100\%, \tag{2.14}$$

where $a_i = 1$ if $(\hat{x}_j - x_{j-1})(x_j - x_{j-1}) > 0$, or otherwise $a_i = 0$.

In addition to the above evaluation metrics, the literature also utilizes statistical tests to compare the different models' performance on various datasets. In general, these tests can be classified into two groups, the group-wise and pair-wise tests. Group-wise tests can

Table 2.6: Forecasting errors in the literature about RVFL.

Metric	Formula
Mean absolute error (MAE)	$\frac{1}{L}\sum_1^L \lvert\hat{x}_j - x_j\rvert$
Mean absolute scaled error (MASE)	$\frac{1}{L}\sum_1^L \frac{\lvert\hat{x}_j - x_j\rvert}{\frac{1}{T-1}\sum_{t=2}^{T}\lvert x_t - x_{t-1}\rvert}$
Mean squared error (MSE)	$\frac{1}{L}\sum_1^L (\hat{x}_j - x_j)^2$
Root mean squared error (RMSE)	$\sqrt{\frac{1}{L}\sum_1^L (\hat{x}_j - x_j)^2}$
Mean absolute percentage error (MAPE)	$\frac{1}{L}\sum_1^L \left\lvert\frac{\hat{x}_j - x_j}{x_j}\right\rvert$.

determine the overall ranking of the models on all the datasets and group them based on the statistical distance. The literature about RVFL utilizes the Nemenyi test to compare the models in a group-wise fashion [64, 229]. The pair-wise tests assist in comparing the models in a pair-wise fashion, which is straightforward to show the better model. For instance, the Wilcoxon test is utilized to ascertain how many algorithms edRVFL significantly outperforms [57]. There are several others statistical tests such as Friedman test, sign-test and so on, to evaluate the performance of machine learning models. We refer the reader to [267, 269, 270] for more detailed information about the application of these tests to machine learning models.

2.3.9 Time series forecasting and other applications

Time series forecasting refers to establishing the model using historical observations, and this model is utilized to make extrapolations for future steps. Accurate and reliable forecasts help the stakeholders and decision-makers plan, organize, maintain and develop the system in advance in a data-driven fashion. Table 2.7 summarizes the representative literature about forecasting by RVFL and its variants. RVFL and the improved versions have demonstrated their outstanding performance on various forecasting tasks from different domains, such as electricity load [64, 229], solar power [271], wind power [234], financial time series [219] and other data [195]. Table 2.7 shows that signal decomposition algorithms are popular for feature extraction on forecasting tasks. The signal decomposition can splits the time series into multiple sub-series with different frequencies. Then the RVFL works on these sub-series for forecasting. This section presents the details of all the literature about RVFL-based forecasting.

Table 2.7: Summary of RVFLs for forecasting.

Year	Literature	Feature extraction	Learning category	Hyper-parameter optimization	Field
2022	Gao et al. [83]	-	edRVFL	Layer-wise grid search	Inpatient discharges
2021	Zayed et al. [193]	-	Kernelized RVFL	Chimp Optimization Algorithm	Solar power
2021	Dash et al. [194]	Signal decomposition	Expanded RVFL	Particle swarm optimization	Solar power
2021	Majumder et al. [272]	-	Online sequential kernel RVFL	-	Solar power
2021	Gao et al. [64]	Signal decomposition	Single model	Grid search	Electricity load
2021	Hu et al. [70]	-	Ensemble RVFL	Evolutionary optimization	Electricity load
2021	Manibardo et al. [273]	Deep architecture	RVFL, deep RVFL and edRVFL	Bayesian optimization	Road traffic
2020	Cheng and Wang [274]	Signal decomposition	Decomposition-based ensemble learning	-	Wind speed
2018/2020	Zhang et al. [125], Qiu et al. [219], Tang et al. [231]	Signal decomposition	Decomposition-based ensemble learning	Grid search	Crude oil price
2020	Wu et al. [236]	Signal decomposition	Decomposition-based ensemble learning	Sine cosine algorithm	Crude oil price
2020	Wu et al. [235]	Signal decomposition	Decomposition-based ensemble learning	whale optimization algorithm	Financial time series
2020	Zhang et al. [275]	Stacked auto-encoder	Incremental RVFL	Grid search	FCCU end-point quality
2020	Hazarika and Gupta [195]	Signal decomposition	Single RVFL	Grid search	COVID-19 cases
2019	Bisoi et al. [119]	Signal decomposition	Single RVFL	Grid search	Crude oil price
2019	Kushwaha and Pindoriya [276]	Signal decomposition	Decomposition-based ensemble learning	Grid search	Solar power
2019	Majumder et al. [277]	-	Kernelized RVFL	Water cycle algorithm	Solar power
2018	Moudiki et al. [278]	-	RVFL with different regularizations	Grid search	Financial time series
2018	Lian et al. [279]	-	Ensemble RVFL	Grid search	Landslide displacement
2018	Li et al. [232]	Signal decomposition	Decomposition-based ensemble learning	Grid search	Travel time
2018	Lu et al. [239]	-	Ensemble RVFL	Grid search	Production rate
2017/2018	Nhabangue and Pillai [202], Qiu et al. [234]	Signal decomposition	Decomposition-based ensemble learning	Grid search	Wind power
2016/2018	Qiu et al. [220, 229]	Signal decomposition	Decomposition-based ensemble learning	Grid search	Electricity load
2016	Zhang et al. [204]	-	Incremental RVFL	Grid search	Molten iron quality
2015	Zhou et al. [205]	PCA	Online sequential RVFL	Grid search	Molten iron quality

2.3.9.1 Electricity load

Electricity load forecasting is crucial for the development, maintenance, and planning of power systems. Among the abundant forecasting methods, RVFL demonstrates its success by many researchers. For instance, a quantile scaling method is proposed to re-distribute the randomly weighted inputs of RVFL to avoid the saturation effects and suppress the outliers in [50]. Signal decomposition techniques are utilized to remedy the unsupervised features of RVFL in [64, 220, 229]. For instance, EMD decomposes the load data into several modes, and then each mode is predicted by an RVFL. Finally, the summation is conducted to combine the predictions for each mode [229]. In [220], a two-stage decomposition method is proposed to decompose the load data, then each load is predicted by an individual RVFL, and finally, all forecasts are aggregated by another RVFL with explanatory variables. Different from the above ensemble methods, a single RVFL is built on all the components generated by EWT in [64]. The same decomposition scheme is combined with edRVFL for short-term load forecasting in [253] and the results demonstrate edRVFL's superiority over a single RVFL. Moreover, a multi-modal evolutionary algorithm is utilized to optimize the enhancement weights, bias, and combination weights of the ensemble RVFL for short-term load forecasting [70].

2.3.9.2 Solar power

With renewable energy development, solar power forecasting is an emerging area. In [271], the authors compare RVFL with SLFN and RWSLFN, and the results demonstrate the superiority of the direct link. Signal decomposition methods also work for solar time series. For instance, in [276], maximum overlap DWT decomposes the power data into sub-series, and an individual RVFL predicts each series. Finally, the aggregation of all forecasts is the forecast for solar power. Moreover, some researchers utilize meta-heuristics algorithms to optimize RVFL's parameters automatically. For instance, in [277], the multi-kernel RVFL whose kernel parameters are optimized by the water cycle algorithm is proposed to forecast short-term solar power. In [193], the Chimp Optimization Algorithm (CHOA) is utilized to determine RVFL's hyper-parameters for predicting output power and the monthly power production of a solar dish/Stirling power plant. Some researchers integrate signal decomposition, evolutionary optimization, and RVFL together to boost forecasting accuracy. For example, in [194], the EWT is utilized to decompose the time series, and the residue is discarded. The remaining sub-series are expanded using trigonometric activation in the direct link, and the enhancement states are a linear combination of two activation functions. Finally, the RVFL is trained with a novel robust objective function by minimizing the variance of training data. Moreover, the added activation functions' hyper-parameters are also optimized by PSO. The new cost function also shows its improvement on RVFL in terms of forecasting accuracy. In [272], an Online Sequential Kernel-based Robust RVFL is trained based on Hampel's cost function to forecast solar and wind power.

2.3.9.3 Wind power

A comparison of RVFL and other machine learning models on wind speed forecasting is conducted in [280]. Some literature about wind power forecasting combines signal decomposition techniques with RVFL [202, 234]. For instance, CEEMD is applied to decompose the raw data into modes, and a kernel ridge regression predicts each mode. Finally, instead of using a simple summation, the RVFL is trained to combine the forecasts of all methods for wind power ramp prediction in [234]. In [202], Chebyshev expansion is utilized as func-

tional nodes to reduce the number of activation nodes. Then it is combined with EMD for wind speed forecasting. In addition, Hampel's cost function is utilized for training an online sequential kernel-based robust RVFL to forecast solar, and wind power in [272]. In Cheng and Wang [274], a multi-objective salp swarm optimizer is adopted to determine the weights that are used to combine the forecasts from four networks, including RVFL, for wind speed forecasting.

2.3.9.4 Financial time series

The financial time series is different from the above data with strong cycles. The financial time series is very volatile, and it is difficult to extract features. Among all the RVFL-related financial time series forecasting literature, most focus on crude oil prices. Similar to the other kinds of time series forecasting literature, many researchers combine RVFL and signal decomposition algorithms for crude oil price forecasting [125, 219, 231]. EEMD decomposes crude oil price, and then different RVFLs are trained for each mode, including the residue. Finally, the summation of all RVFL's outputs is the forecast in [231]. In [219], CEEMD decomposes the raw data into modes, and an individual RVFL is established on each mode. Finally, forecasts of all modes are aggregated using an incremental RVFL. The same decomposition-based structure is utilized. The difference is that improved CEEMD with adaptive noise acts as the decomposition in [235, 236] and the sine cosine algorithm optimizes all the parameters in[236]. In [125], bivariate EMD is utilized to decompose the original time series into sub-series, and an individual RVFL predicts each series. Finally, aggregate the forecasts via summation. The modes generated from VMD are fed into RVFL, and the experimental results demonstrate the superiority of VMD over EMD in [119]. Besides the decomposition-based RVFL, a novel ensemble RVFL with five diversity strategies is proposed for crude oil price forecasting in [71].

Besides the literature about crude oil price forecasting, RVFL also succeeds on other financial time series. For example, in [278], different regularization parameters are imposed to the output weights of the direct link and enhancement nodes to forecast discount rates. In [235], the improved CEEMD with adaptive noise decomposes the data into sub-series, and each sub-series is predicted by an individual RVFL whose parameters are optimized by

a whale optimization algorithm. Finally, the output is the summation of all forecasts. In [281], multilingual search engine data is utilized to derive the input for RVFL to forecast crude oil prices.

2.3.9.5 Other applications

Besides the above popular areas with a large need for forecasting, RVFL and its improved versions have also succeeded in various areas, such as temperature [217], landslide displacement [279], COVID-19 cases [195], travel time [232], molten iron quality [204, 205], energy consumption [233], signal-to-noise ratio [282], algae missing values [199], temperature in subway station [283], inpatient discharges [83] and so on [273].

Among this literature, many utilize different heuristic algorithms to optimize the hyper-parameters [169, 196, 197, 198], weights [284] or select the input features [199]. For instance, firefly algorithm is utilized to select RVFL's hyper-parameters, number of enhancement nodes, bias, direct link, distribution and activation function, for thermal performance prediction [195]. In [285], an incremental method to adjust RVFL's structure is proposed for time series prediction, where the network increases its enhancement nodes when the performance degrades. For instance, in [199], a moth search algorithm is utilized to select input features for RVFL to predict missing values of algae. In [196], Marine Predators Algorithm is utilized to optimize RVFL's hyper-parameters for tensile behavior prediction. In [197], an artificial ecosystem-based optimization algorithm is utilized to optimize RVFL's hyper-parameters for forecasting power consumption and water productivity of seawater. In [169], mayfly-based optimization is utilized to optimize RVFL's hyper-parameters to forecast the performance of Photovoltaic/Thermal Collector. In [284], the RVFL trained by PSOGA is utilized to generate prediction intervals for landslide displacement. The RVFL is first pre-trained using reconstructed intervals, and then the PSOGA trains the RVFL with transferred weights based on original data.

Besides the RVFL-based on meta-heuristics, some literature also focuses on ensemble RVFL [232, 233, 239, 279]. For instance, GA is utilized to assign ensemble weights for each RVFL trained with bootstrap samples, and the RVFLs whose weights are higher than the threshold are selected to construct prediction intervals for landslide displacement [279].

In [232], EEMD is utilized to decompose the travel time into modes, and a different RVFL predicts each mode. Finally, each mode's forecasts are combined with linear addition. In [217], ensemble RVFL is trained based on AdaBoost after selection features via MRMR. Each RVFL is trained using iteratively reweighted least squares for temperature forecasting. In [233], EEMD decomposes time series into sub-series, and the features with high correlation with target variables are used for the corresponding RVFL to forecast building energy consumption. In [239], Negative Correlation Learning is utilized for training ensemble RVFL networks for production rate forecasting. Manibardo et al. [273] applies RVFL, deep RVFL, and edRVFL to the road traffic dataset, and the hyper-parameters are determined by Bayesian optimization. Manibardo et al. [273] claims that the direct link is the fundamental reason for RVFL and its variants' superiority over ELM-based models.

Incremental (online) RVFL also succeeds on various time series [201, 204, 205, 275, 286]. Incremental RVFL updates its structure or weights when new observations are available. For example, in [204], the incremental RVFL adds a new node and updates its weight incrementally until the performance degrades for the prediction of molten iron quality. In [205], the online sequential RVFL is trained using the principal components and its estimation of the previous steps to predict molten iron quality, too. In [201], RVFL and online sequential RVFL are compared on rainfall prediction, and the results demonstrate OS-RVFL's superiority for rainfall forecasting. In [286], an online RVFL-based on sliding window is trained to temperature forecasting. In [275], a stacked auto-encoder is trained in an offline fashion first, and then an incremental RVFL is established based on the SAE's output when a concept drift is detected.

In addition, different novel RVFLs are proposed for other time series. In [195], the level one sub-series generated from discrete wavelet transformation are fed into RVFL for COVID-19 cases forecasting. In [287], Schmidt orthogonalization is utilized to orthogonalize the output vectors, and the hidden nodes are pruned according to the output weights to predict product quality.

Although RVFL and its variants succeed in various time series forecasting tasks, the research on spatial-temporal time series is not mature. There is only one RVFL-related paper touching on this problem to the author's best knowledge. In [203], a kernel RVFL is

established to predict the temporal dynamics decomposed via the Karhunen–Loève method from the spatial-temporal process.

2.3.10 Comparison with other state-of-the-art machine learning techniques

An insightful discussion about the comparison between the RVFLs and other state-of-the-art machine learning techniques significantly contributes to the literature. This section mainly discusses the pros and cons of the RVFLs and other machine learning models.

The early work with randomization techniques can be traced with perceptron and standard feed-forward neural network (SLFN). In perceptron, the weights between sensor units and response units can be generated randomly whereas the rest weights from the associator units and the response units are calculated via reinforcement learning [288, 289]. SLFN [113] also uses randomization technique but there are no direct links in this network. Jacobian neural Network (JNN) [290] is a polynomial time randomized algorithm which gives optimal network with probability one. Moreover, the paper [117] has some theoretical justification for RVFL and other neural networks. Backpropagation-based trained ANNs are sensitive to learning rate setting, slow convergence, and trapped into local minima [96, 97, 98]. On the other hand, RVFL resolves these issues by generating the weights randomly from the input layer to the hidden layer and the rest weights (hidden layer to output layer) are calculated via closed form solution. The direct links in RVFL play an important role in both classification and regression problems [12, 13]. These direct connections separate RVFL from other randomized networks such as radial basis function (RBF) [15] and extreme learning machine (ELM) [16] and so on. RVFL and its deep variants have shown superior performance than ELM, Hierarchical ELM (H-ELM), and multi-layer kernel ELM (ML-KELM) [58]. Support vector machine (SVM) has strong mathematical foundation and has shown state-of-the-art results [17, 30, 291]. However, RVFL with privileged information (RVFL+) and its kernel extension (KRVFL+) have shown superior performance than SVM and its variants such as gSMO-SVM+ and fast SVM+ [160]. From an optimization perspective, RVFL+ has simpler constraints than SVM+, which results in closed form solu-

tion. Furthermore, the DRVFL shows superiority over SVM and random forest on Twitter sentiment datasets. The DRVFL with fuzzy activation outperforms EML and kernel-ELM on the ADNI dataset.

One popular state-of-the-art deep learning method is the Resnet. The Resnet constructs a quite deep architecture with the help of residual links. It utilizes a backpropagation algorithm to optimize the weights and bias, which takes much more time than training the RVFL networks. In addition, the literature has demonstrated the superiority of deep RVFL over the Resnet on tabular data classification [57, 59]. Besides tabular datasets, time series forecasting is also a valuable problem. For forecasting, the long short-term memory (LSTM) and temporal convolutional network (TCN) are two common state-of-the-art methods. Compared with edRVFLs, the training is much slower. However, many literature shows that the advanced RVFLs outperform the LSTM and TCN [253]. Furthermore, the deep stacked RVFL method outperforms stacked denoising auto-encoder on two benchmark MR brain datasets (MD-1 and MD-2). Some literature has demonstrated that RVFL-based models outperform the BPNN [119].

However, RVFL does not include CNN-type feature extraction layers for image or sequence data. The convolution filters aim at mining local patterns from different spaces. Then, multiple stacks of these filters assist in learning a global representation. Finally, the gradient descent algorithms help to learn these features in an end-to-end fashion. Although RVFL does not own the CNN's feature extraction layers, the features learned by CNNs can be utilized as input to the RVFL variants. In other words, the RVFL variants can be the decision module for the features from all kinds of gradient-based deep networks [254].

2.3.11 Summary

Randomized neural networks (RdNNs) have shown their strength among machine learning models. A special kind of RdNN, the RVFL model, has emerged as a very successful model. This review summarizes the developments of RVFLs from theoretical foundations to various applications. The RVFL is a feed-forward neural network with random features and direct links. The randomized features introduce non-linear representations of the in-

put features, and the direct links reserve the linear pattern. The hidden layer's weights are randomly initialized and frozen during training, and only the output layer is trained with a closed-form approach. The randomized features render RVFLs at a fast computational speed. With the renaissance of deep learning, researchers extended the shallow RVFL to deep architectures to enhance its representation ability. In the deep architectures of RVFL, the hidden neurons are randomly initialized and frozen during the training step. Only the output layers are trained, which reduces the computational burden of backpropagation. The RVFLs with deep architectures have demonstrated their superiority over shallow ones on classification, regression, and forecasting. Therefore, the literature shows that the hierarchical enhancement features offer a large modeling capacity and increase performance. Different ensemble learning algorithms, such as bagging, boosting, and stacking, are shown to significantly boost the single RVFL's performance. In addition, the ensemble RVFLs based on signal decomposition demonstrate tremendous success on various forecasting tasks. The signal decomposition algorithms first disaggregate the complex sequential data into multiple components, which assists in the RVFL's representation ability. Then RVFL-based models are built on each component, and the ensemble of all forecasts is the output. The RVFL-based models have achieved significant success in various domains because of their fast computational speed, high accuracy, and powerful representation ability and these models also have achieved state-of-the-art performance in the time series forecasting domain on wind speed, solar energy, electricity load, etc. we hope that this review offers treasure information about the RVFL model to the researchers. We presented a thorough survey on the developments of the RVFL model in many aspects such as shallow RVFLs, ensemble algorithms based on the RVFL models, deep RVFL variants, etc. Also, we discussed the applications of RVFL models, that show their applicability in the real world. The literature has demonstrated the superiority of the RVFL-based deep models over tabular datasets [59].

2.4 Extreme learning machine

Unlike standard RVFL, the ELM [16] doesn't have direct links from input layer to the output layer. Therefore, the objective function of ELM can be formulated as:

$$\min_{\beta \in \mathbb{R}^{L \times m}} \frac{1}{2} \|\beta\|_2^2 + \frac{1}{2} C \|H\beta - Y\|_2^2 . \tag{2.15}$$

The solution of the objective function (2.15) can be calculated as:

$$\beta = \begin{cases} (H^T H + \frac{1}{C} I)^{-1} H^T Y, & L \leq N, \\ H^T (H H^T + \frac{1}{C} I)^{-1} Y, & N < L. \end{cases} \tag{2.16}$$

2.5 Hyperplane based learning

SVM [17] is an effective large margin classifier that was developed using statistical learning theory [18]. Structural risk minimization (SRM) principle is at the core of statistical learning. SVM and its variants have been successfully employed in several domains such as Alzheimer's disease prediction [19], image recognition [20], bioinformatics [21] and so on. From SVM, numerous new algorithms are derived such as least-squares SVM (LSSVM) [176], Pin-SVM [33], fuzzy SVM (FSVM) [292] and so on [293, 294]. However, SVM has a considerable degree of computational complexity, i.e., $O(N^3)$, where N represents the number of dataset. Therefore, it makes sense to improve its effectiveness and simplify its computation. As a result, Jayadeva et al. [22] developed twin SVM (TWSVM), a framework designed to reduce the time required for SVM. The concept underlying TWSVM was derived from proximal SVM via generalized eigenvalues (GEPSVM) [295] in order to obtain two optimal nonparallel proximal hyperplanes. Each hyperplane is as far from the other class and as close as it can be to one of the two classes. TWSVM is comparable to SVM in terms of the classification model, however, it comes with a number of extra advantages over SVM. First, because TWSVM model solves two smaller QPPs rather than one larger one (as in SVM), computing complexity is minimized. Additionally, TWSVM performs well on cross-plane data sets due to its nonparallel nature. Over the last decades,

numerous enhanced TWSVM-based algorithms have been proposed such as least square twin SVM (LSTSVM) [23], twin bound SVM (TBSVM) [24], energy-based least squares twin support vector machine (ELSTSVM) [25], intuitionistic fuzzy weighted least squares TWSVMs [296], and so on [28, 29]. The robust energy based least squares twin support vector machines (RELSTSVM) [26] added an additional regularisation component to the ELSTSVM formulation, making the optimization problems positive definite and improving generalization performance. According to a recent study [30], RELSTSVM is the most efficient classifier among twin support vector machine-based models.

Although TWSVM has been very successful at classifying data, it has a number of flaws, including sparsity, the challenge of inverting matrices for large data sets, sensitivity to feature noise near decision boundaries [33], and the requirement to reconstruct optimization problems in nonlinear cases by taking kernel-generated surfaces into account [34]. In order to improve TWSVM for classification tasks, Tian et al. [35] developed nonparallel support vector machine (NPSVM), which employs a ϵ-insensitive and hinge loss functions, avoids matrix inversion operations (as in TWSVMs), and kernel trick is used directly. The following are some benefits of NPSVM: it constructs the same two convex QPPs for both linear and nonlinear cases; it no longer requires the computation of the inverse matrices prior to training and can be efficiently solved by the sequential minimimal optimization (SMO) algorithm; it has similar sparseness to standard SVMs; and when the parameters are properly chosen, NPSVM degenerates to the TWSVMs. Incorporating structural information with NPSVM, Chen et al. [297] suggested the structural NPSVM (SNPSVM) model. By fully utilizing prior knowledge to boost the generalization capabilities of the algorithm, SNPSVM accounts for both the compactness in both classes via structural information as well as the separability between classes. Several variants of NPSVM model have been developed such as L2-NPBSVM [298], L1-NPSVM [299], Pin-NPSVM [300] and so on.

Here, we discuss the SVM and twin SVM variants. Let $X_1 \in \mathbb{R}^{m_1 \times d}$ matrix contains the samples of class (+1) and $X_2 \in \mathbb{R}^{N_2 \times d}$ matrix contains the samples of class (-1) with $N = m_1 + m_2$. Where, m_1 and m_2 represent the number of samples in class (+1) and class (-1), respectively, and d is the number of features. Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be the kernel function [301] such that $\forall x,\ y, \in X,\ K(x, y) = \langle \phi(x), \phi(y) \rangle$. Where ϕ is a mapping from

X to high dimensional feature space and $\langle , \rangle$ represents the inner product.

2.5.1 Support vector machine

The support vector machine (SVM) [17, 302] aims to find a separating hyperplane, $f(x) = w^T x + b$, where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$. The optimization problem of the standard SVM is as follows:

$$\begin{aligned} \min_{w,b,\xi} \ & \frac{1}{2}||w||^2 + c\sum_{i=1}^{N}\xi_i \\ \text{subject to} \ & y_i(w \cdot x_i + b \geq 1 - \xi_i, \ i = 1, 2, ..., N. \\ & \xi_i \geq 0, \quad i = 1, 2, ..., N. \end{aligned} \tag{2.17}$$

Where $c > 0$ determines the trade-off between the maximum margin and the minimal classification error and ξ_i is the error value. The dual problem corresponding to primal problem 2.17 is as follows:

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i \\ \text{subject to} \ & \sum_{i=1}^{N} y_i\alpha_i = 0, \\ & 0 \leq \alpha_i \leq c, \quad i = 1, 2, ..., N. \end{aligned} \tag{2.18}$$

Where $K(x_i, x_j)$ is the kernel function and α_i is the Lagrangian multiplier. The decision function of standard SVM is as follows:

$$f(x) = \text{sgn}\Big(\sum_{i=1}^{N} y_i\alpha_i K(x_i, x) + b\Big). \tag{2.19}$$

2.5.2 Twin support vector machine

TWSVM [22] finds two nonparallel hyperplanes such that the hyperplanes are proximal to the corresponding classes and at least at a unit distance from the samples of other class.

The hyperplanes are defined as:

$$f_1(x) = w_1^T x + b_1 = 0 \text{ and } f_2(x) = w_2^T x + b_2 = 0, \tag{2.20}$$

where w_1 and w_2 are vectors in $\mathbb{R}^d$ and b_1, b_2 are real numbers. The optimization problem of TWSVM for nonlinear case are as follows:

$$\begin{aligned} &\min_{w_1,b_1,\tau_2} \frac{1}{2} \parallel K(X_1, C^T)w_1 + e_+ b_1 \parallel^2 + C_1 e_-^T \tau_2 \\ &\text{subject to } -(K(X_2, C^T)w_1 + e_- b_1) + \tau_2 \geq e_-,\ \tau_2 \geq 0 \end{aligned} \tag{2.21}$$

and

$$\begin{aligned} &\min_{w_2,a_2,\tau_1} \frac{1}{2} \parallel K(X_2, C^T)w_2 + e_- b_2 \parallel^2 + C_2 e_+^T \tau_1 \\ &\text{subject to } (K(X_1, C^T)w_2 + e_+ b_2) + \tau_1 \geq e_+,\ \tau_1 \geq 0, \end{aligned} \tag{2.22}$$

where $C = [X_1; X_2]$ and $C_j > 0$ for all j, is the adjustable parameter. Moreover, e_+ and e_- represent the vectors having one of appropriate dimension, and $K(.)$ represents the kernel function. τ_1 and τ_2 represent the error corresponding to positive and negative class, respectively.

The Wolfe dual corresponding to optimization problems (2.21) and (2.22) are:

$$\begin{aligned} &\max_{\rho} \ e_-^T \rho - \frac{1}{2} \rho^T Q (P^T P)^{-1} Q^T \rho \\ &\text{subject to } 0 \leq \rho \leq C_1 e_-, \end{aligned} \tag{2.23}$$

and

$$\begin{aligned} &\max_{\sigma} \ e_+^T \sigma - \frac{1}{2} \sigma^T P (Q^T Q)^{-1} P^T \sigma \\ &\text{subject to } 0 \leq \sigma \leq C_2 e_+, \end{aligned} \tag{2.24}$$

where $P = [K(X_1, C^T),\ e_+]$ and $Q = [K(X_2, C^T),\ e_-]$. Now, one can calculate the

final vectors as: $u_1 = -(P^TP + C_1I)^{-1}Q^T\rho$, and $u_2 = (Q^TQ + C_2I)^{-1}P^T\sigma$, where $u_1 = [w_1^T\ b_1]^T$ and $u_2 = [w_2^T\ b_2]^T$. The final decision is taken as given in [22].

2.5.3 Least squares twin support vector machine

The objective function of the linear LSTSVM [23] can be formulated as:

$$\begin{aligned} \min_{w_1,b_1}\ & \frac{1}{2}\|X_1w_1 + e_+b_1\|^2 + \frac{C_1}{2}\|\tau_2\|^2 \\ \text{subject to}\ & -(X_2w_1 + e_-b_1) + \tau_2 = e_-, \end{aligned} \tag{2.25}$$

and

$$\begin{aligned} \min_{w_2,b_2}\ & \frac{1}{2}\|X_2w_2 + e_-b_2\|^2 + \frac{C_2}{2}\|\tau_1\|^2 \\ \text{subject to}\ & (X_1w_2 + e_+b_2) + \tau_1 = e_+, \end{aligned} \tag{2.26}$$

where C_1 and C_2 are two positive parameters and τ_1 and τ_2 are slack variables. Unlike TWSVM, in linear LSTSVM only system of linear equations needs to be solved. Let $P_1 = [X_1\ \ e_+]$ and $P_2 = [X_2\ \ e_+]$. After some calculation [23], the solution of (2.25) and (2.26) are calculated as follow:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(C_1P_2^TP_2 + P_1^TP_1\right)^{-1} C_1\ P_2^T\ e_-, \tag{2.27}$$

and

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(C_2P_1^TP_1 + P_2^TP_2\right)^{-1} C_2\ P_1^T\ e_+. \tag{2.28}$$

Due to matrix singularity issue, uses $\left(C_1P_2^TP_2 + P_1^TP_1 + \epsilon^*I\right)^{-1}$ and $(C_2P_1^tP_1 + P_2^tP_2 + \epsilon^*I)^{-1}$ in (2.27) and (2.28), respectively. Here, I is an identity matrix of appropriate dimension and ϵ^* is a small positive scalar. The new sample $x \in \mathbb{R}^d$

is assigned a class +1 or -1 depending on which of the distance $|w_1^T x + b_1|$ or $|w_2^T x + b_2|$ is minimum. Here, $|.|$ denotes the distance of a point with the hyperplane.

2.5.4 Robust energy-based least squares twin support vector machine

The optimization problem of linear robust energy-based least squares twin support vector machines [303] is as follows:

$$\begin{aligned} \underset{w_1,b_1}{min} \ & \frac{1}{2}\|X_1 w_1 + e_+ b_1\|^2 + \frac{C_1}{2}\|\tau_2\|^2 + \frac{C_3}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 \\ \text{subject to } & -(X_2 w_1 + e_- b_1) + \tau_2 = E_1, \end{aligned} \tag{2.29}$$

and

$$\begin{aligned} \min_{w_2,b_2} \ & \frac{1}{2}\|X_2 w_2 + e_- b_2\|^2 + \frac{C_2}{2}\|\tau_1\|^2 + \frac{C_4}{2}\left\|\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}\right\|^2 \\ \text{subject to } & (X_1 w_2 + e_+ b_2) + \tau_1 = E_2. \end{aligned} \tag{2.30}$$

Now substitute the equality constraints into the objective function then the QPP (2.29) becomes:

$$L_1 = \frac{1}{2}\|X_1 w_1 + e_+ b_1\|^2 + \frac{C_1}{2}\|X_2 w_1 + e_- b_1 + E_1\|^2 + \frac{C_3}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2. \tag{2.31}$$

Taking the partial derivatives of L_1 with respect to w_1 and b_1 and putting it equal to zero, the solution of (2.29) is given as:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(C_1 P_2^T P_2 + P_1^T P_1 + C_3 I\right)^{-1} C_1 \, P_2^T \, E_1, \tag{2.32}$$

and similarly we can do the same calculations for solving the QPP (2.30) and gets the solution as:

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(C_2 P_1^T P_1 + P_2^T P_2 + C_4 I\right)^{-1} C_2 \; P_1^T \; E_2, \tag{2.33}$$

where P_1, P_2 have same meaning as defined [303]. For a new sample x, the final decision is taken as given in [303].

2.5.5 Nonparallel support vector machine

A generalized version of the TWSVM was developed in the recently proposed nonparallel support vector machine (NPSVM) [35]. When compared to standard SVM and TWSVM, the NPSVM model has better generalization performance. The optimization problems of the NPSVM model are defined as follows:

$$\begin{aligned} \min_{w_1, b_1, \eta_+, \eta_+^*, \xi_-} \quad & \frac{1}{2} \parallel w_1 \parallel^2 + c_1 e_+^T (\eta_+ + \eta_+^*) + c_3 e_-^T \xi_- \\ \text{subject to} \quad & X_1 w_1 + e_+ b_1 \leq \epsilon e_+ + \eta_+, \\ & - X_1 w_1 - e_+ b_1 \leq \epsilon e_+ + \eta_+^*, \\ & - (X_2 w_1 + e_- b_1) \geq e_- - \xi_-, \\ & \eta_+, \eta_+^*, \xi_- \geq 0. \end{aligned} \tag{2.34}$$

and

$$\begin{aligned} \min_{w_2,b_2,\eta_-,\eta_-^*,\xi_+} \quad & \frac{1}{2} \parallel w_2 \parallel^2 + c_2 e_-^T(\eta_- + \eta_-^*) + c_4 e_+^T \xi_+ \\ \text{subject to} \quad & X_2 w_2 + e_- b_2 \leq \epsilon e_- + \eta_-, \\ & - X_2 w_2 - e_- b_2 \leq \epsilon e_- + \eta_-^*, \\ & (X_2 w_2 + e_- b_2) \geq e_- - \xi_+, \\ & \eta_-, \eta_-^*, \xi_+ \geq 0. \end{aligned} \tag{2.35}$$

where $\epsilon > 0$, $c_i > 0$, for $i = 1, 2, 3, 4$. η_+ and η_+^* are the error corresponding to class (+1) and ξ_+ represent the error of class (-1). The similar meaning has η_-, η_-^* and ξ_+. By acquiring the dual problems and applying the kernel trick, a kernel method can be immediately generated from 2.34 and 2.35. Finally, after some calculations and the Lagrangian method (for more details see [35]), the following dual QPPs can be defined as:

$$\begin{aligned} \min_{\alpha_+,\alpha_+^*,\beta_2} \quad & \frac{1}{2} \wedge_1 + \epsilon e_+^T(\alpha_-^T + \alpha_-^*) - e_-^T \beta_2, \\ \text{subject to} \quad & e_-^T(\alpha_- - \alpha_-^*) + e_-^T \beta_2 = 0, \\ & 0 \leq \alpha_- \leq c_2 e_+, \; 0 \leq \alpha_-^* \leq c_2 e_+, \\ & 0 \leq \beta_2 \leq c_4 e_-, \end{aligned} \tag{2.36}$$

and

$$\begin{aligned} \min_{\alpha_-,\alpha_-^*,\beta_1} \quad & \frac{1}{2} \wedge_2 + \epsilon e_+^T(\alpha_+^T + \alpha_+^*) - e_+^T \beta_1, \\ \text{such that} \quad & e_+^T(\alpha_+ - \alpha_+^*) - e_+^T \beta_1 = 0, \\ & 0 \leq \alpha_+ \leq c_1 e_+, \; 0 \leq \alpha_+^* \leq c_1 e_+, \\ & 0 \leq \beta_2 \leq c_3 e_-, \end{aligned} \tag{2.37}$$

Where, $\alpha_-, \alpha_-^*, \beta_1$ and $\alpha_+, \alpha_+^*, \beta_2$ are Lagrangian multipliers.

$$\wedge_1 = (\alpha_-^* - \alpha_-)^T K(X_1, X_1^T)(\alpha_- + \alpha_-^*) - 2(\alpha_- + \alpha_-^*)^T K(X_1, X_2^T)\beta_2 + \beta_2^T K(X_2, X_2^T)\beta_2$$

$$\wedge_2 = (\alpha_+^* - \alpha_+)^T K(X_2, X_2^T)(\alpha_+^* - \alpha_+) - 2(\alpha_+ + \alpha_+^*)^T K(X_2, X_1^T)\beta_2 + \beta_2^T K(X_1, X_1^T)\beta_2$$

The decision functions are defined as follows:

$$g_1(x) = K(X_1, x)(\alpha_-^* - \alpha_-)) - K(X_2, x)\beta_2 + b_1$$

and

$$g_2(x) = K(X_2, x)(\alpha_+^* - \alpha_+)) - K(X_1, x)\beta_1 + b_2$$

The final class label is assigned by the following function:

$$\text{class label(x)} = \underset{i=1,2}{\text{argmin}} \frac{|g_i(x)|}{|| \wedge_i ||}. \tag{2.38}$$

2.6 Brief introduction of fuzzy theory, intuitionistic fuzzy theory, and graph embedding theory

This section discusses Fuzzy theory followed by intuitionistic fuzzy theory and graph embedding framework.

2.6.1 Fuzzy theory

For addressing uncertainty issues, Zadeh [304] developed fuzzy set theory. Fuzzy set [305, 306] is an evolution of the traditional notation of a set, often known as a crisp set. The degree of membership of each element of the fuzzy set is defined by the membership function. Let X be a nonempty set. Then A fuzzy set in X can be defined as follows:

$$A = \{(x, \mu_A(x))|x \in A\} \tag{2.39}$$

Where $\mu_A(x)$ is the degree of membership of each element (x) in A. The membership function determines the degree of membership, which is in the range [0, 1] and is defined as: $\mu_A : A \rightarrow [0, 1]$. The tools and technology developed under fuzzy set theory (FST) have been successfully employed with machine learning models in several domains including engineering [307, 308], medical sciences [309, 310], data mining [311] and so on.

2.6.2 Intuitionistic fuzzy theory

The intuitive fuzzy set theory that Atanassov introduced [312] is an expansion of the conventional fuzzy set theory. The traditional membership degree is changed into a membership degree, a nonmembership degree, and a hesitation degree. Finding the best strategy for ranking possibilities based on the given intuitionistic fuzzy information or other associated criteria is known as intuitionistic fuzzy decision-making. Numerous studies have addressed intuitionistic fuzzy decision-making issues to this point, and a variety of decision-making procedures and approaches have been put forth [313]. The intuitionistic fuzzy theory has proved its strength for several tasks such as classification [314], EEG signal classification[315], medical diagnosis [316], and so on. According to Intuitionistic Fuzzy membership (IFM) scheme, each training sample is assigned an intuitionistic fuzzy number (IFN), i.e. (α, α^*), calculated via membership and non-membership functions, respectively. Finally, a score function is defined based on α and α^* values to analyze the outliers in the dataset. Here, the degrees (values) of membership and non-membership functions are calculated in the high-dimensional feature space similar to [317].

- **The Membership function:** The membership function considers the distance between each data sample and its corresponding class centroid in high-dimensional feature space. For each training sample (x_i), the membership function is defined as:

$$\alpha(x_i) = \begin{cases} 1 - \frac{\|\phi(x_i) - D_+\|}{r^+ + \xi}, & y_i = +1 \\ \\ 1 - \frac{\|\phi(x_i) - D_-\|}{r^- + \xi}, & y_i = -1 \end{cases} \tag{2.40}$$

  where $D_+(D_-)$ and $r^+(r^-)$ are the class center and the radius of positive (negative)

class, respectively, and $\xi > 0$ is an adjustable parameter. Here, ϕ is the feature mapping function that maps input space to higher dimensional feature space.

The distance (P) between two samples defined as:

$$P(\phi(x_i), \phi(x_j)) = \|\phi(x_i) - \phi(x_j)\|, \tag{2.41}$$

where $\|.\|$ is the Frobenius norm.

The class center of each class is defined as:

$$D_+ = \frac{1}{n_+} \sum_{y_i=1} \phi(x_i) \text{ and } D_- = \frac{1}{n_-} \sum_{y_i=-1} \phi(x_i), \tag{2.42}$$

where $n_+(n_-)$ denotes the number of positive (negative) class.

The radius of each class is defined as:

$$r^{\pm} = \max_{y_i=\pm 1} \|\phi(x_i) - D_{\pm}\|. \tag{2.43}$$

- **The Non-membership function:** In the IFM scheme, each training sample is also assigned a non-membership degree (value) that gives the proportion between the number of heterogeneous points and the number of all points in its neighborhood. Therefore, the non-membership function is defined as:

$$\alpha^*(x_i) = (1 - \alpha(x_i))\Phi(x_i), \tag{2.44}$$

where $0 \leq \alpha(x_i) + \alpha^*(x_i) \leq 1$ and $\Phi(x_i)$ is calculated as:

$$\Phi(x_i) = \frac{|\{x_j : \|\phi(x_i) - \phi(x_j)\| \leq \eta,\, y_j \neq y_i\}|}{|\{x_j : \|\phi(x_i) - \phi(x_j)\| \leq \eta\}|}, \tag{2.45}$$

where η is the adjustable parameter and and $|\cdot|$ denotes cardinality of a set.

As we can see, the IFN scheme, i.e. calculation of degrees of membership and non-membership functions, is executed based on the inner product distance in high-

dimensional space. Therefore, the kernel function needs to be defined here.

Theorem 1. [317]: Let $K(x, y)$ be the kernel function. Therefore, the inner product distance is defined as:

$$\|\phi(x) - \phi(y)\| = \sqrt{K(x, x) + K(y, y) - 2K(x, y)}. \tag{2.46}$$

Corollary 1. [317]: one can calculate $r^{\pm}$ as follows:

$$(i)\, r^{+} = \max_{y_i=+1} \sqrt{K(x_i, x_i) + \frac{1}{n_+^2} \sum\nolimits_{y_m=+1} \sum\nolimits_{y_n=+1} K(x_m, x_n) - \frac{2}{n_+} \sum\nolimits_{y_j=+1} K(x_i, x_j)},$$

$$(ii)\, r^{-} = \max_{y_i=-1} \sqrt{K(x_i, x_i) + \frac{1}{n_-^2} \sum\nolimits_{y_m=-1} \sum\nolimits_{y_n=-1} K(x_m, x_n) - \frac{2}{n_-} \sum\nolimits_{y_j=-1} K(x_i, x_j)}.$$

- **The score function:** After calculating the degree of membership and non-membership for each sample, the training samples are assigned the IFNs, and training data is given as:

$$X^* = \{(x_1, y_1, \alpha_1, \alpha_1^*), \cdots, (x_N, y_N, \alpha_N, \alpha_N^*)\}.$$

where α_i and α_i^* represent the degree of membership and non-membership of the sample x_i. Finally, a score function based on IFN can be defined as:

$$\Theta_i = \begin{cases} \alpha_i, & \alpha_i^* = 0; \\ 0, & \alpha_i \leq \alpha_i^*; \\ \frac{1-\alpha_i^*}{2-\alpha_i-\alpha_i^*}, & \text{others.} \end{cases} \tag{2.47}$$

2.6.3 Graph embedding

Dimension reduction (DR), which tries to extract low-dimensional features from high-dimensional data, has drawn an increasing amount of interest in the computer vision and machine learning domains. Since the quality of the input features heavily influences how well machine learning algorithms generalize [318], preprocessing the original data using

DR approaches is frequently necessary to improve generalization performance. A vast family of algorithms—supervised or unsupervised; originating from statistics or geometry theory—has been developed over the past few decades to offer various approaches to the problem of dimensionality reduction. Despite the disparate driving forces behind these algorithms, Yan et al. [74] proposed a general formulation known as graph embedding to bring them all together within a single framework. Each algorithm for graph embedding can be thought of as a direct graph embedding or its linear, kernel, or tensor extension of a particular intrinsic graph that characterizes desired statistical or geometric properties of a data set, with constraints imposed by scale normalization or a penalty graph that identifies undesirable statistical or geometric properties. The graph embedding architecture can also be utilized as a broad platform for creating new DR algorithms. Graph embedding has been successfully utilized for various tasks such as GDR-ELM [319], a graph embedding-based DR framework for DR problems, classification [320], and so on.

In graph embedding framework, there is an undirected weighted graph $G = \{X, \Omega\}$, where $X \in \mathbb{R}^{N \times m}$ and $\Omega \in \mathbb{R}^{N \times N}$ are input data and similarity matrix, respectively. Each element in matrix Ω represents the relationship between two graph vertices x_i. Moreover, another penalty graph $G^p = \{X, \Omega^p\}$, here, each element of weight matrix $\Omega^p \in \mathbb{R}^{N \times N}$ represents the penalty weights that are given to the specific relationship between the graph vertices x_i in X. The optimization problem of the graph embedding is defined as:

$$\begin{aligned} w^* &= \underset{tr({w_0}^T X^T V X w_0)=q}{\operatorname{argmin}} \sum_{i \neq j} \left\| {w_0}^T x_i - {w_0}^T x_j \right\|^2 \Omega_{ij} \\ &= \underset{tr({w_0}^T X^T V X w_0)=q}{\operatorname{argmin}} tr({w_0}^T X^T L X w_0). \end{aligned} \tag{2.48}$$

Here, tr (.) is the trace operator, w_0 is the transformation matrix, $L \in \mathbb{R}^{N \times N}$ is the graph Laplacian matrix, i.e., $L = D - \Omega$ of the intrinsic graph G and D is the diagonal matrix wherein each diagonal element $D_{ii} = \sum_j \Omega_{ij}$. Typically, either V is a diagonal matrix for scale normalization or it is the Laplacian matrix of penalty graph G^p, i.e., $V = L^p = D^p - \Omega^p$ and q is a constant. The solution of problem (2.48) is obtained by solving generalized

eigenvalue problem [321],

$$S_i z = \lambda S_p z, \tag{2.49}$$

here, $\mathbf{S}_i = X^T L X$ and $S_p = X^T V X$. It means that eigen vectors of the matrix $S = S_p^{-1} S_i$ will form the transformation matrix. The matrix S considers both the intrinsic and penalty graph relationships within the dataset.

2.7 Decision trees and their ensemble learning

Decision trees [322] are sequential models that logically combine a series of straightforward tests; each test contrasts a nominal or numeric property with a range of potential values. In terms of comprehension, such symbolic classifiers outperform "black-box" techniques like neural networks. The data is iteratively partitioned using the divide and conquer strategy by the decision tree. An unstable classifier is produced by the tree's recursive partition, which is susceptible to changes in the input data. As a result, decision trees are deemed to have high variance and low bias. In order to further enhance the classification performance, unstable classifiers can be utilized with the ensemble methodology [323]. By combining the predictions from two or more base models, ensemble learning algorithms have achieved state-of-the-art performance in a variety of machine learning applications [324]. Amasyali [73] proposed a meta ensemble method (improved space method) in which new features are generated using original features and a new feature space is constructed by concatenating original features and newly generated features, followed by training of each base model (DT) over new feature space. The improved space method is employed over bagging [122], random forest [40] and rotation forest [42] ensemble methods and improved their respective generalization performances. In the literature, multiple techniques [38, 132] have been proposed to construct efficient ensemble models. Recently, similar works have been done for randomized neural networks and their ensemble methods. A brief overview of random forest and rotation forest is given as follows:

2.7.1 Random forest

Originally given by Breiman [40], Random forest (RaF) uses the idea of bagging and random subspaces. RaF relies on a series of tree predictors, where each tree takes values from randomly initialized vectors which are sampled independently and have the same distribution across all the trees in a forest. Diversity among the base classifiers is increased by combining the concepts of bagging and random subspaces. Each tree is trained on the bootstrapped version of the training data with random subspace of features. The number of random subspace features selected controls the number of tests (split tests) to be performed at each node, as each feature is evaluated for the split. Among these features, the one that makes the node more pure is selected. The algorithm of RaF is given in Algorithm 2.1.

Algorithm 2.1 Random forest.

Training Phase:

Given:

$X = M \times n$ is the dataset with M samples each with feature length n.
$Y = M \times 1$ are data labels corresponding to the training dataset.
L is the ensemble size i.e. number of trees in the forest.
Each tree in the random forest is represented as T_i, where $i = 1, \cdots, L$.
"$mtry$" refers to the randomly selected features for splitting at each non-leaf node.
"$minleaf$" is the maximum number of samples in an impure node.

1: Each tree T_i is build using the bootstrapped versions of the training data X with replacement.
2: At each non-leaf node, the best feature split is selected among the "$mtry$" randomly selected features from the training data.
3: Repeatedly execute step 2 until one of the conditions is met:

- Node becomes pure.
- Node contains number of samples less than or equal to $minleaf$.

Classification Phase:

For the classification of a test sample, it is pushed down each tree in the forest, and each tree in the forest assigns the vote to the given sample data. Then the predicted label for the sample data is the one with the highest number of votes among the forest.

2.7.2 Rotation forest

Before constructing each tree, rotation forest (RoF) [325] uses PCA to transform or rotate the dataset. The different decision trees in the forest are uncorrelated as each tree uses a distinct rotation matrix. The heuristics used in RoF is that features are extracted from a subset of features and then a full feature set is reconstructed for each classifier. The authors used all the principal components to construct the feature subspace. The diversity of the model comes with the difference in the possible feature subsets. With rotation heuristic, the number of different partitions of the feature set $T = \frac{n!}{K!(M!)^K}$, where K is the number of subsets of size M, and n is the sample feature length. Different classifiers are generated on different partitions. Under the assumption that partitions of the feature sets are equally likely, the probability that all classifiers will be different is P (different classifiers)$= \frac{T!}{(T-L)!T^L}$, where L is the ensemble size.

The algorithm of RaF is given in Algorithm 2.2.

2.8 A brief overview of Alzheimer's disease

The brain is the most intricate and significant organ in the human body, therefore, disorders affecting it are significant from a medical point of view. According to the data, dementia is a collection of symptoms that worsen with time and a chronic impairment of brain function. The fact that dementia is both incurable and irreversible presents a considerable challenge. Alzheimer's Disease (AD), which was first identified by a German neurologist named A. Alzheimer is one of the most prominent causes of dementia [326]. Dementia is not a single disease but rather a collection of several disorders and progressive symptoms, among which AD is by far the most common disorder. Around the world, there are more than five crore dementia patients, out of which 70 percent are Alzheimer's disease patients [327]. AD is an incurable disorder that alters the biology of the brain and results in structural degeneration. Degeneration may result in memory loss, a drop in mental capacity, and a loss of social skills. The severity of AD may possibly cause a person's death. Temporary memory loss is one of the earliest and perhaps hard to recognize signs of AD. Mild Cognitive Impairment (MCI), a disease stage, exists somewhere between Cognitive Normal (CN) and

Algorithm 2.2 Rotation forest.

Training Phase:

Given: $X = M \times n$ is the dataset with M samples each with feature length n.
$Y = M \times 1$ are class labels corresponding to the training data set.
$L :=$ Number of decision trees in an ensemble.
S is the number of subsets and $\{C_1, \cdots, C_k\}$ is the set of class labels.
F is the feature set.
For $i = 1, \cdots, L$

1: Prepare the rotation matrix R_i^a :

- Split F into S subsets: $F_{i,j}$ with $j = 1, \cdots, S$.
- For $j = 1, \cdots, S$
  - Let $X_{i,j}$ be the data set X for the features in $F_{i,j}$.
  - Eliminate a random subset of classes from $X_{i,j}$.
  - Select a bootstrap sample from $X_{i,j}$ of size 75% of the number of samples in $X_{i,j}$. Denote the new set as $X'_{i,j}$.
  - Apply PCA on $X'_{i,j}$ to get the coefficients in a matrix $Z_{i,j}$.
- Arrange the $Z_{i,j}$, for $j = 1, \cdots, S$ in a rotation matrix R_i.
- For construction of R_i^a, rearrange the columns of R_i to match order of the features in F.

2: With (XR_i^a, Q) as the training data, build the classifier D_i.

Classification Phase:

- For a given sample x, let the classifier D_i assigned the probability $d_{i,j}(xR_i^a)$ to the hypothesis that x belongs to class C_j. For each class C_j, calculate the confidence by the average combination method:

$$\mu_j(x) = \frac{1}{L} \sum_i d_{i,j}(xR_i^a), \quad j = 1, \cdots, k.$$

- Assign x to the class with the largest confidence.

AD. Early disease detection can assist in providing patients with optimal care, postponing AD's negative effects, and slowing the progression of dementia. The application of machine learning models in the diagnosis of AD has been a recent trend. Multiple machine learning models [328] have been proposed for the diagnosis of AD [61, 329]. Orouskhani et al. [330] employed deep triplet network with structural MRI for Alzheimer's disease detection and got better accuracy than the compared state-of-art models. Amini et al. [331] proposed a novel technique that includes fMRI images, robust multitask feature extraction method, and convolutional neural network (CNN) for diagnosis of AD. Recent studies for the diagnosis of AD disease includes [332, 333]. For the details of the machine learning models for the diagnosis of AD, we refer the interested readers to [334].

2.9 Statistical tests

In this section, we discuss the statistical tests used to evaluate the performance of the models statistically.

2.9.1 Friedman test

Friedman test has been proven to be more robust than other approaches and has been practiced by numerous researchers [269, 335] to check the statistical significance among classifiers. This method ranks the algorithms for each dataset separately, the best-performing algorithm on each dataset achieves the lower rank. Then, the average of the rank across all the datasets is taken as the rank of the classifier. Let r_i^j be the rank of the j^{th} algorithm on the i^{th} dataset among k algorithms and N datasets. The Friedman test compares the average rank of algorithms, $R_j = \frac{1}{N}\sum_i r_i^j$. Under the null hypothesis, which states that all the algorithms are equivalent and so their rank R_j should be equal. If k is the number of algorithms and N is the number of datasets then Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{2.50}$$

is distributed according to χ_F^2 with $(k-1)$ degrees of freedom, when N and k are big

enough. In that case, Friedman statistic χ_F^2 is undesirably conservative and derived a better statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}, \tag{2.51}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. If the null hypothesis is rejected, Nemenyi test [336] can be used to check whether the performance of two among k classifiers are significantly different. Two classifiers are said to be statistically different if the rank of two classifiers differs by at least a critical difference (CD). Mathematically, the critical difference is given by

$$\text{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{2.52}$$

where critical value q_α is based on the studentized range statistic divided by $\sqrt{2}$.

2.9.2 Win-tie-loss: sign test

To access the overall performances of classifiers, we count the number of datasets on which algorithm is the overall winner. We use the sign test for the pairwise comparison of algorithms. In this test, under null hypothesis two algorithms are equivalent if each wins on approximately $N/2$ out of N datasets. If the number of wins is at least $N/2 + 1.96\sqrt{N}/2$, the algorithm is significantly better with $p < 0.05$. If the two algorithms end with a tie, then the number should be evenly splitted between the classifiers. However, if the number is odd we ignore one.

Chapter 3

Fuzzy Theory and Graph Embedding based Random Vector Functional Link Network

This chapter aims to develop robust shallow RVFLs. We employ fuzzy/intuitionistic fuzzy theory to handle the outliers and noisy samples in the data, and graph embedding theory is used to consider the geometrical relationship of the data in the training process. In the first approach, we propose a novel intuitionistic fuzzy random vector functional link network (IFRVFL) [76]. Unlike standard RVFL, ELM, and KRR which use uniform weighting approach for generating the optimal classifier, the proposed IFRVFL uses fuzzy weighting approach for calculating the final output parameters of the classifier. Under a uniform weighting approach, standard RVFL assigns equal weights to each sample while training and hence has less generalization performance over datasets with noise or outliers. The proposed IFRVFL assigns each sample an intuitionistic fuzzy number which is calculated using the membership and non-membership score of a sample. The membership score is a function of the sample distance from the centroid of its corresponding class and the non-membership score is a function of sample distance from the centroid as well as the neighbourhood information of the given sample. Therefore, the proposed IFRVFL is more efficient than standard RVFL.

Literature shows that fuzzy theory enhances machine learning models' ability to handle

noise in the data. A key challenge in fuzzy approaches is how to define a fuzzy membership value for each training sample so that the model can discriminate between noise samples and normal samples. Therefore, in the second approach, our objectives are to design an efficient fuzzy function for the classification problem and to develop a robust fuzzy RVFL model. As a result, we propose a class probability-based efficient score function that uses a non-linear function as membership function to determine the membership value of each sample. The proposed score function uses the k-NN technique to determine the class probability to take the neighbourhood information of each training sample. Finally, a score value is assigned to each sample based on the score function, which combines the membership value and class probability value. By incorporating the proposed score function in the objective function of the standard RVFL network, we propose a novel class probability-based fuzzy RVFL (CP-FRVFL).

The proposed IFRVFL model doesn't take into account the geometrical relationship of the data while calculating the final output parameters and hence, it affects the generalization performance of the model. In literature, graph embedded (GE) approaches are used successfully to describe the geometrical relationship within the data. Therefore, in the last approach of this Chapter, we propose graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) [77]. In GE-IFWRVFL, a novel regularization term based on GE framework is introduced into the optimization problem of IFRVFL. The experimental results demonstrate that the proposed GE-IFWRVFL has better generalization performance compared to the baseline models.

3.1 Proposed intuitionistic fuzzy RVFL (IFRVFL) network

The standard RVFL gives equal weight to each sample while calculating the final parameters and hence, gets affected negatively by outliers and noise in the dataset. The proposed IFRVFL network addresses the aforementioned issue of standard RVFL. The IFTWSVM model solves QPPs while the proposed IFRVFL model solves the system of linear equations

for calculating the final parameters. The kernel-based methods such as KRR and intuitionistic fuzzy KRR (IFKRR) suffer from memory issues when number of samples are large. On the other hand, both RVFL and ELM are randomized neural networks that give uniform weights to each sample for calculating the optimal classifier and hence, are not robust to outliers in the dataset. However, the proposed IFRVFL model assigns an intuitionistic fuzzy value, i.e., by using a function of membership and non-membership values, to each sample in the dataset so that the outliers and noise can be handled effectively. The membership value for each sample is measured by calculating the distance between sample and its corresponding class centroid, while the non-membership value considers the relationship between the number of disharmonious samples and the number of samples in its neighbourhood. Using these assigned values to each sample, finally, a score function is defined to handle the outliers and noise in the datasets. The proposed IFRVFL model effectively reduces the negative influence of outliers, and hence has better generalization performance compared to standard RVFL.

3.1.1 Formulation of IFRVFL network

The optimization problem of the proposed IFRVFL is defined as:

$$\begin{aligned} &\min_{\beta} \frac{1}{2}\|\beta\|_2^2 + \frac{1}{2}C\|S\xi\|_2^2 \\ &\text{subject to } H\beta - Y = \xi, \end{aligned} \tag{3.1}$$

where $H = [h(x_1), h(x_2), \ldots, h(x_N)]^T$, $h(x_i)^T = [x_i\ \Psi(x_i)]$; $S = diag(\Theta_1, \Theta_2, \ldots, \Theta_N)$ is the diagonal matrix that contains the final score values at diagonals (see Chapter 2), and $\xi = [\xi_1, \xi_2, \ldots, \xi_N]^T$ is the error term corresponding to N samples. Also, C is the regularization parameter and β is the output weights matrix. The Eq. (3.1) can be reformulated as follows:

$$\min_{\beta \in \mathbb{R}^{(d+L)\times m}} \frac{1}{2}\|\beta\|_2^2 + \frac{C}{2}\|S(H\beta - Y)\|_2^2 . \tag{3.2}$$

The optimization problem in (3.2) is a convex quadratic problem. Therefore, there exists a unique global optimal solution to this problem. Let

$$L(\beta) = \frac{1}{2}\|\beta\|_2^2 + \frac{C}{2}\|S(H\beta - Y)\|_2^2. \tag{3.3}$$

Taking the derivative of $L(\beta)$ with respect to β and set it equal to zero, we have

$$\frac{\partial L}{\partial \beta} = \beta + C(SH)^T(S(H\beta - Y)) = 0. \tag{3.4}$$

If number of hidden nodes is lesser than the number of training samples, i.e., $((d+L) < N)$, then one can calculate β as follows:

$$\beta = \Big(\frac{1}{C}I + (SH)^T(SH)\Big)^{-1}(SH)^T SY. \tag{3.5}$$

Here, I is an identity matrix of order $d + L$. For $((d + L) > N)$, substitute $\beta = (SH)^T\mu$ in Eq. (3.4), then we get,

$$(SH)^T\mu + C(SH)^T\big(S(H(SH)^T\mu - Y)\big) = 0. \tag{3.6}$$

After calculations, we have:

$$\mu = \Big(\frac{1}{C}I + SHH^TS\Big)^{-1}SY, \tag{3.7}$$

$$= S^{-1}\Big(\frac{1}{C}S^{-1}S^{-1} + HH^T\Big)^{-1}Y. \tag{3.8}$$

Finally,

$$\beta = (SH)^T\mu = H^TS\mu = H^T\Big(\frac{1}{C}S^{-1}S^{-1} + HH^T\Big)^{-1}Y. \tag{3.9}$$

Therefore, the optimal solution of (3.2) is as follows:

$$\beta = \begin{cases} \Big(\frac{1}{C}I + (SH)^T(SH)\Big)^{-1}(SH)^TSY, & (d+L) \le N, \\ H^T\Big(\frac{1}{C}S^{-1}S^{-1} + HH^T\Big)^{-1}Y, & N < (d+L). \end{cases} \tag{3.10}$$

In case, S is a singular matrix then Tikhonov regularization can be used.

3.1.2 Computational complexity

Let N be the number of samples in a dataset. Since the membership and non membership values are calculated for each data sample, hence the complexity of calculating these values is $O(N)$ [27]. The complexity of the proposed IFRVFL is mainly determined by the computation of matrix inversion. By the standard procedure, the complexity of the inversion of a matrix of order N is $O(N^3)$ [337]. Thus, the complexity of IFRVFL is $O(N^3) + O(N) \approx O(N^3)$ for large N.

3.2 Experiments

In this section, we evaluate the baseline models and the proposed IFRVFL model on ADNI, UCI, and KEEL datasets.

3.2.1 Experimental setup

All the experiments are performed on a system with MATLAB R2017b, Intel(R) Xeon(R) CPU E5-2697 v4 2.30 GHz, 128-GB RAM, and Windows-10 platform. For generating the intuitionistic fuzzy weights in IFTWSVM, IFKRR, and the proposed IFRVFL models, the samples are transformed into higher dimensional space via kernel function. We used Gaussian kernel, $K(x_1, x_2) = \exp(-\frac{||x_1-x_2||^2}{\mu^2})$, where μ denotes the kernel parameter. The dataset is randomly partitioned into $30 : 70$ ratio of testing and training sets, respectively. The hyperparameters corresponding to different models are optimized using a grid search approach via 5-fold cross validation. In 5-fold cross validation, the dataset is randomly partitioned into 5 disjoint sets wherein 1 set is reserved for testing and the rest are used for training. The performance of each model corresponding to the best hyperparameters is given as the final accuracy of a model. The different parameters involved corresponding to different models are chosen from the following range: $C_i = \{10^{-5}, 10^{-4}, \cdots, 10^4, 10^5\}$, for $i = 1, 2, 3, 4$, number of hidden neurons are taken

from the range as: $N = 3 : 20 : 203$. For ELM, RVFL, and the proposed IFRVFL models, we used 8 activation functions, namely, selu, relu, sigmoid, sin, hardlim, tribas, radbas and sign function.

The performance measures used to evaluate the algorithms are given as follows:

$$\text{Accuracy, AUC} = \frac{TP + TN}{TP + FP + TN + FN}$$
$$\text{Sensitivity or Recall} = \frac{TP}{TP + FN}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{G-mean} = \sqrt{\text{Precision} \times \text{Recall}}$$
$$\text{Specificity} = \frac{TN}{TN + FP}$$

where false positive, true positive, false negative, and true negative are denoted by FP, TP, FN and TN, respectively.

3.2.2 Evaluation on ADNI dataset

ADNI repository (adni.loni.usc.edu) scans are used in this study. The Principal Investigator of ADNI project, Michael W. Weiner, launched it in 2003 with the aim to analyze neuroimaging approaches like positron emission tomography (PET), magnetic resonance imaging (MRI), other tests for the diagnosis of AD from mild cognitive impairment (MCI) stage. For further details, we refer the interested readers to `www.adni-info.org`. We used Volume based morphometry (VolBM) features. The pipeline for feature extraction followed is the same as given in [338]. The performance of the models is evaluated in terms of accuracy for classifying the MCI versus AD cases (MCI_vs_AD), control normal (CN) versus AD cases (CN_vs_AD) and CN versus MCI cases (CN_vs_MCI).

The performances of the models for the diagnosis of AD with 'relu' activation function are given in Table 3.1. One can see that the KRR and the proposed IFRVFL model show the best performance on CN_vs_AD and CN_vs_MCI subjects. Here, KRR model

Table 3.1: Algorithm's performance for the diagnosis of AD.

Subjects	IFTWSVM [27] (AUC, Time) (c_1, c_3, μ)	KRR [339] (AUC, Time) (c, μ)	IFKRR [340] (AUC, Time) (c, μ)	ELM [341] (AUC, Time) (c, N)	RVFL [11] (AUC, Time) (c, N)	IFRVFL (AUC, Time) (c, μ, N)
CN_vs_AD case	$(0.8801, 0.0335)$ $(0.00001, 0.001, 32)$	$(\mathbf{0.9186}, 0.0036)$ $(0.1, 32)$	$(0.7737, 0.0101)$ $(0.00001, 2)$	$(0.8444, 0.0022)$ $(0.0001, 183)$	$(0.8903, 0.0008)$ $(0.01, 3)$	$(\mathbf{0.898}, 0.0168)$ $(0.1, 32, 3)$
CN_vs_MCI case	$(0.6623, 0.0817)$ $(100000, 1, 32)$	$(\mathbf{0.6775}, 0.0098)$ $(1, 32)$	$(0.6329, 0.0293)$ $(0.1, 4)$	$(0.6318, 0.0027)$ $(0.001, 183)$	$(0.6295, 0.0009)$ $(0.01, 3)$	$(\mathbf{0.6763}, 0.0276)$ $(0.1, 8, 163)$
MCI_vs_AD case	$(0.6212, 0.0723)$ $(100000, 1, 32)$	$(0.5797, 0.0088)$ $(1, 32)$	$(0.5787, 0.0278)$ $(0.1, 4)$	$(0.5858, 0.0009)$ $(10, 63)$	$(\mathbf{0.6621}, 0.0014)$ $(0.1, 43)$	$(\mathbf{0.6813}, 0.4195)$ $(100000, 0.03125, 3)$
Average AUC	0.7212	0.7253	0.6618	0.6873	**0.7273**	**0.7519**

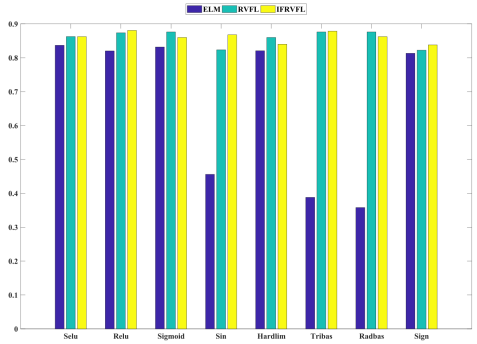
Here, boldface denotes the performance of the top two models.

has winning performance with AUC equal to 91.86% and 67.75% for the case CN_vs_AD and CN_vs_MCI, respectively. The proposed IFRVFL model has second position with AUC equal to 89.8% and 67.63% for the case CN_vs_AD and CN_vs_MCI, respectively. Also, the accuracy of other compared models, i.e., RVFL, IFTWSVM, ELM and IFKRR, are $89.03\%, 88.01\%, 84.44\%$ and 77.37%, respectively on CN_vs_AD subject which is inferior in comparison with the proposed IFRVFL model. For CN_vs_MCI subjects, the IFTWSVM, IFKRR, ELM and RVFL has AUC equal to $66.23\%, 63.029\%, 63.18\%$ and 62.95%, respectively which is lower compared to the proposed IFRVFL model. For MCI_vs_AD subject, the proposed IFRVFL model emerged as the best classifier with AUC equal to 68.13%, followed by RVFL and IFTWSVM with AUC equal to 66.21% and 62.12%, respectively. The rest of the models, i.e., KRR, IFKRR and ELM have AUC equal to 57.97%, 57.87% and 58.58%, respectively on MCI_vs_AD subject. Thus, on MCI_vs_AD subject the proposed IFRVFL model showed approximately. 10% more accuracy compared to the KRR, IFKRR and ELM models. Moreover, in terms of average accuracy, the proposed IFRVFL model emerged as the best classifier with AUC equal to 75.19% followed by RVFL model with AUC equal to 72.73%. The overall performance of the models (with 'relu' activation function for ELM, RVFL and IFRVFL) in terms of F-measure is given in Figure 3.1d. It is clear that the performance of the proposed IFRVFL model is competitive or better among the baseline models. Figure 3.2d gives the analysis of the models (with 'relu' activation function for ELM, RVFL and IFRVFL).

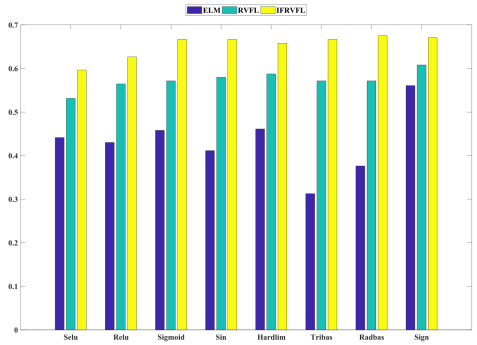
Activation functions play an important role in the performance of neural networks. Hence, we used 8 activation functions to check the suitability of each for the diagnosis of AD. The performance of ELM, RVFL and IFRVFL models on different activation functions

Table 3.2: The performance analysis of ELM, RVFL and IFRVFL model for the diagnosis of AD across different activation functions.

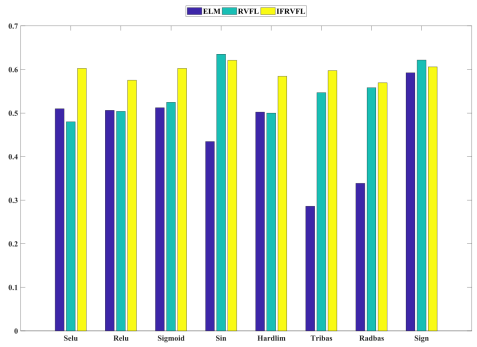
Activation	Model	Subjects	AUC	Sens.	Spec.	Prec.
Selu	ELM	CN_vs_AD case	0.859	0.8604	0.8575	0.815
Selu	ELM	CN_vs_MCI case	0.635	0.5246	0.7453	0.497
Selu	ELM	MCI_vs_AD case	0.5946	0.3785	0.8107	0.5366
Selu	RVFL	CN_vs_AD case	0.8808	0.8302	0.9315	0.898
Selu	RVFL	CN_vs_MCI case	0.6131	0.4918	0.7344	0.4688
Selu	RVFL	MCI_vs_AD case	0.6321	0.5231	0.7411	0.5397
Selu	IFRVFL	CN_vs_AD case	0.8818	0.8868	0.8767	0.8393
Selu	IFRVFL	CN_vs_MCI case	0.6977	0.7705	0.625	0.4947
Selu	IFRVFL	MCI_vs_AD case	0.6549	0.7385	0.5714	0.5
Relu	ELM	CN_vs_AD case	0.8444	0.834	0.8548	0.8083
Relu	ELM	CN_vs_MCI case	0.6318	0.5246	0.7391	0.4894
Relu	ELM	MCI_vs_AD case	0.5858	0.3662	0.8054	0.5263
Relu	RVFL	CN_vs_AD case	0.8903	0.8491	0.9315	0.9
Relu	RVFL	CN_vs_MCI case	0.6295	0.5246	0.7344	0.4848
Relu	RVFL	MCI_vs_AD case	0.6621	0.5385	0.7857	0.5932
Relu	IFRVFL	CN_vs_AD case	0.898	0.9057	0.8904	0.8571
Relu	IFRVFL	CN_vs_MCI case	0.6763	0.6885	0.6641	0.4941
Relu	IFRVFL	MCI_vs_AD case	0.6813	0.8	0.5625	0.5149
Sigmoid	ELM	CN_vs_AD case	0.8547	0.8491	0.8603	0.8159
Sigmoid	ELM	CN_vs_MCI case	0.6324	0.5508	0.7141	0.4789
Sigmoid	ELM	MCI_vs_AD case	0.5896	0.4185	0.7607	0.5095
Sigmoid	RVFL	CN_vs_AD case	0.8929	0.8679	0.9178	0.8846
Sigmoid	RVFL	CN_vs_MCI case	0.649	0.5246	0.7734	0.5246
Sigmoid	RVFL	MCI_vs_AD case	0.6653	0.5538	0.7768	0.5902
Sigmoid	IFRVFL	CN_vs_AD case	0.8783	0.8113	0.9452	0.9149
Sigmoid	IFRVFL	CN_vs_MCI case	0.6977	0.7705	0.625	0.4947
Sigmoid	IFRVFL	MCI_vs_AD case	0.7284	0.7692	0.6875	0.5882
Sin	ELM	CN_vs_AD case	0.497	0.4981	0.4959	0.4214
Sin	ELM	CN_vs_MCI case	0.5382	0.5639	0.5125	0.3538
Sin	ELM	MCI_vs_AD case	0.4852	0.4954	0.475	0.3526
Sin	RVFL	CN_vs_AD case	0.8483	0.7925	0.9041	0.8571
Sin	RVFL	CN_vs_MCI case	0.729	0.7705	0.6875	0.5402
Sin	RVFL	MCI_vs_AD case	0.6604	0.6154	0.7054	0.5479
Sin	IFRVFL	CN_vs_AD case	0.886	0.8679	0.9041	0.8679
Sin	IFRVFL	CN_vs_MCI case	0.7145	0.8197	0.6094	0.5
Sin	IFRVFL	MCI_vs_AD case	0.7284	0.7692	0.6875	0.5882
Hardlim	ELM	CN_vs_AD case	0.8449	0.8377	0.8521	0.8046
Hardlim	ELM	CN_vs_MCI case	0.6378	0.4787	0.7969	0.5299
Hardlim	ELM	MCI_vs_AD case	0.6019	0.4092	0.7946	0.5325
Hardlim	RVFL	CN_vs_AD case	0.8783	0.8113	0.9452	0.9149
Hardlim	RVFL	CN_vs_MCI case	0.6326	0.4918	0.7734	0.5085
Hardlim	RVFL	MCI_vs_AD case	0.6775	0.5692	0.7857	0.6066
Hardlim	IFRVFL	CN_vs_AD case	0.862	0.7925	0.9315	0.8936
Hardlim	IFRVFL	CN_vs_MCI case	0.6814	0.7377	0.625	0.4839
Hardlim	IFRVFL	MCI_vs_AD case	0.7207	0.7538	0.6875	0.5833
Tribas	ELM	CN_vs_AD case	0.4752	0.3887	0.5616	0.3899
Tribas	ELM	CN_vs_MCI case	0.4802	0.2885	0.6719	0.2898
Tribas	ELM	MCI_vs_AD case	0.5039	0.2738	0.7339	0.3679
Tribas	RVFL	CN_vs_AD case	0.8929	0.8679	0.9178	0.8846
Tribas	RVFL	CN_vs_MCI case	0.6619	0.5738	0.75	0.5224
Tribas	RVFL	MCI_vs_AD case	0.6584	0.5846	0.7321	0.5588
Tribas	IFRVFL	CN_vs_AD case	0.8955	0.8868	0.9041	0.8704
Tribas	IFRVFL	CN_vs_MCI case	0.6935	0.7541	0.6328	0.4946
Tribas	IFRVFL	MCI_vs_AD case	0.7271	0.7846	0.6696	0.5795
Radbas	ELM	CN_vs_AD case	0.4734	0.3358	0.611	0.3856
Radbas	ELM	CN_vs_MCI case	0.521	0.3279	0.7141	0.3528
Radbas	ELM	MCI_vs_AD case	0.5152	0.3662	0.6643	0.3897
Radbas	RVFL	CN_vs_AD case	0.8929	0.8679	0.9178	0.8846
Radbas	RVFL	CN_vs_MCI case	0.6701	0.5902	0.75	0.5294
Radbas	RVFL	MCI_vs_AD case	0.6584	0.5846	0.7321	0.5588
Radbas	IFRVFL	CN_vs_AD case	0.8818	0.8868	0.8767	0.8393
Radbas	IFRVFL	CN_vs_MCI case	0.6657	0.7377	0.5938	0.4639
Radbas	IFRVFL	MCI_vs_AD case	0.7361	0.7846	0.6875	0.593
Sign	ELM	CN_vs_AD case	0.8387	0.8226	0.8548	0.8043
Sign	ELM	CN_vs_MCI case	0.6903	0.718	0.6625	0.505
Sign	ELM	MCI_vs_AD case	0.6504	0.5723	0.7286	0.5513
Sign	RVFL	CN_vs_AD case	0.8466	0.8302	0.863	0.8148
Sign	RVFL	CN_vs_MCI case	0.7169	0.7541	0.6797	0.5287
Sign	RVFL	MCI_vs_AD case	0.6941	0.5846	0.8036	0.6333
Sign	IFRVFL	CN_vs_AD case	0.8603	0.8302	0.8904	0.8462
Sign	IFRVFL	CN_vs_MCI case	0.6989	0.8197	0.5781	0.4808
Sign	IFRVFL	MCI_vs_AD case	0.7291	0.8154	0.6429	0.5699



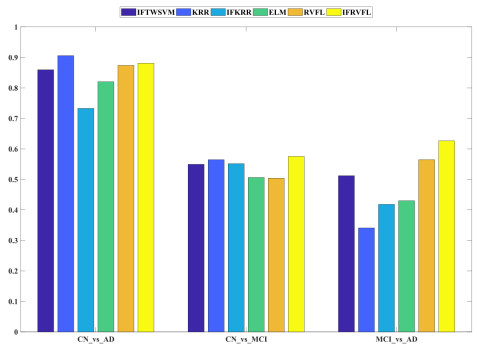
(a) CN_vs_AD



(b) MCI_vs_AD

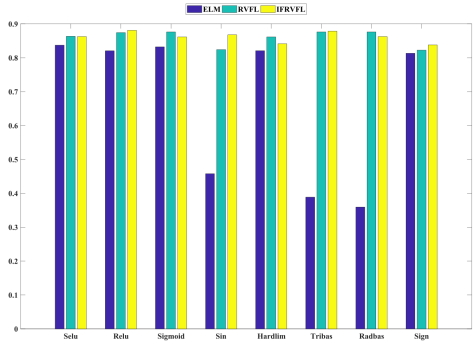


(c) CN_vs_MCI

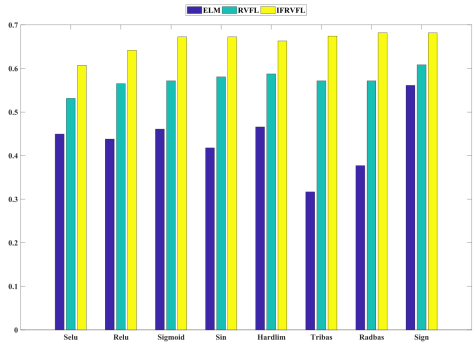


(d) All Models

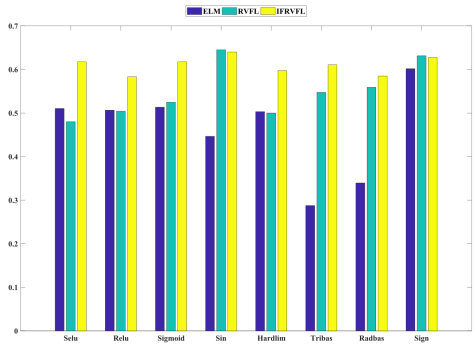
Figure 3.1: F-measure analysis of the classification models for AD.



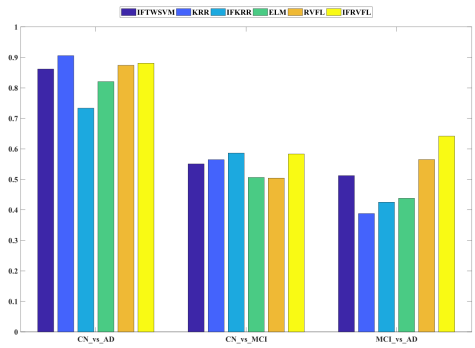
(a) CN_vs_AD



(b) MCI_vs_AD



(c) CN_vs_MCI



(d) All Models

Figure 3.2: G-mean analysis of the classification models for AD.

is given in Table 3.2.

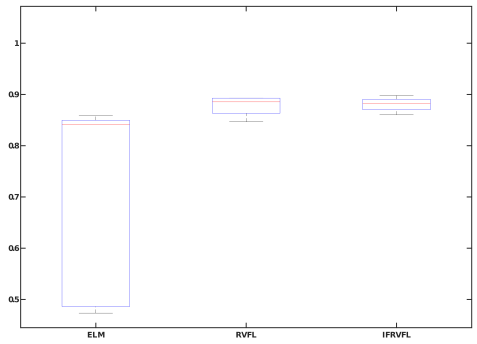
For CN versus AD subject, The proposed IFRVFL model has highest (among all models) AUC equal to 89.8% with 'relu' activation function and RVFL has best accuracy equal to 89.29% with 'sigmoid', 'tribas' and 'radbas' activation functions. For MCI versus AD subject, ELM has poor performance (among all models) with AUC equal to 48.52% with 'sin' activation function and best performance 65.04% accuracy with 'sign' activation function. The proposed IFRVFL model has overall winning performance with AUC equal to 73.61% with 'radbas' activation function. For CN versus MCI subject, RVFL with 72.9% accuracy is overall winner (among all models) with 'sin' activation function. Figure 6.3a to Figure 3.1c gives the F-measure analysis of the models (NNs) with different activation functions. For CN_vs_AD subject, the proposed IFRVFL and RVFL models have competition and for MCI_vs_AD and CN_vs_MCI subjects, the proposed IFRVFL model has overall winning performance.

Figure 3.2a to Figure 3.2c gives the G-mean analysis of the models (NNs) across different activation functions. For CN_vs_AD subject, the proposed IFRVFL model has best performance with 'selu', 'relu', 'sin', 'tribas' and 'sign' activation functions. For MCI_vs_AD subject, the proposed IFRVFL model is overall winner and ELM has lower performance with all activation functions. For CN_vs_MCI subject, the proposed IFRVFL model is performing best except with 'sin' and 'sign' activation functions. Figure 3.3 shows the overall generalization of the models across different activation functions. From the Figure, it is clear that the proposed IFRVFL model shows superior performance compared to ELM and RVFL models in MCI_vs_AD subject. Moreover, in CN_vs_AD and CN_vs_MCI subjects, the proposed IFRVFL model is competitive to RVFL model and better than the ELM model.

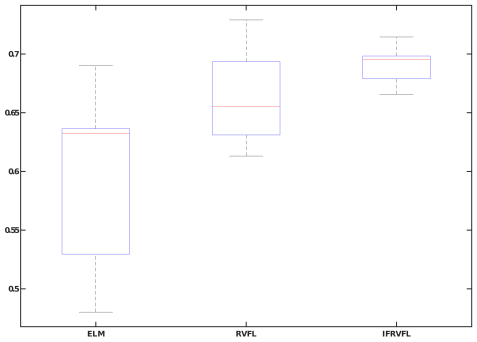
3.2.3 Evaluation on UCI and KEEL datasets

To check the overall performance of the models, we evaluated them on benchmark datasets from the UCI [342] and KEEL [343] repository. The classification performance of the models in terms of AUC is given in Table 3.3.

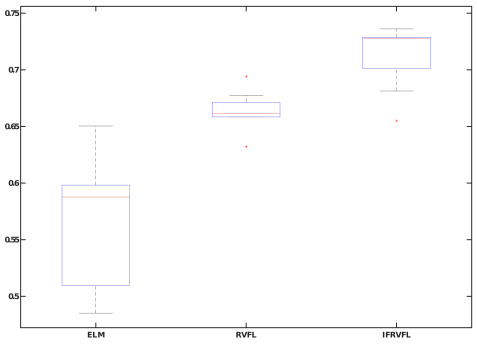
From this Table, it is clear that the proposed IFRVFL model is best with average accu-



(a) CN_vs_AD



(b) CN_vs_MCI



(c) MCI_vs_AD

Figure 3.3: Performance evaluation of ELM, RVFL and proposed IFRVFL models on AD.

Table 3.3: Evaluation of algorithms on UCI and KEEL datasets.

Dataset Name	IFTWSVM [27] (AUC, Time) (c_1, c_3, μ)	KRR [339] (AUC, Time) (c, μ)	IFKRR [340] (AUC, Time) (c, μ)	ELM [341] (AUC, Time) (c, N)	RVFL [11] (AUC, Time) (c, N)	IFRVFL (AUC, Time) (c, μ, N)
abalone9-18	(0.7502, 0.1185) (0.001, 10, 0.25)	(0.6964, 0.0176) (0.0001, 4)	(0.7022, 0.054) (10, 0.125)	(0.6614, 0.0015) (100, 83)	(0.7035, 0.0124) (100, 183)	(0.7683, 0.0334) (100000, 0.25, 3)
aus	(0.8482, 0.0846) (1000, 10, 4)	(0.842, 0.0138) (0.1, 32)	(0.8023, 0.0445) (0.00001, 0.125)	(0.8484, 0.0032) (0.01, 203)	(0.8492, 0.001) (0.001, 43)	(0.8526, 0.0306) (0.001, 32, 43)
checkerboard_Data	(0.8482, 0.0809) (1000, 10, 4)	(0.842, 0.0112) (0.1, 32)	(0.8023, 0.1005) (0.00001, 0.125)	(0.8484, 0.0033) (0.01, 203)	(0.8492, 0.0007) (0.001, 43)	(0.8526, 0.0314) (0.001, 32, 43)
cleve	(0.8216, 0.0198) (0.1, 10, 16)	(0.8162, 0.0022) (10, 4)	(0.7422, 0.0114) (0.01, 1)	(0.8371, 0.0024) (0.0001, 143)	(0.851, 0.0016) (0.0001, 203)	(0.7725, 0.0097) (0.0001, 16, 3)
ecoli-0-1-4-6_vs_5	(0.9321, 0.0329) (100000, 0.01, 0.5)	(0.9753, 0.0024) (10000, 0.03125)	(0.9938, 0.0083) (0.00001, 0.0625)	(0.9963, 0.0016) (100, 143)	(1, 0.0006) (1000, 63)	(0.9753, 0.0075) (10, 0.125, 63)
ecoli-0-1-4-7_vs_2-3-5-6	(0.9174, 0.0459) (100000, 10, 1)	(0.9, 0.0024) (0.01, 8)	(0.8391, 0.0467) (0.1, 0.25)	(0.83, 0.0005) (10, 43)	(0.8, 0.0036) (10, 143)	(0.8446, 0.0066) (100000, 2, 43)
ecoli-0-1-4-7_vs_5-6	(0.8642, 0.0282) (10, 0.001, 0.5)	(0.875, 0.061) (100000, 0.125)	(0.8696, 0.0113) (1, 0.0625)	(0.875, 0.001) (10, 83)	(0.875, 0.0024) (10, 163)	(0.875, 0.0107) (10, 2, 103)
ecoli-0-1_vs_5	(0.8868, 0.0286) (100, 0.01, 0.5)	(0.8259, 0.0024) (1, 1)	(0.8259, 0.009) (0.01, 0.25)	(0.8303, 0.0003) (1000, 23)	(0.8259, 0.0013) (0.1, 63)	(0.8333, 0.0042) (0.1, 8, 63)
ecoli-0-2-6-7_vs_3-5	(0.6667, 0.0268) (10000, 0.1, 8)	(0.8333, 0.0015) (1, 2)	(0.6288, 0.0051) (0.1, 0.03125)	(0.9333, 0.0016) (0.1, 83)	(1, 0.0003) (1, 23)	(0.9924, 0.0078) (100, 16, 23)
ecoli-0-3-4-6_vs_5	(0.8083, 0.0278) (10, 0.001, 1)	(0.825, 0.0015) (0.00001, 0.125)	(0.825, 0.0046) (0.00001, 0.125)	(0.795, 0.0004) (0.1, 23)	(0.8333, 0.0012) (0.01, 203)	(0.8333, 0.0038) (10, 2, 63)
ecoli-0-3-4-7_vs_5-6	(0.863, 0.0273) (1000, 10, 16)	(0.9428, 0.0019) (0.001, 32)	(0.9138, 0.0174) (0.00001, 0.03125)	(0.9228, 0.0005) (10, 43)	(0.8928, 0.0002) (0.1, 23)	(0.8138, 0.0048) (0.1, 2, 3)
ecoli-0-4-6_vs_5	(0.9286, 0.0233) (100, 0.0001, 0.5)	(0.8571, 0.0012) (100, 0.0625)	(0.8571, 0.0051) (0.00001, 0.03125)	(0.8605, 0.0011) (1000, 103)	(0.8571, 0.0004) (1000, 43)	(0.9013, 0.0063) (100, 1, 103)
ecoli-0-6-7_vs_3-5	(0.8833, 0.0238) (100, 10, 1)	(0.9208, 0.0033) (0.00001, 0.0625)	(0.9125, 0.0047) (10, 0.03125)	(0.8983, 0.0005) (100000, 43)	(0.9375, 0.0003) (10, 43)	(0.9833, 0.0072) (1000, 8, 23)
ecoli0137vs26	(0.9872, 0.0321) (100000, 1000, 4)	(0.9706, 0.0024) (10, 0.125)	(0.9321, 0.0114) (1, 0.125)	(0.9529, 0.0006) (0.01, 43)	(0.9706, 0.0015) (1, 143)	(0.9449, 0.0145) (0.01, 1, 123)
ecoli2	(0.8571, 0.0328) (0.00001, 100, 1)	(0.7218, 0.0025) (1, 4)	(0.8087, 0.0077) (100, 0.03125)	(0.6926, 0.0011) (10, 63)	(0.8072, 0.0007) (10, 63)	(0.9341, 0.0086) (100, 2, 23)
ecoli3	(0.8571, 0.0283) (0.00001, 100, 1)	(0.7218, 0.004) (1, 4)	(0.8087, 0.0347) (100, 0.03125)	(0.6926, 0.0008) (10, 63)	(0.8072, 0.0103) (10, 63)	(0.9341, 0.0092) (100, 2, 23)
heart-stat	(0.8398, 0.0238) (10, 10, 16)	(0.8717, 0.0035) (10, 16)	(0.7866, 0.0679) (0.01, 0.5)	(0.8653, 0.0019) (0.0001, 183)	(0.8574, 0.0013) (0.0001, 123)	(0.8647, 0.0065) (100000, 0.125, 123)
new-thyroid1	(0.9444, 0.0256) (0.0001, 0.0001, 4)	(0.9912, 0.0013) (0.00001, 0.25)	(0.9825, 0.0169) (10, 0.0625)	(0.9947, 0.0004) (10, 43)	(1, 0.0003) (10, 23)	(0.9825, 0.0043) (10, 0.03125, 23)
pima	(0.7582, 0.0907) (0.001, 1, 2)	(0.7613, 0.0535) (1, 16)	(0.6886, 0.0563) (1, 0.03125)	(0.73, 0.0091) (0.001, 83)	(0.7452, 0.0007) (100, 23)	(0.7827, 0.059) (1000, 0.5, 3)
shuttle-6_vs_2-3	(0.75, 0.0262) (0.0001, 0.0001, 2)	(0.9924, 0.0028) (0.00001, 0.0625)	(0.9394, 0.0072) (0.00001, 0.03125)	(0.9985, 0.0003) (10, 23)	(1, 0.0002) (0.1, 3)	(0.9924, 0.0221) (0.1, 2, 3)
shuttle-c0-vs-c4	(1, 0.6015) (0.0001, 0.00001, 16)	(0.9865, 0.2812) (0.01, 32)	(1, 0.5837) (0.00001, 2)	(0.9892, 0.0007) (0.1, 23)	(0.9865, 0.0017) (0.1, 43)	(0.9865, 0.2233) (0.0001, 0.5, 23)
sonar	(0.7877, 0.0216) (0.001, 0.01, 32)	(0.8192, 0.004) (0.00001, 8)	(0.8626, 0.0046) (0.001, 4)	(0.7591, 0.0013) (0.01, 183)	(0.736, 0.0024) (0.1, 183)	(0.7394, 0.0067) (0.1, 32, 43)
vehicle1	(0.7859, 0.1416) (0.001, 0.1, 16)	(0.7766, 0.0181) (0.001, 32)	(0.7366, 0.0782) (1, 0.5)	(0.7626, 0.0029) (100, 163)	(0.7683, 0.0011) (10, 83)	(0.7954, 0.0379) (100000, 0.03125, 203)
vehicle2	(0.9819, 0.1389) (0.001, 0.01, 32)	(0.9974, 0.0246) (0.001, 16)	(0.9738, 0.1828) (0.01, 0.25)	(0.9889, 0.0046) (10, 203)	(0.9893, 0.0066) (1, 203)	(0.9758, 0.0558) (1000, 32, 103)
vowel	(0.9482, 0.202) (0.00001, 0.001, 1)	(0.8279, 0.0365) (0.00001, 0.0625)	(0.8279, 0.0914) (0.00001, 0.0625)	(0.7582, 0.002) (100, 103)	(0.8704, 0.0023) (100, 143)	(0.8631, 0.0637) (100, 2, 163)
wpbc	(0.564, 0.0216) (0.01, 0.1, 32)	(0.4944, 0.0025) (0.00001, 8)	(0.5195, 0.0115) (0.1, 1)	(0.5811, 0.0015) (0.1, 143)	(0.6716, 0.0003) (10, 3)	(0.7356, 0.0044) (0.00001, 0.5, 203)
yeast-0-2-5-6_vs_3-7-8-9	(0.7525, 0.1946) (10, 1000, 16)	(0.7492, 0.0303) (0.1, 0.125)	(0.6332, 0.3775) (0.0001, 0.25)	(0.7122, 0.0023) (100000, 123)	(0.6924, 0.0023) (10, 143)	(0.743, 0.0897) (1, 0.03125, 143)
yeast-0-5-6-7-9_vs_4	(0.7137, 0.0664) (100, 10, 16)	(0.7137, 0.0076) (1000, 0.03125)	(0.6619, 0.024) (10, 0.125)	(0.6517, 0.0017) (10, 123)	(0.6625, 0.0027) (1, 163)	(0.7277, 0.0164) (10000, 0.25, 203)
Average Accuracy	0.8409	0.841	0.817	0.8309	0.8514	**0.8679**
Average Rank	3.3214	3.4821	4.6071	3.8036	3.0893	**2.6964**

racy equal to 86.79% and average rank equal to 2.6964. In terms of average accuracy, the proposed IFRVFL model is followed by RVFL with 85.14%, KRR with 84.1%, IFTWSVM with 84.09%, ELM with 83.09% and IFKRR with 81.7%. Since average accuracy can be a biased measure, as the outperformance in one dataset may compensate the loss over other datasets. Hence, we rank the model on each dataset to analyze the performance.

The average rank of IFTWSVM, KRR, IFKRR, ELM, RVFL and the proposed IFRVFL are $3.32,\ 3.48,\ 4.61,\ 3.8,\ 3.09$ and 2.7, respectively. After simple calculations, we get

Table 3.4: Pairwise win-tie-loss

	IFTWSVM [27]	KRR [339]	IFKRR [340]	ELM [341]	RVFL [11]
KRR [339]	[12, 1, 15]				
IFKRR [340]	[8, 1, 19]	[7, 4, 17]			
ELM [341]	[13, 0, 15]	[11, 1, 16]	[18, 0, 10]		
RVFL [11]	[14, 0, 14]	[14, 5, 9]	[20, 2, 6]	[19, 1, 8]	
IFRVFL	[17, 0, 11]	[15, 4, 9]	[23, 1, 4]	[17, 1, 10]	[16, 3, 9]

$\chi_F^2 = 17.3040, F_F = 3.8079$, for $N = 28$, k= 6. From statistical F-distribution Table, at 5% level of significance $F_F(5, 135) = 2.28$. Since $3.8079 > 2.28$, hence we refuse the null hypothesis. Thus, there is significant difference among models. To check the significant difference between different pairs of the models, we follow the Nemenyi posthoc test. For Nemenyi test, the critical difference, $CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 2.85\sqrt{\frac{6\times7}{6\times28}} = 1.425$. Two models are said to be significantly different if their average rank differs by at least CD. With the Nemenyi test, significant difference exist between (IFRVFL, IFKRR) with difference equal to 1.91 and (RVFL, IFKRR) with difference equal to 1.52. However, the Nemenyi test fails to detect the significant difference between the proposed IFRVFL model and other baseline models i.e., IFTWSVM, KRR, ELM and RVFL models. However, one can see that the average rank of the proposed IFRVFL model is better compared to the given baseline models. Furthermore, we employ the pairwise win-tie-loss sign test to analyze the models. The pairwise analysis of the models is given in Table 3.4. With $N = 28$, the two models are significantly different if either of the models win on approximately. ≥ 19.1857 datasets. From Table 3.4, it is evident that significant difference exists between IFRVFL and IFKRR models with proposed IFRVFL model being better compared to IFKRR model. Also, RVFL is statistically better compared to IFKRR model. However, win-tie-loss sign test fails to detect the significant difference between proposed IFRVFL model and other baseline models like IFTWSVM, KRR and ELM models. One can see that the proposed IFRVFL model wins on more datasets compared to IFTWSVM, KRR and ELM models. With the aforementioned analysis of the proposed IFRVFL model with different performance measures and statistical tests, it is clear that the performance of the proposed IFRVFL is competitive or better compared to the baseline models.

3.3 Class probability-based fuzzy random vector functional link (CP-FRVFL) network

In Section 3.1, we developed intuitionistic fuzzy RVFL (IFRVFL) network that uses membership and non membership value based score function to deal with noisy and outlier samples in the data. IFRVFL network uses a linear function to calculate the membership value of a sample. Under this scheme, training samples which are present on the boundary of the classes, a very low membership value (near zero) is assigned. Therefore, one may lose the important samples and hence model may have lower generalization performance. In this approach, our objectives are to design an efficient fuzzy function for classification problem and then to develop a robust Fuzzy RVFL network. As a result, we propose a class probability-based efficient score function that uses a non-linear function (smooth, bounded, and continuous function) to determine the membership value of each sample. The proposed fuzzy function uses the k-NN technique to determine the class probability value to take the neighbourhood information of each training sample. Finally, a score value is assigned to each sample based on the score function integrating the membership function and class probability. By incorporating the proposed score or fuzzy value in the objective function of the standard RVFL network, we propose a novel class probability-based fuzzy RVFL (CP-FRVFL) network.

The way a fuzzy function is defined is crucial because it decides how actively training samples will participate in the classification process?. Next, we present the proposed score function followed by the mathematical formulation of the proposed CP-FRVFL network.

3.3.1 Membership function

Designing an effective fuzzy function that assigns an appropriate membership value to each training sample to assess its contribution in calculating the classifier's parameters is necessary to minimize the impact of noise and outlier samples [344, 345]. By calculating each sample's distance from the center of its class, the membership function assigns a value to each sample in order to identify its class membership. Intuitively, samples close to the

class center should be valued higher than ones farther from the center. The membership function (M_f) is defined as follows:

$$M_f(x) = \begin{cases} sech\theta\left(\frac{\|x-x^+\|}{r^+}\right), & x \in C_+, \\ \\ sech\theta\left(\frac{\|x-x^-\|}{r^-}\right), & x \in C_-. \end{cases} \tag{3.11}$$

Here, x^+ and x^- are the center of the positive (C_+) and negative (C_-) classes, respectively. Moreover, θ is the parameter that needs to be tuned to get an appropriate membership function. Each class center is defined as follows:

$$x^+ = \frac{1}{n_1}\sum_{x_i \in C_+} x_i \text{ and } x^- = \frac{1}{n_2}\sum_{x_i \in C_-} x_i, \tag{3.12}$$

where $n_1(n_2)$ denotes the number of samples in positive (negative) class, respectively. The radius ($r^{\pm}$) of positive (negative) class is defined as follows:

$$r^{\pm} = \max_{x \in C_{\pm}} \left\| x - x^{\pm} \right\|. \tag{3.13}$$

3.3.2 Proposed score function

The data points which are on equal distance (from the class center) may have different contributions in calculating the final parameters of the classifier. Therefore, the membership value (MV) based on only distance function is not enough to determine the belongingness of a sample to its own class. While assessing if a training sample belongs to its own class, we also consider its relationship to other class in order to take additional information. Here, to describe the class probabilities of the samples to their respective classes, we employ the k-NN approach [346]. Let x_i be the feature vector of the i^{th} training sample, we selects its k-nearest neighbours $\{x_{i1}, x_{i2}, ..., x_{ik}\}$ in the original feature space. Let v_i be the number of samples of same class as of x_i in its k-nearest neighbour. Finally, the class probability of x_i

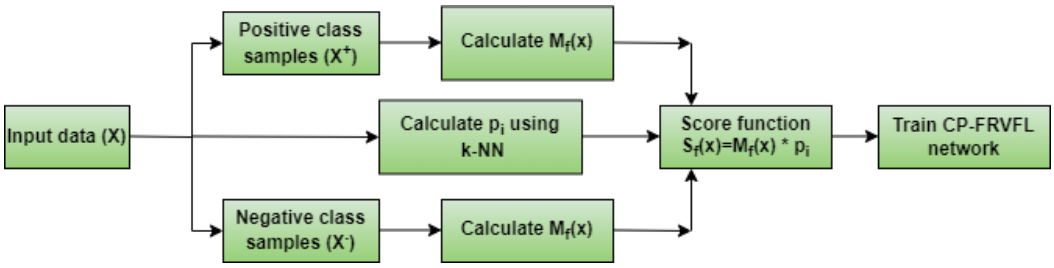


Figure 3.4: Flowchart of the proposed CP-RVFL.

is determined as follow:

$$p_i = \frac{v_i}{k}, \quad i = 1, 2, ..., N. \tag{3.14}$$

The higher probability (p_i) of x_i indicates that it has more surrounding samples of its own class and therefore, less possibility of it being a noise or outlier sample and vice versa. Finally, we define a score function (S) that integrates the membership function and class probability as follows:

$$S_f(x_i) = M_f(x_i) \times p_i, \quad i = 1, 2, ..., N. \tag{3.15}$$

The proposed score function S_f takes into account both a sample's distance from the corresponding class center and its surrounding information. The importance of a point x in the training process increases/decreases as the value of S_f increases/decreases. Next, we present the CP-FRVFL's mathematical formulation.

3.3.3 The proposed CP-FRVFL network

By incorporating the membership function and class probability-based score value into the objective function of the RVFL network, we propose CP-FRVFL network. In contrast to the standard RVFL network, the proposed CP-FRVFL network gives specific consideration and weights to each sample to determine their role in calculating the network's parameters. Figure 3.4 shows the flowchart of the proposed model. The proposed optimization problem

with equality constraints is formulated as follows:

$$\min_{\beta \in \mathbb{R}^{(d+L)\times m}} \frac{1}{2}\|\beta\|_2^2 + \frac{1}{2}\lambda \|D^{1/2}\zeta\|_2^2$$
$$\text{subject to } G\beta - Y = \zeta, \tag{3.16}$$

where G is the feature matrix containing original features and randomized features calculated via hidden layer. β is the output weights matrix and $D = diag(s_1, s_2, \cdots, s_N)$ is the diagonal matrix which contains the score values calculated via (3.15) and $\zeta = [\zeta_1, \zeta_2, \cdots, \zeta_N]^T$ is the weights matrix and error term corresponding to N samples. The first term in the proposed objective function corresponds to the regularization term that prevents the CP-FRVFL network from over-fitting; the second term is the weighted error and λ is the parameter that controls the trade-off between the error and regularization term. The optimization problem (3.16) is a convex quadratic problem and hence, it has a unique solution. The Lagrange function of the problem (3.16) is defined as follows:

$$L(\beta, \zeta, \alpha) = \frac{1}{2}\lambda \left\|D^{1/2}\zeta\right\|^2 + \frac{1}{2}\|\beta\|^2 - \alpha^T(G\beta - Y - \zeta), \tag{3.17}$$

and obtain the partial derivatives of L w.r.t β, ζ and α and set them equal to zero.

$$\frac{\partial L}{\partial \beta} = 0 \implies \beta = G^T\alpha, \tag{3.18}$$
$$\frac{\partial L}{\partial \zeta} = 0 \implies \alpha = -\lambda D\zeta, \tag{3.19}$$
$$\frac{\partial L}{\partial \alpha} = 0 \implies G\beta - Y - \zeta = 0. \tag{3.20}$$

From (3.19) and (3.20), we obtain $\alpha = -\lambda D(G\beta - Y)$. By substituting the value of α in (3.18), we get:

$$\beta = (G^T DG + \frac{1}{\lambda}I)^{-1}G^T DY, \tag{3.21}$$

After substituting (3.18) and (3.19) into (3.20), we obtain $\alpha = (DGG^T + \frac{1}{\lambda}I)^{-1}DY$, By substituting the value of α in (3.18),

$$\beta = G^T(DGG^T + \frac{1}{\lambda}I)^{-1}DY. \tag{3.22}$$

Therefore, in this case the optimal solution of (3.16) is given by,

$$\beta = \begin{cases} (G^T DG + \frac{1}{\lambda}I)^{-1}G^T DY, & (d+L) \leq N, \\ G^T(DGG^T + \frac{1}{\lambda}I)^{-1}DY, & N < (d+L), \end{cases} \tag{3.23}$$

where λ is the regularization parameter to be tuned and I is an identity matrix of appropriate dimension. Let x be the unseen sample and O^i be the output of the final layer's i^{th} node, $i = 1, 2, ..., m$, then the final class is assigned to x by the decision function is as follows:

$$\text{class\_label (x)} = \underset{i=1,2,...,m}{\text{argmax}} \{O^i\}. \tag{3.24}$$

3.3.4 Computation complexity analysis

In CP-FRVFL, matrix inverses are computed to calculate the output layer weights. Hence, the complexity of the models is determined by the size of the matrices to be inverted. By the standard procedure, $O(N^2)$ memory and $O(N^3)$ time are required to calculate the inverse a matrix of size $N \times N$ [337]. The CP-FRVFL model calculates the output layer weights either in the primal or dual, which results in reducing the complexity of the model by choosing the $min(N, d+L)$. Thus, the time complexity of the CP-FRVFL model is approximately either $O(N^3)$ or $O((d+L)^3)$. Moreover, ELM has a time complexity of either $O(N^3)$ or $O(L^3)$.

3.3.5 Experiments

In this section, we discuss the experimental results among the baseline models and the proposed CP-FRVFL model on UCI and KEEL datasets.

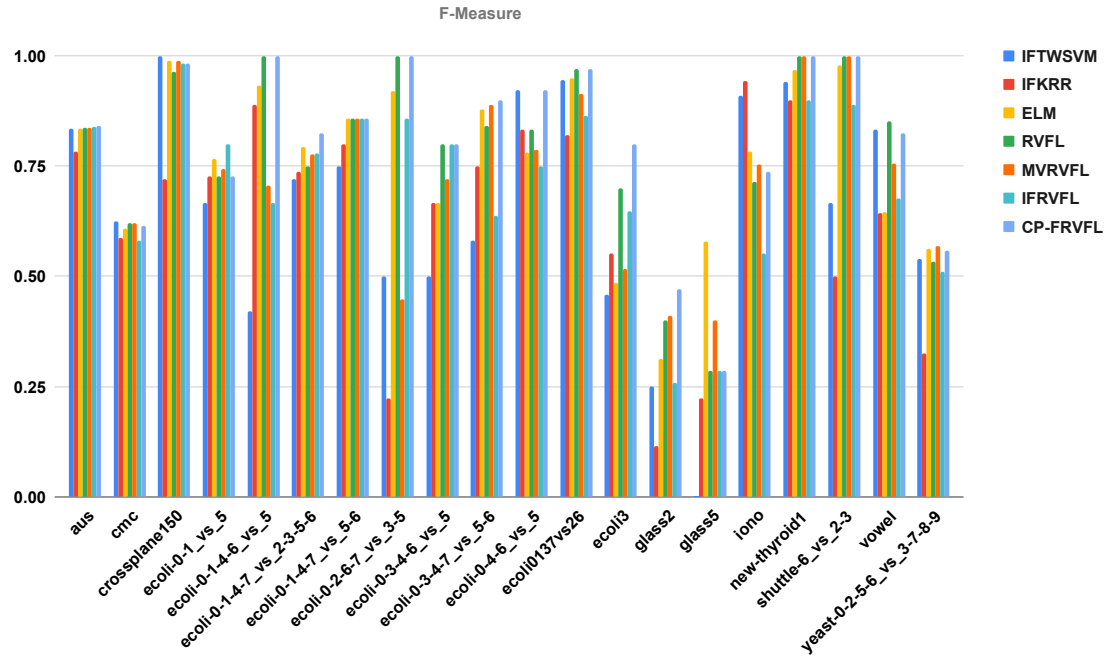


Figure 3.5: F-measure plot of the models on KEEL datasets

3.3.5.1 Experimental setup

All the experiments are performed with MATLAB R2017b and all the tuning parameters related to IFTWSVM, IFKRR, ELM, RVFL, MVRVFL and IFRVFL and the proposed CP-FRVFL are the same as described in Subsection (4.2.1). For the proposed CP-FRVFL, the θ parameter is tuned from the range $\{.3, .7, .9, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3\}$. The experiments are conducted over UCI and KEEL datasets [342, 343]. The dataset is randomly partitioned into $30 : 70$ ratio of testing and training sets, respectively. The hyperparameters corresponding to different models are optimized using grid search approach via 5-fold cross validation. The performance of each model corresponding to the best hyperparameters is given as the final accuracy of a model.

3.3.5.2 Experimental results over KEEL and UCI datasets

The experimental results corresponding to the proposed CP-FRVFL and the baseline models are shown in Table 3.5. One can see that the proposed CP-FRVFL model has the highest average AUC equal to 88.18%, followed by IFRVFL with average AUC equal to

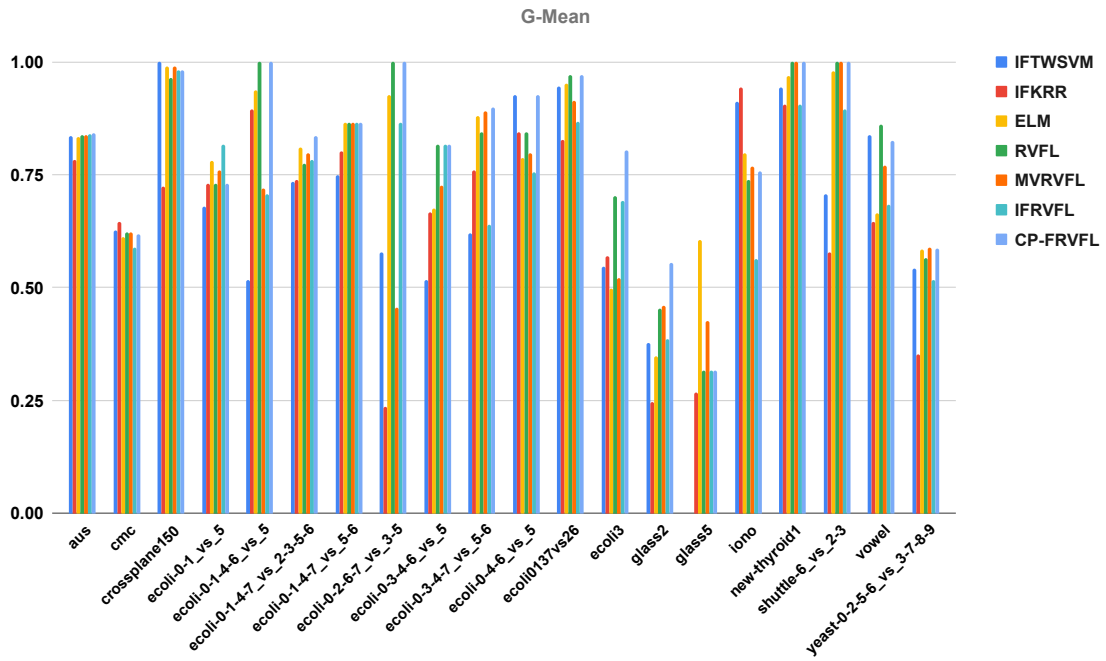


Figure 3.6: G-mean plot of the models on KEEL datasets

86.239%. Moreover, IFTWSVM, IFKRR, ELM, RVFL and MVRVFL have average AUC equal to 84.316%, 79.636%, 84.755%, 86.17% and 86.239%, respectively. The results shows that the proposed CP-FRVFL model is almost 2% superior to RVFL and IFRVFL models and 4% superior to IFTWSVM model. Sometimes average accuracy might be skewed because performance in one dataset can sometimes make up for performance in another. To evaluate the performance, we rank the model on each dataset. In this case, each model is ranked on the datasets, with the lower performing model receiving a higher rank and the higher performing model receiving a lower rank. Therefore, the most successful model has a lower rank. To validate the performance of the models statistically, we employ the Friedman test. After some calculation, $\chi_F^2 = 12.86$ and $F_F = 2.2806$. Here, we have $N = 20$ number of datasets and $k = 7$ compared models. Therefore, from the statistical F-distribution table, $F(6, 114) = 2.18$. Since, $2.2806 > 2.18$, hence, the null hypothesis is rejected. Thus, there is a significant difference among the compared models. To evaluate the classification performance of the proposed CP-FRVFL and RdNN models further, we also conduct the experiments on UCI datasets and the corresponding

Table 3.5: The experimental results of the proposed CP-FRVFL and the compared models on KEEL datasets.

Dataset Name	IFTWSVM [27] (AUC,Sen.) (Spec.,Prec.)	IFKRR [340] (AUC,Sen.) (Spec.,Prec.)	ELM [341] (AUC,Sen.) (Spec.,Prec.)	RVFL [11] (AUC,Sen.) (Spec.,Prec.)	MVRVFL[145] (AUC,Sen.) (Spec.,Prec.)	IFRVFL [76] (AUC,Sen.) (Spec.,Prec.)	CP-FRVFL (AUC,Sen.) (Spec.,Prec.)
aus	(0.8482, 0.8617)	(0.8023, 0.7872)	(0.8484, 0.8447)	(0.8492, 0.8723)	(0.8518, 0.8532)	(0.8526, 0.8617)	(0.8536, 0.8723)
	(0.8348, 0.81)	(0.8174, 0.7789)	(0.8522, 0.8238)	(0.8261, 0.8039)	(0.8504, 0.8236)	(0.8435, 0.8182)	(0.8348, 0.8119)
cmc	(0.6881, 0.5924)	(0.5, 1)	(0.6791, 0.5652)	(0.6877, 0.5761)	(0.6892, 0.5761)	(0.6708, 0.5)	(0.6884, 0.5543)
	(0.7838, 0.6606)	(0, 0.4153)	(0.7931, 0.66)	(0.7992, 0.6709)	(0.8023, 0.675)	(0.8417, 0.6917)	(0.8224, 0.6892)
crossplane150	(1, 1)	(0.5873, 0.7857)	(0.9893, 0.9786)	(0.9643, 0.9286)	(0.9893, 0.9786)	(0.9821, 0.9643)	(0.9821, 0.9643)
	(1, 1)	(0.3889, 0.6667)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)
ecoli-0-1_vs_5	(0.8868, 0.8333)	(0.8259, 0.6667)	(0.8303, 0.6667)	(0.8259, 0.6667)	(0.8137, 0.6333)	(0.8333, 0.6667)	(0.8259, 0.6667)
	(0.9403, 0.5556)	(0.9851, 0.8)	(0.994, 0.9267)	(0.9851, 0.8)	(0.994, 0.92)	(1, 1)	(0.9851, 0.8)
ecoli-0-1-4-6_vs_5	(0.9321, 1)	(0.9938, 1)	(0.9963, 1)	(1, 1)	(0.9102, 0.85)	(0.9753, 1)	(1, 1)
	(0.8642, 0.2667)	(0.9877, 0.8)	(0.9926, 0.88)	(1, 1)	(0.9704, 0.6222)	(0.9506, 0.5)	(1, 1)
ecoli-0-1-4-7_vs_2-3-5-6	(0.9174, 0.9)	(0.8391, 0.7)	(0.83, 0.66)	(0.8, 0.6)	(0.82, 0.64)	(0.8446, 0.7)	(0.85, 0.7)
	(0.9348, 0.6)	(0.9783, 0.7778)	(1, 1)	(1, 1)	(1, 1)	(0.9891, 0.875)	(1, 1)
ecoli-0-1-4-7_vs_5-6	(0.8642, 0.75)	(0.8696, 0.75)	(0.875, 0.75)	(0.875, 0.75)	(0.875, 0.75)	(0.875, 0.75)	(0.875, 0.75)
	(0.9785, 0.75)	(0.9892, 0.8571)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)
ecoli-0-2-6-7_vs_3-5	(0.6667, 0.3333)	(0.6288, 0.3333)	(0.9333, 0.8667)	(1, 1)	(0.7455, 0.5333)	(0.9924, 1)	(1, 1)
	(1, 1)	(0.9242, 0.1667)	(1, 1)	(1, 1)	(0.9576, 0.3933)	(0.9848, 0.75)	(1, 1)
ecoli-0-3-4-6_vs_5	(0.8083, 0.6667)	(0.825, 0.6667)	(0.795, 0.6)	(0.8333, 0.6667)	(0.8283, 0.6667)	(0.8333, 0.6667)	(0.8333, 0.6667)
	(0.95, 0.4)	(0.9833, 0.6667)	(0.99, 0.7667)	(1, 1)	(0.99, 0.8)	(1, 1)	(1, 1)
ecoli-0-3-4-7_vs_5-6	(0.863, 0.9)	(0.9138, 0.9)	(0.9228, 0.86)	(0.8928, 0.8)	(0.9328, 0.88)	(0.8138, 0.7)	(0.9428, 0.9)
	(0.8261, 0.4286)	(0.9275, 0.6429)	(0.9855, 0.9014)	(0.9855, 0.8889)	(0.9855, 0.9036)	(0.9275, 0.5833)	(0.9855, 0.9)
ecoli-0-4-6_vs_5	(0.9286, 0.8571)	(0.8571, 0.7143)	(0.8605, 0.7429)	(0.8571, 0.7143)	(0.8392, 0.6857)	(0.9013, 0.8571)	(0.9286, 0.8571)
	(1, 1)	(1, 1)	(0.9782, 0.8429)	(1, 1)	(0.9927, 0.9333)	(0.9455, 0.6667)	(1, 1)
ecoli0137vs26	(0.9872, 1)	(0.9321, 0.9412)	(0.9529, 0.9059)	(0.9706, 0.9412)	(0.9532, 0.9294)	(0.9449, 0.9412)	(0.9706, 0.9412)
	(0.9744, 0.8947)	(0.9231, 0.7273)	(1, 1)	(1, 1)	(0.9769, 0.8991)	(0.9487, 0.8)	(1, 1)
ecoli3	(0.8571, 1)	(0.8087, 0.7273)	(0.6926, 0.4182)	(0.8072, 0.6364)	(0.7213, 0.4909)	(0.9341, 1)	(0.8581, 0.7273)
	(0.7143, 0.2973)	(0.8901, 0.4444)	(0.967, 0.6014)	(0.978, 0.7778)	(0.9516, 0.5609)	(0.8681, 0.4783)	(0.989, 0.8889)
glass2	(0.8065, 1)	(0.5, 1)	(0.7121, 0.55)	(0.8105, 0.75)	(0.8137, 0.75)	(0.8145, 1)	(0.9274, 1)
	(0.6129, 0.1429)	(0, 0.0606)	(0.8742, 0.22)	(0.871, 0.2727)	(0.8774, 0.2818)	(0.629, 0.1481)	(0.8548, 0.3077)
glass5	(0.4922, 0)	(0.7031, 0.5)	(0.7453, 0.5)	(0.7188, 0.5)	(0.775, 0.6)	(0.7188, 0.5)	(0.7188, 0.5)
	(0.9844, 0)	(0.9063, 0.1429)	(0.9906, 0.7667)	(0.9375, 0.2)	(0.95, 0.3067)	(0.9375, 0.2)	(0.9375, 0.2)
iono	(0.9216, 0.8571)	(0.9575, 0.9429)	(0.8245, 0.6629)	(0.7788, 0.5714)	(0.8075, 0.64)	(0.68, 0.4571)	(0.7931, 0.6)
	(0.9861, 0.9677)	(0.9722, 0.9429)	(0.9861, 0.9592)	(0.9861, 0.9524)	(0.975, 0.9252)	(0.9028, 0.6957)	(0.9861, 0.9545)
new-thyroid1	(0.9444, 0.8889)	(0.9825, 1)	(0.9947, 1)	(1, 1)	(1, 1)	(0.9825, 1)	(1, 1)
	(1, 1)	(0.9649, 0.8182)	(0.9895, 0.94)	(1, 1)	(1, 1)	(0.9649, 0.8182)	(1, 1)
shuttle-6_vs_2-3	(0.75, 0.5)	(0.9394, 1)	(0.9985, 1)	(1, 1)	(1, 1)	(0.9924, 1)	(1, 1)
	(1, 1)	(0.8788, 0.3333)	(0.997, 0.96)	(1, 1)	(1, 1)	(0.9848, 0.8)	(1, 1)
vowel	(0.9482, 0.9259)	(0.8279, 0.7037)	(0.7582, 0.5259)	(0.8704, 0.7407)	(0.8193, 0.6444)	(0.8631, 0.7778)	(0.8834, 0.7778)
	(0.9705, 0.7576)	(0.952, 0.5938)	(0.9904, 0.8445)	(1, 1)	(0.9941, 0.9289)	(0.9483, 0.6)	(0.9889, 0.875)
yeast-0-2-5-6_vs_3-7-8-9	(0.7525, 0.6)	(0.6332, 0.525)	(0.7122, 0.445)	(0.6924, 0.4)	(0.7169, 0.455)	(0.743, 0.6)	(0.7049, 0.425)
	(0.9049, 0.4898)	(0.7414, 0.236)	(0.9795, 0.7697)	(0.9848, 0.8)	(0.9787, 0.7666)	(0.8859, 0.4444)	(0.9848, 0.8095)
Average AUC	0.84316	0.79636	0.84755	0.8617	0.8451	0.86239	**0.8818**
Average rank	3.925	5.5	4.325	3.95	3.925	3.85	**2.525**

Table 3.6: The experimental results of the proposed CP-FRVFL and the compared models on UCI datasets.

Dataset Name	ELM [341] (AUC,Sen.) (Spec.,Prec.)	RVFL [11] (AUC,Sen.) (Spec.,Prec.)	MVRVFL[145] (AUC,Sen.) (Spec.,Prec.)	IFRVFL [76] (AUC,Sen.) (Spec.,Prec.)	CP-FRVFL (AUC,Sen.) (Spec.,Prec.)
blood	$(0.6502, 0.3613)$	$(0.648, 0.3387)$	$(0.6641, 0.371)$	$(0.6938, 0.5645)$	$(0.6661, 0.3871)$
	$(0.939, 0.6941)$	$(0.9573, 0.75)$	$(0.9573, 0.7667)$	$(0.8232, 0.5469)$	$(0.9451, 0.7273)$
breast-cancer-wisc	$(0.9568, 0.9436)$	$(0.9786, 0.9872)$	$(0.9363, 0.9103)$	$(0.9812, 1)$	$(0.9812, 1)$
	$(0.9699, 0.9484)$	$(0.9699, 0.9506)$	$(0.9624, 0.9342)$	$(0.9624, 0.9398)$	$(0.9624, 0.9398)$
breast-cancer-wisc-prog	$(0.5274, 0.2923)$	$(0.4663, 0.3077)$	$(0.5673, 0.3846)$	$(0.4447, 0.0769)$	$(0.4663, 0.3077)$
	$(0.7625, 0.2454)$	$(0.625, 0.1818)$	$(0.75, 0.2941)$	$(0.8125, 0.1)$	$(0.625, 0.1818)$
chess-krvkp	$(0.9512, 0.942)$	$(0.9602, 0.951)$	$(0.9624, 0.9595)$	$(0.953, 0.9488)$	$(0.9538, 0.9382)$
	$(0.9605, 0.958)$	$(0.9695, 0.9675)$	$(0.9654, 0.9636)$	$(0.9572, 0.9549)$	$(0.9695, 0.967)$
heart-hungarian	$(0.8381, 0.7625)$	$(0.8335, 0.7188)$	$(0.7231, 0.6875)$	$(0.8303, 0.7813)$	$(0.8335, 0.7188)$
	$(0.9138, 0.831)$	$(0.9483, 0.8846)$	$(0.7586, 0.6111)$	$(0.8793, 0.7813)$	$(0.9483, 0.8846)$
ilpd-indian-liver	$(0.6049, 0.3617)$	$(0.5869, 0.2979)$	$(0.6178, 0.383)$	$(0.5924, 0.5957)$	$(0.5976, 0.3191)$
	$(0.8481, 0.4642)$	$(0.876, 0.4667)$	$(0.8527, 0.4865)$	$(0.5891, 0.3457)$	$(0.876, 0.4839)$
mushroom	$(0.9999, 0.9998)$	$(1, 1)$	$(1, 1)$	$(0.9971, 0.9941)$	$(1, 1)$
	$(1, 1)$	$(1, 1)$	$(1, 1)$	$(1, 1)$	$(1, 1)$
parkinsons	$(0.8422, 0.7467)$	$(0.8556, 0.8)$	$(0.8778, 0.8)$	$(0.8556, 0.8)$	$(0.8667, 0.8)$
	$(0.9378, 0.8037)$	$(0.9111, 0.75)$	$(0.9556, 0.8571)$	$(0.9111, 0.75)$	$(0.9333, 0.8)$
spambase	$(0.9066, 0.8708)$	$(0.9191, 0.8904)$	$(0.9179, 0.8904)$	$(0.915, 0.8904)$	$(0.9185, 0.8904)$
	$(0.9425, 0.9013)$	$(0.9478, 0.9114)$	$(0.9455, 0.9078)$	$(0.9397, 0.899)$	$(0.9466, 0.9096)$
statlog-heart	$(0.847, 0.8722)$	$(0.8684, 0.8889)$	$(0.7198, 0.7222)$	$(0.8684, 0.8889)$	$(0.8684, 0.8889)$
	$(0.8217, 0.7932)$	$(0.8478, 0.8205)$	$(0.7174, 0.6667)$	$(0.8478, 0.8205)$	$(0.8478, 0.8205)$
tic-tac-toe	$(0.9789, 0.9579)$	$(0.9684, 0.9368)$	$(0.9737, 0.9474)$	$(0.9684, 0.9368)$	$(0.9737, 0.9474)$
	$(1, 1)$	$(1, 1)$	$(1, 1)$	$(1, 1)$	$(1, 1)$
titanic	$(0.7016, 0.5019)$	$(0.7094, 0.5189)$	$(0.6874, 0.4104)$	$(0.7091, 0.5472)$	$(0.7094, 0.5189)$
	$(0.9013, 0.7055)$	$(0.9, 0.7097)$	$(0.9644, 0.8447)$	$(0.8711, 0.6667)$	$(0.9, 0.7097)$
twonorm	$(0.9735, 0.9677)$	$(0.9757, 0.9686)$	$(0.9703, 0.9633)$	$(0.9757, 0.9686)$	$(0.9753, 0.9704)$
	$(0.9792, 0.9792)$	$(0.9828, 0.9827)$	$(0.9774, 0.9773)$	$(0.9828, 0.9827)$	$(0.9801, 0.9801)$
Average AUC	0.8291	0.82847	0.81676	0.82959	**0.83158**
Average rank	3.46	2.85	3.04	3.31	**2.35**

results are shown in Table 3.6. The proposed CP-FRVFL model has winning performance with average AUC equal to 83.158% and IFRVFL has second position with average AUC equal to 82.959%. Moreover, we ranked the models on each dataset and hence, the proposed CP-FRVFL model has smallest average rank, i.e., 2.35. The existing models including ELM, RVFL, MVRVFL and IFRVFL has average rank equal to 3.46, 2.85, 3.04 and 3.31, respectively. Figure 3.5 and 3.6 show the F-measure and G-mean plots of the models on KEEL datasets, respectively. The significance of F-measure lies in its ability to effectively capture both precision and recall within a single metric. The Figure 3.5 show that the proposed CP-FRVFL model has winning performance on most of the datatsets namely, aus, ecoli-0-1-4-6_vs_5, ecoli-0-2-6-7_vs_3-5, ecoli0137vs26, ecoli3, glass2, new-thyroid1, shuttle-6_vs_2-3 and vowel. Another metric is G-mean which gives complete information of both classes and Figure 3.6 shows that the proposed CP-FRVFL model has highest G-mean over several datasets namely, aus, ecoli-0-1-4-6_vs_5, ecoli-0-1-4-7_vs_2-3-5-6, ecoli-0-1-4-7_vs_5-6, ecoli-0-2-6-7_vs_3-5, ecoli-0-3-4-6_vs_5, ecoli-0-3-4-7_vs_5-6, ecoli-0-4-6_vs_5, ecoli0137vs26, ecoli3, glass2, new-thyroid1, and shuttle-6_vs_2-3.

3.4 Graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) network

RVFL assumes that all the samples are equally important, however, this may not be true in the real-world datasets. Moreover, RVFL ignores the geometric and discriminative information of the data. To overcome these shortcomings simultaneously, we propose a graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) network. The proposed GE-IFWRVFL model assigns weights to each data point based on the membership and non membership functions similar to Section 3.1. Also, the proposed GE-IFWRVFL employ the geometric relationship of the data under the GE framework (see Chapter 2) and enhances the generalization performance. A novel graph regularization term is incorporated in the proposed GE-IFWRVFL model to handle the topological structure of the data. Similar to IFRVFL model, the proposed GE-IFWRVFL optimizes the output layer weights assigning

appropriate weights to each sample.

3.4.1 Formulation of the proposed GE-IFRVFL network

The proposed GE-IFRVFL model minimizes the weighted error term, output layer weights norm and a novel regularization term under GE framework that handles the geometrical relationship of the data. The optimization problem of the proposed GE-IFWRVFL model can be formulated as:

$$\begin{aligned} &\min \frac{1}{2}\lambda\|S^{\frac{1}{2}}\zeta\|_2^2 + \frac{1}{2}\|\beta\|_2^2 + \frac{1}{2}\alpha\|S_w^{\frac{1}{2}}\beta\|_2^2 \\ &\text{subject to } B\beta - Y = \zeta, \end{aligned} \tag{3.25}$$

where, both λ and α are tunable parameters. And, $S = diag(s_1, s_2, \cdots, s_N)$ is a diagonal matrix, wherein s_i is the score value corresponding to each sample (similar to Section 3.1) and ζ is the error matrix. Here, $B = [X\ H]$ is the concatenated matrix that contains original features and randomized features. Eq. 3.25 give special importance to each sample by giving weights and consider the geometrical relationship of samples by incorporating GE term ($\|S_w^{\frac{1}{2}}\beta\|_2^2$). Moreover, both intrinsic graph and penalty graph are defined over the concatenated matrix B, i.e. $G^{int} = \{B, \Omega^{int}\}$ and $G^{pen} = \{B, \Omega^{pen}\}$. So, $\mathbf{S}_i = B^T L B$ (see Chapter 2) and $S_p = B^T V B$. In the literature, for the linear discriminant analysis (LDA) [347], the graph weights for intrinsic and penalty graph are defined as:

$$\Omega_{ij} = \begin{cases} \frac{1}{N_{c_i}}, & c_j = c_i, \\ 0, & \text{otherwise}, \end{cases} \tag{3.26}$$

$$\Omega_{ij}^{p} = \begin{cases} \frac{1}{N} - \frac{1}{N_{c_i}}, & c_j = c_i, \\ \frac{1}{N}, & \text{otherwise}, \end{cases} \tag{3.27}$$

respectively. Here, c_i represents the i^{th} class and N_{c_i} represents the number of sample

in class (c_i) and finally, $S_w = S_p^{-1} S_i$. The Lagrangian function of (3.25) is as follows:

$$L = \frac{1}{2}\lambda \| S^{\frac{1}{2}}(B\beta - Y)\|_2^2 + \frac{1}{2}\,\|\beta\|_2^2 + \frac{1}{2}\alpha \| S_w^{\frac{1}{2}}\beta\|_2^2. \tag{3.28}$$

Taking the partial derivative of (3.28) with respect to β and using the karush-kuhn-Tucker (K.K.T.) conditions, we get,

$$\frac{\partial L}{\partial \beta} = \lambda B^t S(B\beta - T) + \beta + \alpha S_w \beta = 0. \tag{3.29}$$

After calculations, the obtained output parameter is given as:

$$\beta = (B^t S B + \frac{1}{\lambda} I + \frac{\alpha}{\lambda} S_w)^{-1} B^t S Y. \tag{3.30}$$

3.4.2 Computational complexity analysis

Let (X, Y) be the training set with $X \in \mathbb{R}^{N \times d}$ and $Y \in \mathbb{R}^{N \times c}$. Here, N and d are the number of samples and number of features respectively. In the proposed model, we need to calculate the matrices (S_w), S, and the inverse of a square matrix of order $(d + L)$. So the computational cost of GE-IFWRVFL is approximately $O(d + L)^3 + O(N(d + L)^3) + O(N^2(d + L))$.

3.4.3 Experimental results

This section discusses the experimental results of the proposed GE-IFWRVFL model and the baseline models.

3.4.3.1 Experimental setup

In this research, the simulations are performed on Windows-10 operating system, $8-$GB RAM, Intel(R) Xeon(R) CPU E5$-$2697 v4 2.30 GHz and MATLAB R2017b. We used Gaussian kernel, $K(x, y)$ = exp$(-(\|x - y\|^2)/\mu^2)$ to calculate the weights in the kernel space. Here, μ represents the kernel parameter. The experiments are conducted over UCI and KEEL datasets [342, 343]. The dataset is randomly divided into $70 : 30$ ratio for

Table 3.7: Experimental results of the baseline models, i.e. ELM, IFTWSVM, KRR, RVFL, IFKRR, IFRVFL, MVRVFL and the proposed GE-IFWRVFL model.

Dataset	IFTWSVM [348]	KRR [339]	IFKRR [340]	ELM [341]	RVFL [110]	IFRVFL [76]	MVRVFL [145]	GE-IFWRVFL
abalone9-18	75.02	69.64	70.22	66.14	70.35	76.83	64.78	82
brwisconsin	98.76	99.38	50	99.5	99.07	99.38	99.07	98.45
bupa-or-liver-disorders	69.65	67.89	59.45	71.89	69.89	64.96	70.51	70.67
checkerboard_Data	84.82	84.2	80.23	84.84	84.92	85.26	85.18	84.49
cmc	68.81	69.69	50	67.91	68.77	67.08	68.92	68.92
crossplane130	100	100	51.33	100	100	100	99.6	100
crossplane150	100	96.43	58.73	98.93	96.43	98.21	98.93	98.21
ecoli-0-1_vs_5	88.68	82.59	82.59	83.03	82.59	83.33	81.37	83.33
ecoli-0-1-4-6_vs_5	93.21	97.53	99.38	99.63	100	97.53	91.02	100
ecoli-0-1-4-7_vs_5-6	86.42	87.5	86.96	87.5	87.5	87.5	87.5	87.5
ecoli-0-2-3-4_vs_5	92.11	97.37	98.25	99.65	89.12	92.11	96.77	99.12
ecoli-0-2-6-7_vs_3-5	66.67	83.33	62.88	93.33	100	99.24	74.55	100
ecoli-0-3-4-6_vs_5	80.83	82.5	82.5	79.5	83.33	83.33	82.83	82.5
ecoli-0-6-7_vs_3-5	88.33	92.08	91.25	89.83	93.75	98.33	78.08	93.75
ecoli-0-6-7_vs_5	73.36	74.18	71.72	81.69	75	79.23	82.51	80.87
ecoli0137vs26	98.72	97.06	93.21	95.29	97.06	94.49	95.32	97.06
ecoli2	85.71	72.18	80.87	69.26	80.72	93.41	72.13	92.16
ecoli3	85.71	72.18	80.87	69.26	80.72	93.41	72.13	92.16
glass5	49.22	72.66	70.31	74.53	71.88	71.88	77.5	99.22
heart-stat	83.98	87.17	78.66	86.53	85.74	86.47	87.53	86.11
iono	92.16	96.53	95.75	82.45	77.88	68	80.75	88.69
led7digit-0-2-4-5-6-7-8-9_vs_1	90.16	90.97	50	90.73	94.19	92.58	90.08	92.58
new-thyroid1	94.44	99.12	98.25	99.47	100	98.25	100	99.12
pima	75.82	76.13	68.86	73	74.52	78.27	73.41	79.85
ripley	92.03	92.04	50	91.5	91.77	91.33	90.77	92.03
Average Accuracy	84.58	85.61	74.49	85.42	86.21	87.22	84.05	**89.95**
Average Rank	4.94	4.24	6.5	4.38	4.12	3.94	4.86	**3.02**

Table 3.8: Significance difference among the models based on the Nemenyi test.

	IFTWSVM [348]	KRR [339]	IFKRR [340]	ELM [341]	RVFL [110]	IFRVFL [76]	MVRVFL [145]
GE-IFWRVFL			✓				

Here, ✓represents the significant difference exists between the row and the column method. Empty entries represent that there is no significance difference between the row and the column method.

training and testing sets, respectively. Grid search is applied for the optimization of hyper-parameters of different models with 5 fold cross validation scheme. The adaptable parameters, i.e. λ and α corresponding to the models are taken from the following range: $\{10^{-5}, 10^{-4}, \ldots, 10^{4}, 10^{5}\}$. The number of hidden layer neurons h_L is selected from the range $3:20:203$ and relu activation function is employed.

3.4.3.2 Experiments analysis

The experimental results corresponding to the proposed GE-IFWRVFL and the baseline models are shown in the Table 3.7. The results show that the proposed GE-IFWRVFL model has better generalization performance compared to the baseline models in term of average accuracy and average rank. The baseline models MVRVFL, KRR, IFTWSVM, IFKRR, ELM, RVFL, IFRVFL, and the proposed GE-IFWRVFL models have the average accuracy 84.05%, 85.61%, 84.58%, 74.49%, 85.42%, 86.21%, 87.22% and 89.95%, respectively.

Table 3.9: Win-tie-loss count (pairwise).

	IFTWSVM [348]	KRR [339]	IFKRR [340]	ELM [341]	RVFL [110]	IFRVFL[76]	MVRVFL[145]
KRR [339]	[16, 1, 8]						
IFKRR [340]	[8, 0, 17]	[5, 2, 18]					
ELM [341]	[13, 1, 11]	[11, 2, 12]	[19, 0, 6]				
RVFL [110]	[13, 1, 11]	[12, 5, 8]	[20, 1, 4]	[14, 2, 9]			
IFRVFL [76]	[16, 2, 7]	[12, 4, 9]	[21, 1, 3]	[10, 2, 13]	[12, 4, 9]		
MVRVFL [145]	[12, 0, 13]	[8, 1, 16]	[17, 0, 8]	[11, 2, 12]	[9, 3, 13]	[10, 1, 14]	
GE-IFWRVFL	[17, 2, 6]	[15, 5, 5]	[23, 1, 1]	[15, 2, 8]	[14, 6, 5]	[13, 5, 7]	[16, 2, 7]

Here, win-tie-loss [x,y,z] represents that the row method has x-times wins, y-times ties, and z-times loses corresponding to the column method.

Statistical tests are conducted to evaluate the performance of the models from statistical perspective. In Friedman test [269], algorithms are assigned rank based on their classification accuracy on each dataset. The average rank of the baseline models IFTWSVM, KRR, IFKRR, ELM, RVFL, IFRVFL, MVRVFL and the proposed GE-IFWRVFL model are 4.94, 4.24, 6.5, 4.38, 4.12, 3.94, 4.86 and 3.02, respectively, are given in Table 3.7. One can see that the proposed GE-IFWRVFL model has the lowest rank among all models. After performing simple calculations, χ_F^2 is 29.39 and F_F is 4.8442 for $k = 8$ models with $N = 25$ datasets. The critical value at $F_F(7, 168)$ is 2.065 from the F distribution table at $\alpha = 0.05$ significant level. Here, the value $F_F = 4.8442 > 2.065$. Therefore, we reject the null hypothesis and it indicates that there are statistical differences among the compared models. In addition, for pairwise comparison between the models, the Nemenyi post hoc [269] test is performed. If the average rank of the models differs by at least critical difference (CD), then the models are said to be significantly different. After calculation, we get CD=2.1. Table 3.8 shows that there are significant differences exist between IFKRR and the proposed GE-IFWRVFL method. Moreover, we also examine the win-tie-loss (pairwise) sign test for the statistical analysis of the models. The two models' performances are said to be statistically different, if the number of total wins of a model is at least equal to 17.4. Therefore, significant difference exists between two models if win $>= 17.4$. Table 3.9 shows the total count of pairwise win-tie-loss of baseline models as well as the proposed GE-IFWRVFL model. It can be observed that the proposed GE-IFWRVFL model has wins between 13 to 23 out of 25 datasets with respect to baseline models. It shows the superiority of the proposed GE-IFWRVFL model over the baseline models. The proposed GE-IFWRVFL model is significantly different from IFTWSVM and IFKRR model based on the sign test. The

results indicate that the graph regularization term improves the generalization performance of IFRVFL model. The overall results indicate that the incorporation of the intuitionistic fuzzy theory and graph embedding into the proposed GE-IFWRVFL model results in better generalization performance from the baseline models.

3.5 Experimental results analysis of all proposed models on ADNI dataset

We have conducted the experiments of the proposed IFRVFL, GE-IFWRVFL, and CP-FRVFL models on the ADNI dataset to evaluate their performance. Table 3.10 shows the experimental results on a common experimental setup. The proposed CP-FRVFL has superior performance at 75.79% average AUC, followed by GE-IFWRVFL at 75.36%, and IFRVFL at 75.19% and RVFL has last place at 72.73%. one can observe that IFRVFl has the lowest average rank (1.67) among all three proposed models and CP-FRVFL has a second position at 2.33.And, IFRVFL has overall 2 wins with no tie and loss. Moreover, CP-FRVFL has 1 win and GE-IFWRVFL has no wins. IFRVFL has a winning performance for the MCI_vs_AD case and as per the literature, it is the hardest case to diagnose. The proposed GE-IFWRVFL has better performance than the proposed IFRVFL in terms of average AUC. For CN_vs_MCI case, both the proposed CP-FRVFL and GE-IFRVFL models have better performance than the standard RVFL and proposed IFRVFL models.

Table 3.10: Experimental results of the proposed IFRVFL, GE-IFWRVFL, and CP-FRVFL models on the ADNI dataset.

	RVFL [110]	IFRVFL[76]	GE-IFWRVFL[77]	CP-FRVFL
CN_vs_AD	0.8903	0.898	0.8834	0.8843
CN_vs_MCI	0.6295	0.6763	0.7001	0.7188
MCI_vs_AD	0.6621	0.6813	0.6773	0.6705
Average AUC	0.7273	0.7519	0.7536	0.7579
Average Rank	3.33	1.67	2.67	2.33
Overall win tie loss	$[0, 0, 2]$	$[2, 0, 0]$	$[0, 0, 1]$	$[1, 0, 0]$

3.6 Summary

Standard RVFL has lower generalization performance over datasets with noise or outliers because it uses a uniform weighting strategy, which gives each training sample the same weight. In this chapter, we addressed this issue using fuzzy/intuitionic fuzzy techniques and to maintain the topological structure of the data, we use graph embedding technique. We proposed three variants of RVFL namely, IFRVFL, CP-FRVFL, and GE-IFWRVFL models. First, the proposed IFRVFL model uses intuitionistic fuzzy weighted approach for calculating the final parameters and hence, gets better optimal parameters than standard RVFL. The proposed IFRVFL model is robust to outliers and gives special attention to each sample by assigning different weights to consider its importance in classification problems. The proposed IFRVFL model is fast compared to IFTWSVM model as the former optimizes the objective function via optimization of system of linear equations while the later optimizes the pair of quadratic programming problems. Also, as the size of the dataset increases IFTWSVM, KRR and IFKRR suffer from memory issues and have more computational complexity. However, the proposed IFRVFL model has less complexity and fewer memory issues when either the sample size is large or feature size is large but not both. We evaluated the proposed IFRVFL and the existing models for the diagnosis of AD. The proposed IFRVFL emerged as the best classifier for the diagnosis of MCI versus AD subject, which is hard to classify as per the literature. Also, in CN versus AD and CN versus MCI case, the proposed IFRVFL model is among the top two performing models. Furthermore, we evaluated the effect of 8 different activation functions on the performance of randomized neural networks i.e., ELM, RVFL, and the proposed IFRVFL model. To check the robustness of the classification models, we evaluated the models on benchmark binary datasets. Second, we proposed a class probability-based effective score function that calculates each sample's membership value using a non-linear membership function. The proposed score function leverages the k-NN technique to calculate the class probability to use each training sample's neighbourhood information. Each sample is then given a score value based on the score function, which combines the class probability and membership value. Then, we proposed a novel class probability-based fuzzy RVFL (CP-FRVFL) by inserting the proposed score

values into the objective function of the standard RVFL network. The experimental results on KEEL and UCI datasets demonstrate that the proposed CP-FRVFL model has superior performance as compared to the baseline models. Third, we proposed a novel Graph Embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) network. Graph Embedding defines intrinsic and penalty graph to describe the geometrical properties of the data. Unlike RVFL and IFRVFL models, the proposed GE-IFWRVFL model considers the geometrical relationship of the data while calculating the final output parameters. The experimental results demonstrate that the proposed GE-IFWRVFL model has better generalization performance with the highest average accuracy and lowest average rank compared to the baseline models. The results also indicate that after having geometrical information about the data, the proposed GE-IFWRVFL model performs better than the IFRVFL and RVFL models. Moreover, results show that the proposed GE-IFWRVFL model performs better than the MVRVFL and IFTWSVM models.

This chapter developed three improved variants of the standard RVFL. The proposed IFRVFL and GE-IFWRVFL models are suitable for medium-sized datasets that contain noise and outliers. Nevertheless, these models are inefficient when dealing with massive data sets due to the necessity of calculating score weights in the kernel space. Moreover, the proposed CP-FRVFL is suitable for datasets having noise/outliers. RVFL still faces several issues. For instance, it utilizes a least square classifier, which may struggle to differentiate potentially overlapping patterns in the dataset. Additionally, it relies on computing the inverse of the matrix to get the final output parameters. In the following chapter, we address these concerns by employing hyperplane-based classification approaches.

Chapter 4

Random Vector Functional Link Network and Support Vector Machines based Novel Architectures

RVFL is an efficient network and has good generalization performance for pattern classification. However, it still has a few challenges, i.e. RVFL employs a least square classifier, which may encounter difficulty in distinguishing potentially overlapping patterns in the dataset and it computes the inverse of matrix to calculate the final output parameters. SVMs have better classification performance on datasets having overlapping patterns. On the other hand, SVMs do not have the inherent capacity to learn hierarchical or deep representations from raw data. The Kernel-based models such as SVMs suffer from higher computation and memory issues when the number of samples is large. A novel framework that combines SVMs with RVFL neural network can adapt and leverage the strengths of both hyperplane-based classifiers and neural networks, providing a more flexible solution for classification problems. There are several research [16] that combine SVM-based techniques with ANN architectures to take benefits from both fields. In this chapter, inspired by the work [31, 69] and the nonparallel classifier, i.e., NPSVM [35] and RVFL network, first, we propose a novel nonparallel RVFL (NPRVFL) model, which has superior properties as compared to NPSVM and RVFL network. Unlike RVFL, the proposed NPRVFL model finds two nonparallel proximal hyperplanes in RVFL's feature space (original features +

randomized features) to classify the data. This dual-hyperplane classification mechanism serves to significantly enhance the RVFL's efficacy in capturing complex patterns inherent in the data. Unlike RVFL and Twin SVM, the proposed NPRVFL doesn't need to calculate the inverse of a matrix while calculating the final output parameters of the model. Second, we propose extended robust energy-based least squares twin support vector machines (ext-RELS-TSVM) [78] and extended least squares twin support vector machines (ext-LSTSVM) [78], wherein the hyperplanes are calculated over extended feature space to improve the generalization performance of the baseline models.

4.1 Proposed nonparallel RVFL (NPRVFL) model

Unlike the standard RVFL network, in the proposed NPRVFL model, we need to construct two nonparallel proximal hyperplanes in the RVFL's feature space (D_1 and D_2) that pass through the origin.

$$P_+(x)\text{: } \hat{d}(x^+)\beta_+ = 0 \text{ and } P_-(x)\text{: } \hat{d}(x^-)\beta_- = 0 \tag{4.1}$$

Let $D_1 = [X_1\ H_1]$, $D_2 = [X_2\ H_2]$ and $\hat{d}(x) = [x\ \ \Psi(x)]$. Here $\Psi(.)$ is an activation function. Let positive class (+1) has m_1 samples and negative class (-1) has m_2 samples, where $N = m_1 + m_2$. Assume that $X_1 \in \mathbb{R}^{m_1 \times d}$ represents all positive class samples and $X_2 \in \mathbb{R}^{m_1 \times d}$ represents all negative class samples. Moreover, β_+ and β_- are the weights vector of hyperplanes corresponding to positive and negative classes, respectively. Let us define matrices H_1 and H_2 to represent the hidden layer (having L nodes) and output of the samples belonging to positive and negative classes, respectively. Thus,

$$H_1 = \begin{bmatrix} \Psi_1(x_1^+) & \cdots & \Psi_L(x_1^+) \\ \vdots & \vdots & \vdots \\ \Psi_1(x_{m_1}^+) & \cdots & \Psi_L(x_{m_1}^+) \end{bmatrix} \text{ and } H_2 = \begin{bmatrix} \Psi_1(x_1^-) & \cdots & \Psi_L(x_1^-) \\ \vdots & \vdots & \vdots \\ \Psi_1(x_{m_2}^-) & \cdots & \Psi_L(x_{m_2}^-) \end{bmatrix} \tag{4.2}$$

We seek two nonparallel hyperplanes (Eq. 4.1) by solving following convex QPPs:

$$\begin{aligned}
\min_{\beta_+,\eta_+,\eta_+^*,\xi_-} \quad & \frac{1}{2} \parallel \beta_+ \parallel^2 + c_1 e_+^T(\eta_+ + \eta_+^*) + c_2 e_-^T \xi_- \\
\text{subject to} \quad & D_1\beta_+ \leq \epsilon e_+ + \eta_+, \\
& -D_1\beta_+ \leq \epsilon e_+ + \eta_+^*, \\
& -D_2\beta_+ \geq e_- - \xi_-, \\
& \eta_+, \eta_+^*, \xi_- \geq 0,
\end{aligned} \tag{4.3}$$

and

$$\begin{aligned}
\min_{\beta_-,\eta_-,\eta_-^*,\xi_+} \quad & \frac{1}{2} \parallel \beta_- \parallel^2 + c_3 e_-^T(\eta_- + \eta_-^*) + c_4 e_+^T \xi_+ \\
\text{subject to} \quad & D_2\beta_- \leq \epsilon e_- + \eta_-, \\
& -D_2\beta_- \leq \epsilon e_- + \eta_-^*, \\
& D_1\beta_- \geq e_+ - \xi_+, \\
& \eta_-, \eta_-^*, \xi_+ \geq 0,
\end{aligned} \tag{4.4}$$

where c_i $(i = 1, 2, 3, 4)$ represents the regularization parameters, e_+ and e_- are vectors of one of appropriate dimension. To get the solution of QPPs (4.3) and (4.4), Karush Kuhn Tucker condition is employed. The Lagrangian of the problem (4.3) is calculated as:

$$\begin{aligned}
L(\beta_+, \eta_+, \eta_+^*, \xi_-, \alpha_1^+, \alpha_2^+, \alpha_3^-, \alpha_4^+, \alpha_5^+, \alpha_6^-) = & \frac{1}{2}\beta_+^T\beta_+ + c_1 e_+^T(\eta_+ + \eta_+^*) + c_2 e_-^T\xi_- \\
& + (\alpha_1^+)^T(D_1\beta_+ - \epsilon e_+ - \eta_+) \\
& + (\alpha_2^+)^T(-D_1\beta_+ - \epsilon e_+ - \eta_+^*) \\
& + (\alpha_3^-)^T(D_2\beta_+ + e_- - \xi_-) \\
& - (\alpha_4^+)^T\eta_+ - (\alpha_5^+)^T\eta_+^* - (\alpha_6^-)^T\xi_-.
\end{aligned} \tag{4.5}$$

Using the necessary and sufficient K.K.T. conditions, we have

$$\frac{\partial L}{\partial \beta_+} = \beta_+ + D_1^T\alpha_1^+ - D_1^T\alpha_2^+ + D_2^T\alpha_3^- = 0, \tag{4.6}$$

$$\frac{\partial L}{\partial \eta_+} = c_1 e_+ - \alpha_1^+ - \alpha_4^+ = 0, \tag{4.7}$$

$$\frac{\partial L}{\partial \eta_+^*} = c_1 e_+ - \alpha_2^+ - \alpha_5^+ = 0, \tag{4.8}$$

$$\frac{\partial L}{\partial \xi_-} = c_2 e_- - \alpha_3^- - \alpha_6^- = 0. \tag{4.9}$$

From Eq. (4.6), we have

$$\beta_+ = D_1^T(\alpha_2^+ - \alpha_1^+) - D_2^T\alpha_3^+. \tag{4.10}$$

Since all $\alpha_i^{\pm} \geq 0$ and from Eqs. (4.7), (4.8) and (4.9), we have, $0 \leq \alpha_1^+, \alpha_2^+ \leq c_1 e_+$ and $0 \leq \alpha_3^- \leq c_2 e_-$. After putting Eq. (4.10) into the Lagrangian (4.5) and using Eqs. (4.6)-(4.9), we obtain the dual QPP (4.3) as follows:

$$\begin{aligned} \min_{\alpha^*, \alpha_3^-} \quad & \frac{1}{2} \wedge_1 + \epsilon e_+^T(\alpha_2^+ + \alpha_1^+) - e_-^T\alpha_3^- \\ \text{subject to} \quad & 0 \leq \alpha_1^+, \alpha_2^+ \leq c_1 e_+, \\ & 0 \leq \alpha_3^- \leq c_2 e_-, \end{aligned} \tag{4.11}$$

where $\wedge_1 = (\alpha_2^+ - \alpha_1^+)^T D_1 D_1^T(\alpha_2^+ - \alpha_1^+) - 2(\alpha_2^+ - \alpha_1^+)^T D_1 D_2^T \alpha_3^+ + (\alpha_3^-)^T D_2 D_2^T \alpha_3^+$ and $\alpha^* = [(\alpha_2^+)^T\ (\alpha_1^+)^T]^T$.

Assuming $\Lambda = \begin{bmatrix} M_1 & -M_2 \\ -M_2^T & M_3 \end{bmatrix}$, $\bar{\pi} = [(\alpha_2^+)^T\ (\alpha_1^+)^T\ (\alpha_3^-)^T]^T$, $\bar{K} = [(\epsilon e_+)^T\ (\epsilon e_+)^T\ -e_-^T]^T$, $\bar{e} = [-e_+^T, e_+^T, e_-^T]^T$, $\bar{C} = [c_1 e_+^T\ c_1 e_+^T\ c_2 e_-^T]^T$, $M_1 = \begin{bmatrix} D_1 D_1^T & -D_1 D_1^T \\ -D_1 D_1^T & D_1 D_1^T \end{bmatrix}$, $M_2 =$

$\begin{bmatrix} D_1 D_2^T \\ -D_1 D_2^T \end{bmatrix}$, $M_3 = D_2 D_2^T$, the problem (4.11) is reformulated as:

$$\begin{aligned} \min_{\bar{\pi}} \quad & \frac{1}{2}\bar{\pi}^T \Lambda \bar{\pi} + \bar{K}^T \bar{\pi} \\ \text{subject to} \quad & 0 \leq \bar{\pi} \leq \bar{C}. \end{aligned} \tag{4.12}$$

After solving QPP (4.12), we get β_+. The Lagrangian of the QPP (4.4) is calculated as:

$$\begin{aligned} L(\beta_-, \eta_-, \eta_-^*, \xi_+, \alpha_1^-, \alpha_2^-, \alpha_3^+, \alpha_4^-, \alpha_5^-, \alpha_6^+) = & \frac{1}{2}\beta_-^T \beta_- + c_1 e_-^T (\eta_+ + \eta_+^*) + c_2 e_+^T \xi_+ \\ & + (\alpha_1^-)^T (D_2 \beta_- - \epsilon e_- - \eta_-) \\ & + (\alpha_2^-)^T (-D_- \beta_- - \epsilon e_- - \eta_-^*) \\ & + (\alpha_3^+)^T (D_2 \beta_+ - e_+ + \xi_+) \\ & - (\alpha_4^-)^T \eta_+ - (\alpha_5^-)^T \eta_+^* - (\alpha_6^+)^T \xi_-. \end{aligned} \tag{4.13}$$

Using the necessary and sufficient K.K.T. conditions, we have

$$\frac{\partial L}{\partial \beta_-} = \beta_- + D_2^T \alpha_1^- - D_2^T \alpha_2^- - D_1^T \alpha_3^+ = 0, \tag{4.14}$$

$$\frac{\partial L}{\partial \eta_-} = c_3 e_- - \alpha_1^- - \alpha_4^- = 0, \tag{4.15}$$

$$\frac{\partial L}{\partial \eta_-^*} = c_3 e_- - \alpha_2^- - \alpha_5^- = 0, \tag{4.16}$$

$$\frac{\partial L}{\partial \xi_-} = c_4 e_+ - \alpha_3^+ - \alpha_6^+ = 0. \tag{4.17}$$

From Eq. (4.14), we have

$$\beta_- = D_2^T (\alpha_2^- - \alpha_1^-) - D_1^T \alpha_3^+. \tag{4.18}$$

Since all $\alpha_i^{\pm} \geq 0$ and from Eqs. (4.15), (4.16) and (4.17), we have, $0 \leq \alpha_1^-, \alpha_2^- \leq c_3 e_-$ and $0 \leq \alpha_4^- \leq c_4 e_+$. After putting Eq. (4.18) into the Lagrangian (4.13) and using Eqs. (4.14)-(4.17), we obtain the dual problem of the problem (4.4) as follows:

$$\begin{aligned} \min_{\alpha_-^*, \alpha_3^+} \quad & \frac{1}{2}\Lambda_2 + \epsilon e_-^T(\alpha_2^- + \alpha_1^-) - e_+^T\alpha_3^+ \\ \text{subject to} \quad & 0 \leq \alpha_1^-, \alpha_2^- \leq c_3 e_-, \\ & 0 \leq \alpha_3^+ \leq c_4 e_+, \end{aligned} \tag{4.19}$$

where $\Lambda_2 = (\alpha_2^- - \alpha_1^-)^T D_2 D_2^T(\alpha_2^- - \alpha_1^-) - (\alpha_2^- - \alpha_1^-)^T D_2 D_1^T \alpha_3^+ + \frac{1}{2}(\alpha_3^+)^T D_1 D_1^T \alpha_3^+$ and $\alpha_-^* = [(\alpha_1^-)^T \ (\alpha_2^-)^T]^T$.

Assuming $\hat{\Lambda} = \begin{bmatrix} M_1 & M_2 \\ M_2^T & M_3 \end{bmatrix}$, $\hat{\pi} = [(\alpha_2^-)^T \ (\alpha_1^-)^T \ (\alpha_3^+)^T]^T$, $\hat{K} = [(\epsilon e_-)^T \ (\epsilon e_-)^T \ -e_+]^T$, $M_1 = \begin{bmatrix} D_2 D_2^T & -D_2 D_2^T \\ -D_2 D_2^T & D_2 D_2^T \end{bmatrix}$, $M_2 = \begin{bmatrix} D_2 D_1^T \\ -D_2 D_1^T \end{bmatrix}$, $M_3 = D_1 D_1^T$.

Using the above notations, the problem (4.19) is reformulated as

$$\begin{aligned} \min_{\hat{\pi}} \quad & \frac{1}{2}\hat{\pi}^T \hat{\Lambda} \hat{\pi} + \hat{K}^T \hat{\pi} \\ \text{subject to} \quad & 0 \leq \hat{\pi} \leq \hat{C}, \end{aligned} \tag{4.20}$$

where $\hat{C} = [c_3(e_-)^T \ c_3(e_-)^T \ c_4(e_-)^T]^T$. After solving QPP (4.20), we get β_-.

For be a new sample x, a class is assigned using the following decision function:

$$g(x) = \underset{\{m=+,-\}}{\text{argmin}} |\hat{d}(x)\beta_m|. \tag{4.21}$$

4.2 Experimental results

In this section, we evaluate the baseline models and the proposed NPRVFL model on UCI datasets.

4.2.1 Experimental setup

All the experiments are performed on a system with MATLAB R2017b, Intel(R) Xeon(R) CPU E5-2697 v4 2.30 GHz, 128-GB RAM, and Windows-10 platform. We used

Gaussian kernel, $K(x_1, x_2) = exp(-\frac{||x_1-x_2||^2}{\mu^2})$, where μ denotes the kernel parameter. The dataset is randomly partitioned into $30 : 70$ ratio of testing and training sets, respectively. The hyperparameters corresponding to different models are optimized using grid search approach via 5-fold cross validation. In 5-fold cross validation, the dataset is randomly partitioned into 5 disjoint sets wherein 1 set is reserved for testing and the rest are used for training. The performance of each model corresponding to the best hyperparameters is given as the final accuracy of a model. For general TWSVM with pinball loss function (pin-GTSVM) [349], SVM, TWSVM, NPSVM and the proposed NPRVFL models, the different parameters involved corresponding to different models are chosen from the following range: $c_i = \{10^{-5}, 10^{-4}, \ldots, 10^4, 10^5\}$ for $i = 1, 2, 3, 4$. For ELM, RVFL, MVRVFL, and the proposed NPRVFL models, the number of hidden neurons is taken from the range as $N = 3 : 20 : 203$. Here, we take relu activation function in all NN-based models. The parameter ϵ is chosen from the range as: $0 : 0.05 : 0.5$.

Table 4.1: The experimental results of the proposed NPRVFL and the baseline models on UCI datasets.

Datasets	pin-GTSVM [349]	SVM [17]	TWSVM [22]	NPSVM [35]	ELM [341]	RVFL [11]	MVRVFL [145]	NPRVFL
balloons	0.5	0.75	0.75	0.75	0.6	0.5	0.5	**0.75**
breast-cancer	0.6964	0.6478	0.6944	0.6915	0.6774	0.6994	0.7123	0.7044
breast-cancer-wisc	0.9571	0.939	0.8937	0.9246	0.9568	0.9786	0.9363	0.9427
breast-cancer-wisc-diag	0.9444	0.9515	0.9444	0.9154	0.9521	0.9589	0.9129	**0.9637**
congressional-voting	0.5895	0.5586	0.5895	0.4621	0.4931	0.5021	0.5123	**0.6186**
credit-approval	0.7969	0.8078	0.7571	0.8831	0.8559	0.8654	0.831	0.8638
echocardiogram	0.7273	0.7682	0.8015	0.8182	0.8091	0.8182	0.8136	**0.8182**
fertility	0.4348	0.4701	0.413	0.5707	0.4946	0.5408	0.4565	0.4348
haberman-survival	0.4975	0.5623	0.5	0.4806	0.5418	0.5589	0.5993	**0.6818**
heart-hungarian	0.7419	0.6659	0.6902	0.8023	0.8381	0.8335	0.7231	0.8303
hepatitis	0.8206	0.7718	0.7369	0.6986	0.6794	0.6185	0.784	**0.8554**
ionosphere	0.9726	0.8932	0.8539	0.808	0.8793	0.8334	0.8334	0.9384
molec-biol-promoter	0.735	0.735	0.735	0.735	0.694	0.6898	0.7519	**0.7613**
musk-1	0.9575	0.9508	0.9512	0.8531	0.836	0.9031	0.7573	0.8685
parkinsons	0.8889	0.8889	0.9333	0.8	0.8422	0.8556	0.8778	**0.9333**
pittsburg-bridges-T-OR-D	0.4286	0.5179	0.5893	0.6071	0.45	0.4821	0.4643	**0.7857**
statlog-australian-credit	0.4594	0.4801	0.5066	0.519	0.5302	0.5603	0.5275	0.5197
statlog-german-credit	0.6092	0.6475	0.6644	0.6729	0.649	0.6619	0.6485	**0.6895**
statlog-heart	0.8031	0.718	0.7693	0.8176	0.847	0.8684	0.7198	0.8188
vertebral-column-2clases	0.7668	0.7424	0.7776	0.8333	0.8295	0.822	0.8117	**0.8582**
Average AUC.	0.7164	0.7233	0.7276	0.7322	0.7228	0.7275	0.7087	**0.7819**

4.2.2 Experimental analysis on UCI datasets

The experimental results corresponding to the proposed NPRVFL and the baseline models are shown in Table 4.1. One can see that the proposed NPRVFL model has the highest average AUC equal to 78.19%, followed by NPSVM with average AUC equal to 73.22%.

Table 4.2: Friedman ranks of the proposed NPRVFL and the baseline models on UCI datasets.

Dataset Name	pin-GTSVM [349]	SVM [17]	TWSVM [22]	NPSVM [35]	ELM [341]	RVFL [11]	MVRVFL [145]	NPRVFL
balloons	7	2.5	2.5	2.5	5	7	7	2.5
breast-cancer	4	8	5	6	7	3	1	2
breast-cancer-wisc	2	5	8	7	3	1	6	4
breast-cancer-wisc-diag	5.5	4	5.5	7	3	2	8	1
congressional-voting	2.5	4	2.5	8	7	6	5	1
credit-approval	7	6	8	1	4	2	5	3
echocardiogram	8	7	6	2	5	2	4	2
fertility	6.5	4	8	1	3	2	5	6.5
haberman-survival	7	3	6	8	5	4	2	1
heart-hungarian	5	8	7	4	1	2	6	3
hepatitis	2	4	5	6	7	8	3	1
ionosphere	1	3	5	8	4	6.5	6.5	2
molec-biol-promoter	4.5	4.5	4.5	4.5	7	8	2	1
musk-1	1	3	2	6	7	4	8	5
parkinsons	3.5	3.5	1.5	8	7	6	5	1.5
pittsburg-bridges-T-OR-D	8	4	3	2	7	5	6	1
statlog-australian-credit	8	7	6	5	2	1	3	4
statlog-german-credit	8	7	3	2	5	4	6	1
statlog-heart	5	8	6	4	2	1	7	3
vertebral-column-2clases	7	8	6	2	3	4	5	1
Average rank	5.125	5.175	5.025	4.7	4.7	3.925	5.025	**2.325**

Further, pin-GTSVM, SVM, TWSVM, ELM, RVFL, and MVRVFL have average AUC equal to 71.64%, 72.33%, 72.76%, 72.28%, 72.75% and 70.87%, respectively. The results show that the proposed NPRVFL model is almost 5% superior than NPSVM and 6% superior than RVFL model. It demonstrates that the proposed hybrid NPRVFL model has better generalization performance than its constitutes models. One can observe that the proposed NPRVFL model has AUC equal to 96.37% on breast-cancer-wisc-diag dataset, whereas, the existing NPSVM has AUC equal to 91.54%. Moreover, the proposed NPRVFL model has AUC equal to 85.54% and NPSVM has AUC equal to 69.86% over the hepatitis dataset. To evaluate the performance further, we rank the model on each dataset. Therefore, the most successful model should have a lower rank. Table 4.2 shows the Friedman ranks and one can observe that the proposed NPRVFL model has smallest rank, i.e., 2.35, whereas RVFL model has second position with 3.925 rank. Both NPSVM and ELM have 4.7 ranks. Moreover, we employ the Friedman test. After calculation, $\chi_F^2 = 22.67$ and $F_F = 3.67$. Here, we have $N = 20$ number of datasets and $k = 8$ compared models. Therefore, from the statistical F-distribution table, $F_{(7,133)} = 2.08$. Since, $3.67 > 2.08$, hence, the null hypothesis is rejected. Thus, there is a significant difference among the compared models. To check the pairwise significant difference between the models, we follow the Nemenyi posthoc test. For the Nemenyi test, after calculations, the critical difference (C.D.) = 2.34.

The models are deemed to be significantly distinct if their average ranks differ by at least 2.34. Therefore, the proposed NPRVFL model is statistically different from pin-GTSVM, SVM, TWSVM, NPSVM, ELM, and MVRVFL models.

4.2.3 Computational complexity analysis

Let m be the number of samples. The computational complexity of classical SVM is approximately $O(m^3)$. For TWSVM, it solves two problems and each of which is about size $(m/2)$. Thus, its computation complexity is $O(2(m/2)^3)$. TWSVM is nearly four times faster than SVM. As for NPSVM, it also solves two problems and each of them is about size $(3m/2)$. Thus, the optimization of NPSVM costs around $O(2(3m/2)^3)$ [350]. The computational cost of NPRVFL comes from calculating the RVFL's feature space and solving two QPPs and each of them is nearly size $(3m/2)$. The optimization of NPRVFL costs around $O(2(3m/2)^3)$.

4.3 Support vector machines with randomized neural network-based autoencoder

Auto-encoder is a special type of ANN that is used to learn informative features from data. RVFL uses two types of features, i.e., original features and randomized features, that make it a special randomized NN. These hybrid features improve the generalization performance of the RVFL network. We used a sparse auto-encoder with L_1 norm regularization to learn the auxiliary feature representation from the original feature space. These new autoencoder-based features are concatenated with the original features to get the extended feature space. Finally, the proposed extended RELS-TSVM (ext-RELS-TSVM) and extended LSTSVM (ext-LSTSVM) get the hyperplanes over extended feature space.

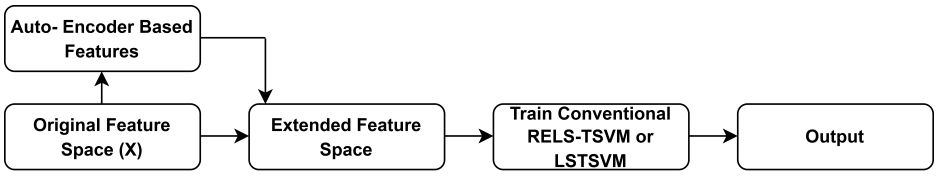


Figure 4.1: Flowchart of the proposed models.

4.3.1 Proposed ext-RELS-TSVM and ext-LSTSVM

In this section, we discuss the proposed binary classification algorithms. The optimization problem of the sparse auto-encoder with L hidden neurons can be written as:

$$\Omega_\theta = \min_\theta ||\hat{H}\theta - X||_2^2 + ||\theta||_1, \tag{4.22}$$

where $\hat{H} \in \mathbb{R}^{N\times L}$ represents the hidden layer output matrix, and $\theta \in \mathbb{R}^{L\times m}$ is the output weights matrix. The optimization problem (4.22) is solved via a fast iterative shrinkage thresholding algorithm (FISTA) [111] and the hidden biases are calculated as:

$$\sigma_i = \frac{1}{m}\sum_{j=1}^{m}\theta_{ij}, \quad i = 1, 2, \ldots, L, \tag{4.23}$$

where σ_i represents the bias in the i^{th} enhancement node of the hidden layer. The process of constructing the proposed ext-RELS-TSVM model is given in Algorithm 4.1 and a similar process is followed for constructing the proposed ext-LSTSVM with extended feature space. The flowchart of the proposed ext-RELS-TSVM model is given in Figure 4.1.

4.3.1.1 Computational complexity analysis

Let $M \times d$ be a binary class training dataset with m_1 number of samples belonging to the positive class and m_2 number of samples belonging to a negative class such that $M = m_1 + m_2$. By standard mathematical implementation, the inversion of $d \times d$ requires $O(d^3)$ times [351]. In the LSTSVM model, two matrix inversions are required for each of size $(d+1)\times(d+1)$. Most of the computational complexity comes from matrix inversion. Thus, the time complexity of LSTSVM is given as: $2 \times O((d+1)^3) \approx 2 \times O(d^3)$. RELS-TSVM has a similar computational complexity to the LSTSVM model. The approximate

Algorithm 4.1 Extended RELS-TSVM

- Input: Let X contains the original training samples. Let $\phi(\cdot)$ and L be the given activation function and number of neurons in the hidden layer used in auto-encoder 4.22 and let the matrix $F_1 \in \mathbb{R}^{N_1 \times m}$ contains samples of class +1 (from X) and the rest samples of class -1 is contained in matrix $F_2 \in \mathbb{R}^{N_2 \times m}$.

- Calculate the output weight matrix θ by solving 4.22 and then calculate the biases by 4.23. Apply the non-linear activation function $\phi(.)$ to calculate the additional features matrix from the original feature matrix as:

$$\hat{X} = \phi(X\theta^T + \sigma) \tag{4.24}$$

  $\hat{X}$ is the additional feature representation of the original feature space X.

- Define the extended feature space as $X^* = [X, \hat{X}]$ and obtained $F_1^* = [F_1, \hat{F}_1]$ and $F_2^* = [F_2, \hat{F}_2]$, where $\hat{F}_1$ and $\hat{F}_2$ are the samples of class +1 and -1, respectively from $\hat{X}$.

- After getting the extended feature space, define the objective function of ext-RELS-TSVM as:

$$\begin{aligned} \underset{w_1,b_1}{min} \ & \frac{1}{2}\|F_1^* w_1 + e_1 b_1\|^2 + \frac{\lambda_1}{2}\|\eta_1\|^2 + \frac{\lambda_3}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 \\ s.t. \ & -(F_2^* \bar{w}_1 + e_2 b_1) + \eta_1 = E_1 \end{aligned} \tag{4.25}$$

  and

$$\begin{aligned} \underset{w_2,b_2}{min} \ & \frac{1}{2}\|F_2^* w_2 + e_2 b_2\|^2 + \frac{\lambda_2}{2}\|\eta_2\|^2 + \frac{\lambda_4}{2}\left\|\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}\right\|^2 \\ s.t. \ & (F_1^* w_2 + e_1 b_2) + \eta_2 = E_2. \end{aligned} \tag{4.26}$$

- Solve (4.25) and (4.26) as discussed in [303] and obtained the corresponding solutions. Let $x^{'}$ be the new sample, then the final decision of the proposed ext-RELS-TSVM is taken as in the conventional RELS-TSVM.

Table 4.3: Classification accuracy (%) of LSTSVM, RELS-TSVM, SP-RVFL and proposed ext-LSTSVM, ext-RELS-TSVM. The best results are highlighted in bold.

Datasets	LSTSVM [23]	RELS-TSVM [303]	SP-RVFL [120]	ext-LSTSVM	ext-RELS-TSVM
breast-cancer	72.5352	72.8873	**75**	73.5915	69.7183
breast-cancer-wisc	97	**97.4286**	97	**97.4286**	**97.4286**
chess-krvkp	95.3692	95.4631	96.9962	96.9337	**97.0901**
congress-voting	58.945	58.945	**61.4679**	**61.4679**	58.945
credit-approval	87.064	86.7733	86.6279	**87.5**	86.6279
echocardiogram	84.8485	84.8485	**85.6061**	**85.6061**	84.8485
heart-hungarian	84.9315	85.6164	85.274	**86.3014**	82.8767
molec-biol-promoter	65.3846	64.4231	81.7308	56.7308	**89.4231**
oocytes_trisopterus_nucleus_2f	80.5921	80.5921	81.0307	82.5658	**84.7588**
ozone	**97.1609**	**97.1609**	**97.1609**	**97.1609**	97.082
parkinsons	88.7755	86.7347	90.8163	88.7755	**91.8367**
pima	76.6927	76.5625	76.8229	**77.6042**	76.4323
statlog-australian-credit	67.0058	66.8605	**68.0233**	67.0058	66.4244
statlog-german-credit	78.5	77.7	77.3	77.4	**79.4**
tic-tac-toe	97.5941	97.5941	**99.0586**	98.954	97.5941
Average accuracy	82.16	81.97	83.99	82.34	**84.03**

Table 4.4: Friedman rank of LSTSVM, RELS-TSVM, SP-RVFL and proposed ext-LSTSVM, ext-RELS-TSVM.

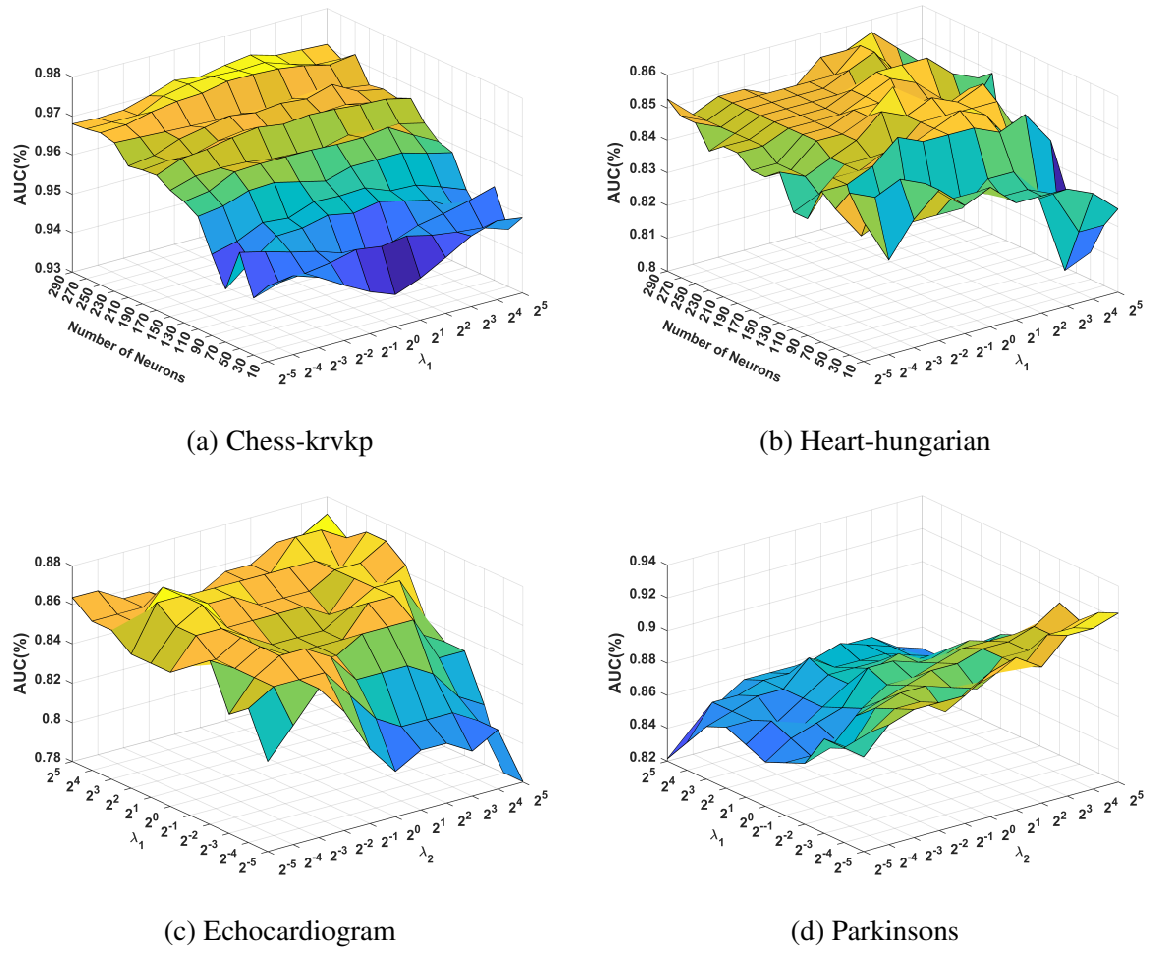
Datasets	LSTSVM [23]	RELS-TSVM [303]	SP-RVFL [120]	ext-LSTSVM	ext-RELS-TSVM
breast-cancer	4	3	1	2	5
breast-cancer-wisc	4.5	2	4.5	2	2
chess-krvkp	5	4	2	3	1
congress-voting	4	4	1.5	1.5	4
credit-approval	2	3	4.5	1	4.5
echocardiogram	4	4	1.5	1.5	4
heart-hungarian	4	2	3	1	5
molec-biol-promoter	3	4	2	5	1
oocytes_trisopterus_nucleus_2f	4.5	4.5	3	2	1
ozone	2.5	2.5	2.5	2.5	5
parkinsons	3.5	5	2	3.5	1
pima	3	4	2	1	5
statlog-australian-credit	2.5	4	1	2.5	5
statlog-german-credit	2	3	5	4	1
tic-tac-toe	4	4	1	2	4
Average rank	3.5	3.53	2.43	**2.3**	3.23

computational complexity of each proposed ext-LSTSVM and ext-RELS-TSVM model has $2 \times O((d+L)^3)$, where L is the extended features. In addition to this, some computational cost comes from calculating the extended feature space by using a sparse autoencoder with L_1 norm.

4.3.1.2 Experimental setup and results analysis

In this section, we discuss the experimental setup details and illustrate the obtained results. All the experiments are carried out over the publicly available binary datasets in the UCI repository [352] and we set up the same environment and naming convention as in [267]. We used relu, selu, and sigmoid activation functions and via grid search selected the optimal function. The number of neurons varies in range 10: 20: 300. The parameters $\lambda_{(1,2,3,4)}$ and the energy parameters $E_{(1,2)}$ are taken from the range as follows: $\{2^j \mid j = -5, -4, \ldots, 5\}$ and $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, respectively. All experiments are conducted on a computer running Windows 10 with 16 GB of RAM and MATLAB R2021a.

Table 4.3 shows the average accuracy of the proposed ext-LSTSVM and ext-RELS-TSVM models and baseline models. One can see that the proposed model ext-RELS-TSVM has the highest accuracy (84.03 %) which is almost 2% higher than RELS-TSVM (81.97 %). The other models have the following accuracy, i.e., ext-LSTSVM (82.34 %), LSTSVM (82.16 %), and SP-RVFL (83.99 %). There is a small improvement of 0.16% in ext-LSTSVM as compared to standard LSTSVM. The proposed ext-LSTSVM model has better accuracy than LSTSVM on 10 out of 15 data sets. There are some datasets where one can observe the impact of additional features approaches such as the proposed ext-LSTSVM has winning performance with significant improvements over the following datasets, i.e., chess-krvkp, congress-voting, echocardiogram, heart-hungarian and pima with accuracy 96.93%, 61.47%, 85.61%, 86.30% and 77.60%, respectively. The proposed ext-RELS-TSVM has better accuracy (84.03 %) than existing RELS-TSVM (81.97 %). One can observe that on chess-krvkp dataset there is almost 2% improvement in accuracy in the proposed ext-RELS-TSVM model as compared to the RELS-TSVM model. It shows that having additional features in the training process is an effective approach. Moreover, we conducted a Friedman rank test that rank the models according to their performance on each



(a) Chess-krvkp

(b) Heart-hungarian

(c) Echocardiogram

(d) Parkinsons

Figure 4.2: Performance of ext-RELS-TSVM with varying number of neurons, λ_1 and λ_2.

dataset. Table 4.4 shows the average rank of each given model. The proposed ext-LSTSVM has the smallest average rank (2.3) among the compared models, the second average rank model is SP-RVFL (2.43) and the third average rank model is ext-RELS-TSVM (3.23). The lower rank reflects that the model performance is good over maximum datasets. It shows that the proposed models have comparable results with SP-RVFL. It can also be observed that the proposed ext-RELS-TSVM and ext-LSTSVM have lower rank than the conventional RELS-TSVM and LSTSVM, respectively. It reflects that there is advantage of using autoencoder based features that improve the generalization performance of the baseline models. Figure 5.2 shows the classification performance of the proposed ext-RELS-TSVM model on chess-krvkp, heart-hungarian, echocardiogram and parkinsons data sets. In Figure 5.2a and 5.2b, x-axis and y- axis represents λ_1 and number of neurons in the hidden layer parameters, respectively and z-axis shows the accuracy of the proposed ext-RELS-TSVM model. In Figure 5.2a, one can see that the accuracy of the proposed ext-RELS-TSVM model is increasing as the number the neurons increases. The decreasing performance of the ext-RELS-TSVM can be seen in Figure 5.2b with higher λ_1 values and small number of neurons. It shows that having proper tuning of the neurons, one can get the optimal performance of ext-RELS-TSVM model. In Figure 5.2c and 5.2d, x-axis and y- axis represent λ_2 and λ_1 parameters, respectively and z-axis shows the classification performance of the proposed ext-RELS-TSVM model. Figure 5.2c shows that the proposed ext-RELS-TSVM has lower performance with higher λ_2 values and small λ_1 values and as the value of the λ_2 parameter is decreasing the performance of ext-RELS-TSVM is increasing. The proposed ext-RELS-TSVM model has decreasing performance with higher λ_1 values on Parkinsons dataset in Figure 5.2d. Therefore, to get the best performance of the ext-RELS-TSVM model, one must carefully adjust the parameters according to the problems.

4.4 Experimental results analysis of the proposed NPRVFL, ext-RELS-TSVM, and ext-LSTSVM models on ADNI dataset

To evaluate the common performance of the proposed models, we conducted the experiment on the ADNI dataset. Table 4.5 shows the experimental results. The proposed NPRVFL has superior performance at 76.78% average accuracy, followed by ext-RELS-TSVM at 72.89%, ext-LSTSVM at 72.07%, and RVFL at 72.73%. Except NPRVFL, all models have the same average rank and the proposed NPRVFL has the lowest rank at 1, which indicates that in all three cases, NPRVFL has a winning performance. One can observe that the proposed NPRVFL has 70.33% AUC for MCI_vs_AD case, which is the hardest case to classify. Table 4.6 shows the experimental results of the models proposed in chapters 3 and 4. One can observe that NPRVFL has the overall best performance w.r.t average accuracy and average rank.

Table 4.5: Experimental results of the RVFL and the proposed ext-LSTSVM, ext-RELS-TSVM, and NPRVFL models on the ADNI dataset.

	RVFL [110]	ext-LSTSVM	ext-RELS-TSVM	NPRVFL
CN_vs_AD	0.8903	0.8492	0.8818	0.9117
CN_vs_MCI	0.6295	0.6353	0.6728	0.6883
MCI_vs_AD	0.6621	0.6777	0.6321	0.7033
Average accuracy	0.7273	0.7207	0.7289	0.7678
Average rank	3	3	3	1
Overall win tie loss	$[0, 0, 1]$	$[0, 0, 1]$	$[0, 0, 1]$	$[3, 0, 0]$

Table 4.6: Experimental results of the RVFL and the proposed models, IFRVFL, GE-IFWRVFL, CP-FRVFL, ext-LSTSVM, and ext-RELS-TSVM, and the NPRVFL on the ADNI dataset.

	RVFL [110]	IFRVFL	GE-IFWRVFL	CP-FRVFL	ext-LSTSVM	ext-RELS-TSVM	NPRVFL
CN_vs_AD	0.8903	0.898	0.8834	0.8843	0.8492	0.8818	0.9117
CN_vs_MCI	0.6295	0.6763	0.7001	0.7188	0.6353	0.6728	0.6883
MCI_vs_AD	0.6621	0.6813	0.6773	0.6705	0.6777	0.6321	0.7033
Average accuracy	0.7273	0.7519	0.7536	0.7579	0.7207	0.7289	0.7678
Average rank	5.33	2.67	3.67	3.33	5.33	6	1.67
Overall win tie loss	$[0, 0, 1]$	$[0, 0, 0]$	$[0, 0, 0]$	$[1, 0, 0]$	$[0, 0, 1]$	$[0, 0, 1]$	$[2, 0, 0]$

4.5 Summary

In this chapter, we presented three variants of the standard RVFL model. Unlike standard RVFL, which uses a least square classifier to classify the data, all three proposed models use hyperplane-based techniques for classification tasks. The proposed models have the potential to maintain transparency in the decision-making process, simultaneously capitalizing on the expressive representation capabilities inherent in neural networks. We proposed nonparallel RVFL model which classifies the data using proximal hyperplanes. Unlike NPSVM, the hyperplanes in the proposed NPRVFL pass through the origin. The proposed NPRVFL constructed two primal problems with inequality constraints and uses ϵ insensitive loss function and soft margin loss function (Hinge loss function). We conducted the experiments over several UCI datasets and the experimental results demonstrate that the proposed NPRVFL model has superior performance than baseline models, i.e., RVFL and NPSVM models. The proposed NPRVFL model has almost 5% more accuracy than NPSVM and RVFL.

In the second approach, we proposed improved variants of RELS-TSVM and LSTSVM with extended feature space for binary classification problems. First, the additional features are calculated via sparse auto-encoder with L_1-norm regularization from the original feature space and second, the conventional RELS-TSVM and LSTSVM are trained over new extended feature space (original + additional features). The experiments over 15 UCI binary datasets demonstrate that the proposed idea improves the generalization performance of these baseline models. There are almost 2% improvements in the performance (average accuracy) in RELS-TSVM model (by using spare autoencoder-based features) and the proposed ext-LSTSVM has won over 10 out of 15 datasets as compared to LSTSVM model. It shows that sparse auto-encoder-based features have significant information and improve the generalization performance of baseline models.

Here, we developed three variants of the standard RVFL. The proposed models are suitable for datasets that contain overlapping patterns. The proposed NPRVFL can adapt and leverage the strengths of both hyperplane-based classifiers and neural networks, providing a more flexible solution for classification problems. Unlike RVFL, ext-RELS-TSVM, and ext-LSTSVM, the proposed NPRVFL doesn't need to calculate the inverse of the matri-

ces to calculate the final output parameters. The proposed dual-hyperplane classification mechanism serves to significantly enhance the RVFL's efficacy in capturing complex patterns inherent in the data. However, RVFL and the proposed models use a random feature mapping mechanism, and hence, it has an unstable performance. Ensembles often enhance generalization by combining the strengths of multiple models. They can mitigate overfitting and capture diverse patterns in the data. The variability in individual RVFL models can be averaged out in an ensemble, leading to a more stable overall prediction. In the next chapter, we study RVFL with ensemble learning approaches.

Chapter 5

Random Vector Functional Link Network based Ensemble Learning

In the previous chapters, the proposed RVFL variants are robust, however, they are unstable classifiers and have less features representation learning. Single RVFL may encounter challenges in effectively capturing intricate and complex patterns inherent in the dataset. Ensemble methodologies typically enhance the generalization performance by harnessing the strengths of multiple models. They possess the capability to mitigate overfitting issues and adeptly capture a diverse spectrum of patterns within the dataset. Ensembles frequently surpass the predictive efficacy of individual models, thereby achieving heightened levels of accuracy and reliability in their predictions. In this chapter, we improve the robustness, stability, and generalization performance of RVFL with ensemble learning methods. With the randomization process, RVFL is unstable and sensitive to noise/outliers due to L_2 norm. The standard RVFL has only one hidden layer to extract the meaningful features from the original dataset and hence, it is less capable of representing more complex hidden patterns within dataset. To make the shallow RVFL model powerful in representation learning and motivated by [73, 353], in this chapter, we propose extended feature RVFL (efRVFL) [79] model that is trained over extended feature space generated from original dataset. Using regression technique, new features are constructed via original features. The new features are concatenated with original features to get extended feature space. This feature mechanism makes the proposed efRVFL model more capable of capturing highly non-linear hidden

patterns within datasets. In the proposed efRVFL model, the extended feature space is generated via randomization process. Therefore, the feature matrix has original features, newly generated features (supervised) features, and randomized features (unsupervised). In the proposed efRVFL, the parameters between the input layer and hidden layer are generated randomly and fixed throughout the training phase. And, the output parameters between the hidden layer and the output layer are calculated via the closed-form method. Furthermore, to increase the stability and robustness capability of the proposed efRVFL, we also develop an ensemble of extended feature RVFL [79], i.e., en-efRVFL. In the proposed en-efRVFL model, each base model (efRVFL) is trained over different extended feature spaces to increase individual accuracy and diversity among the base models. The final decision of en-efRVFL is taken via average voting scheme. Three fundamental reasons, i.e., statistical, computational, and representational [354], justify why an ensemble model may perform better than a single model. The statistical problem arises when the size of hypothesis space is very large compared to the amount of training data on which the classifier is trained. Several machine learning algorithms (such as NNs) perform local search to get the solution therefore it might get stuck in local optima. Optimal training of NNs is NP-hard problem. Therefore, to construct an ensemble model by doing the local search from different starting points may give a better approximation function than any of the individual base models. Since the proposed efRVFL model is a flexible classifier therefore by constructing an ensemble (en-efRVFL) out of several efRVFL models, the en-efRVFL model can reduce the risk of choosing the wrong classifier. Rotation forest [42] algorithm generates diverse base models, i.e., DT wherein each DT is trained over different feature space generated via principal component analysis (PCA) technique. We propose an ensemble of RVFL in RoF framework (RoF-RVFL) [80], where RVFL has been employed as base model. The proposed methodologies are discussed in the subsequent sections as follows:

5.1 Proposed extended feature RVFL (efRVFL) and an ensemble of efRVFL (en-efRVFL) classifiers

In this section, we discuss the proposed efRVFL and en-efRVFL.

5.1.1 Feature generation method

The feature generation method [73] (see algorithm 5.1) has two parameters μ and ω, where μ is the number of randomly selected features (from X) and ω is the order of design matrix (A). Let $\mu = 2$ and $\omega = 1$ then using the linear regression method, the new features are calculated as follows:

$$A = [1's\ X_i\ X_j], \tag{5.1}$$

$$\upsilon = (A^T A)^{-1} A^T Y, \tag{5.2}$$

$$gf = A\upsilon, \tag{5.3}$$

where X_i and X_j are two original features chosen randomly from X. A and Y represent the design matrix and the class labels vector, respectively. gf is the newly generated feature vector. Similarly, if we take $\mu = 4$ and $\omega = 2$, the matrix A is calculated as follows:

$$A = [1's\ X_i\ X_j\ X_k\ X_l\ X_iX_j\ X_iX_k\ X_iX_l\ X_jX_k\ X_jX_l\ X_kX_lX_i^2\ X_j^2\ X_k^2\ X_l^2].$$

For binary classification problems, Y vector has only 1 and -1 values. One versus all approach is used in multi-class classification problems, i.e., one class is randomly chosen as positive and all other classes are considered as negative. The procedure for computing the extended feature space is given in Algorithm 5.1. The *find*υ function takes input training matrix (X), design matrix order (ω), class labels (Y), randomly selected (μ) feature indexes and returns υ vector of c-dim using standard linear regression method, where, c is measured as $C(\mu + \omega, \omega) = \binom{\mu+\omega}{\omega}$. In the proposed efRVFL model, the original feature space of dimension d is extended analytically upto $2d$ and it is called extended feature space (E).

Algorithm 5.1 Extended feature space [73]

1: Let $X \in \mathbb{R}^{N \times d}$ be the given input dataset and Y contains the target labels.
2: The feature generation method has two parameters μ and ω, where μ is the number of randomly selected features and ω is the order of design matrix A. Initialize μ and ω and let $E = X$.
3: **For** $i = 1, 2, \ldots, \mu$ **do**
4: Generate a vector V containing a random permutation of the integers from 1 to d.
5: **For** $j = 1 : \mu : d - (\mu - 1)$ **do**
6: $P_{i,j,1\cdots\mu}$= choose μ feature indices as $V(j : j + (\mu - 1))$ **Comment:** $P_{i,j}$ have selected μ feature indices for j^{th} new feature.
7: $\upsilon_{i,j,1\cdots c}$ =find$\upsilon(X, \omega, Y, P_{i,j,1\ldots\mu})$ **Comment:** $\upsilon_{i,j,1\ldots c}$ has dim c.
8: Calculate a new feature as $gf = A \ldots \upsilon_{i,j,1..c}$.
9: Generate the extended feature space $E = [E\ gf]$.
10: **EndFor**
11: **EndFor**

5.1.2 Extended feature RVFL (efRVFL) model

Similar to standard RVFL, the proposed efRVFL also has direct links from the input layer to the output layer. Having these direct links, efRVFL gets the capability to deal with linear relationships within the dataset. Most of the data coming from the real world has non-linear complex hidden relationships. Therefore, to capture non-linear relationships, in efRVFL, nonlinear transformation (activation function, Φ) is employed from the input layer to the hidden layer. In the proposed efRVFL model, we construct supervised randomized features from original inputs without having multiple hidden layers. Hence, extended feature space is generated using original features. Finally, the extended feature space is nonlinearly transformed into randomized feature space. Therefore, the proposed efRVFL model uses original features, extended features, and randomized features for data modelling process. Let $\bar{X}$ contains the features constructed via feature generation technique given in Algorithm 5.1, and if μ divides d then atmost d new features can be added into the original feature space. In this case, let $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \in \mathbb{R}^d$ and $\bar{x}_i = [x_{i(d+1)}, \ldots, x_{i(2d)}] \in \mathbb{R}^d$, where $\bar{x}_i$ is the newly generated features vector corresponding to x_i input vector. The extended feature space is defined as:

$$E = [X\ \bar{X}]_{N \times 2d}, \tag{5.4}$$

let $\hat{x}_i = [x_i \;\; \bar{x_i}] \in \mathbb{R}^{2d}$ be the extended feature vector corresponding to x_i vector and here,

$$\bar{X} = \begin{bmatrix} x_{1(d+1)} & \dots & x_{1(2d)} \\ \vdots & \ddots & \vdots \\ x_{N(d+1)} & \dots & x_{N(2d)} \end{bmatrix}_{N \times d} \tag{5.5}$$

The proposed efRVFL network i.e., $g : \mathbb{R}^{2d} \rightarrow \mathbb{R}^m$ is defined as follows:

$$g(\hat{x_i}) = \sum_{k=1}^{d} \hat{\beta}_k x_{ik} + \sum_{k=d+1}^{2d} \hat{\beta}_k x_{ik} + \sum_{k=2d+1}^{2d+L} \hat{\beta}_k \Phi(\xi_k \cdot \hat{x_i} + \delta_k), \text{ for all } i. \tag{5.6}$$

In the proposed efRVFL model, the hidden layer output matrix is calculated as:

$$\hat{H} = [X \; \bar{X} \; H_0], \tag{5.7}$$

where

$$H_0 = \begin{bmatrix} \Phi(\xi_1 \cdot \hat{x}_1 + \delta_1) & \dots & \Phi(\xi_L \cdot \hat{x}_1 + \delta_L) \\ \vdots & \ddots & \vdots \\ \Phi(\xi_1 \cdot \hat{x}_N + \delta_1) & \dots & \Phi(\xi_L \cdot \hat{x}_N + \delta_L) \end{bmatrix}_{N \times L}, \tag{5.8}$$

$\hat{H} \in \mathbb{R}^{N \times (2d+L)}$ and H_0 is the matrix of non-linear transformed features calculated via hidden layer with input as extended feature space. The optimization problem of the proposed efRVFL model is formulated as follows:

$$\min_{\hat{\beta} \in \mathbb{R}^{(2d+L) \times m}} \frac{1}{2} \left\| \hat{H}\hat{\beta} - Y \right\|_2^2 + \alpha \frac{1}{2} \left\| \hat{\beta} \right\|_2^2, \tag{5.9}$$

The final parameter $\hat{\beta}$ is calculated via closed form solution given in Chapter (2).

5.1.3 Ensemble of extended feature RVFL (en-efRVFL) model

Ensemble learning improves the stability, robustness, and generalization capability of a model. The proposed efRVFL model is an unstable classifier. Therefore, we propose

an ensemble of efRVFL (en-efRVFL). The diagram of the proposed en-efRVFL is given in Figure 5.1. The aforementioned feature generation method uses randomization process (as vector V) for calculating the new (additional) features. Hence, each efRVFL model is trained over different extended feature spaces. This mechanism generates diverse and accurate base models (efRVFL).

Algorithm 5.2 Ensemble method of extended feature RVFL

1: **Given:** Let X be the training dataset of order $N \times d$ and Y contains the class labels. Here, N and d are the number of samples and the number of features, respectively. Let $\bar{T}$ be the number of base classifiers in an ensemble and g_i denotes the base model (efRVFL). Initialize the value of μ and ω that is the parameters of the feature generation method.
2: **Training Phase**
3: **For** $i = 1, 2, \ldots, \bar{T}$ **do**
4: Construct the extended feature space (E_i) via Algorithm 5.1 for base model (g_i).
5: Train the base model g_i over E_i.
6: **EndFor**
7: **Classification phase**
8: **For** $i = 1, 2, \ldots, \bar{T}$ **do**
9: A new sample x is given as the input to each base model (g_i).
10: **EndFor**
11: The final decision of the ensemble is taken by the average voting scheme.

Let g_j be the base model (efRVFL) and T be the number of base models in the proposed en-efRVFL. As output, each (g_j) gives the class probabilities. For a given sample x, the output of each g_j is an m-dim label vector $(g_j^1, g_j^2, \ldots, g_j^m)$, where g_j^i is the output of g_j for the class label c_i. The ensemble model (F) takes its decision on the basis of the averaging voting scheme. The pseudocode for constructing the proposed en-efRVFL is given in Algorithm 5.2.

5.1.4 Computational complexity

Let (X, Y) be the training set, where $X \in \mathbb{R}^{N \times d}$ and $Y \in \mathbb{R}^{N \times m}$. Here, N and d are the number of samples and number of features, respectively, and the number of classes is m. Suppose each layer has L number of hidden neurons. In deep RVFL models, let L_0 be the number of layers. In all RVFL-based networks, the computation of matrix inverses is involved in calculating the output layer weights. Hence, the complexity of the models is

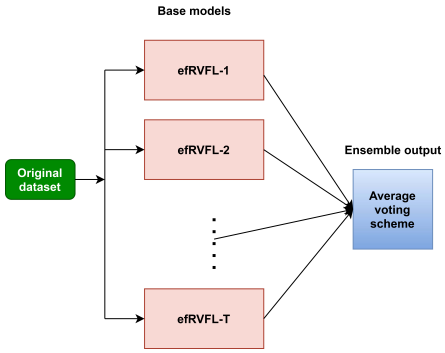


Figure 5.1: Architecture of the proposed en-efRVFL model.

determined by the size of the matrices to be inverted. By the standard procedure, $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ time are required to calculate the inverse a matrix of size $N \times N$ [337]. The standard RVFL model solves the output layer weights either in the primal or dual, which results in reducing the complexity of the model by choosing among the $min(N, d + L)$. Thus, the time complexity of RVFL model is either $\mathcal{O}(N^3)$ or $\mathcal{O}((d+L)^3)$. The complexity of dRVFL is either $\mathcal{O}(N^3)$ or $\mathcal{O}((d + LL_0)^3)$ whichever results in lower computation. In the edRVFL model, the matrix inversions are required which are of the size $N \times N$ or $((L + d) \times (L + d))$ for the first layer and $((2L + d) \times (2L + d))$ for the higher layer RVFL models. In the proposed models, we need to calculate the extended feature space wherein atmost d new feature can be generated. For each new feature, we need to take the inverse of a square matrix of order c. Therefore, the computational complexity of calculating d features is equal to $\mathcal{O}(dc^3)$. The computational complexity of the proposed efRVFL model is either $\mathcal{O}(dc^3) + \mathcal{O}(N^3)$ or $\mathcal{O}(dc^3) + \mathcal{O}((2d + L)^3)$ depending on whether the parameters are being calculated in primal or dual space, respectively. In the proposed en-efRVFL model, we have $\bar{T}$ number of base models. Therefore, it has either $\mathcal{O}(\bar{T}dc^3) + \mathcal{O}(\bar{T}N^3)$ or $\mathcal{O}(\bar{T}dc^3) + \mathcal{O}(\bar{T}(2d + L)^3)$ complexity.

5.1.5 Experiments

In this section, we give the details of the datasets and discuss the experimental results of the proposed efRVFL and en-efRVFL on the UCI dataset, sparse datasets and NDC dataset [355]. Moreover, we perform the statistical analysis of the models.

Table 5.1: The statistics of the UCI classification datasets.

Datasets	#Samples	#Attributes	#Classes
adult	48,842	15	2
car	1728	7	4
hill valley	1212	101	2
steel plates	1941	28	7
cardiotocography 3clases	2126	22	10
connect 4	67557	43	2
contrac	1473	10	3
abalone	4177	9	3
image segmentation	2310	19	7
cardiotocography 10clases	2126	22	10
led display	1000	8	10
miniboone	130,064	51	2
pt shape	1600	65	100
musk 2	6598	167	2
wq white	4898	9	7
chess krvkp	3196	37	2
bank	4521	17	2
nursery	12,960	9	5
om states 2f	1022	26	2
waveform noise	5000	41	3
optical	5620	63	10
chess krvk	28,056	7	18
ozone	2536	73	2
letter	20,000	17	26
page blocks	5473	11	5
pendigits	10,992	17	10
om nucleus 4d	1022	42	2
pt margin	1600	65	100
ringnorm	7400	21	2
magic	19020	11	2
spam base	4601	58	2
statlog german-credit	1000	25	2
molec biol splice	3190	61	3
statlog image	2310	19	7
statlog landsat	6435	37	6
statlog shuttle	58000	10	7
thyroid	7200	22	3
titanic	2201	4	2
pt texture	1600	65	100
yeast	1484	9	10
twonorm	7400	21	2
wall following	5456	25	4
waveform	5000	22	3
wq red	1599	12	6
mushroom	8124	22	2
semeion	1593	257	10

Here, om, pt and wq denotes oocytes merluccius, plant and Wine-quality, respectively.

5.1.5.1 Datasets

We compare the proposed efRVFL and en-efRVFL models over UCI datasets [267] against several FNN models [11, 45, 268, 356, 357, 358, 359, 360] and deep RVFL classifiers [57]. We performed the experiments on 46 UCI datasets that are used in [57]. These datasets come from diverse domain such as biology, physics, etc. Table 5.1 shows the statistics of these publicly available classification datasets. To further evaluate the effectiveness of the proposed models, we conducted the experiments on 12 sparse datasets used in [57]. These sparse datasets have high feature dimensions. The technical details of the sparse datasets are given in Table 5.2.

Table 5.2: The sparse dataset details.

Dataset	#Samples	#Attributes	#Classes
RELATHE	1427	1000	2
bbc	2225	1000	5
bbcsport	737	1000	5
hill-valley	1212	101	2
musk-1	476	167	2
arrhythmiả	452	263	13
BASEHOCK	1993	1000	2
musk-2	6598	167	2
low-res-spect	531	101	9
RCV1	9625	1000	4
semeion	1593	257	10
TDT2	9394	1000	30

5.1.5.2 Experimental setup details

All the experiments have been done on the PC with Window 10 and 16-GB RAM and MATLAB R2017a and Intel(R) Core(TM) i7 CPU @ 3.00GHz processor. For the proposed efRVFL and en-efRVFL models, the number of hidden nodes is chosen from the range $\{256, 512, 1024\}$ for those datasets whose samples are less than $10,000$ and for the rest datasets the hidden nodes are chosen from the range $\{1024, 2048, 4096\}$. The regularization parameter α for the proposed models is set to be $1/C$, where C is chosen from the range $\{2^{-6}, 2^{-4}, \cdots, 2^{12}\}$. The number of base models in the proposed en-efRVFL model is set to be 10. We used three activation functions, i.e., selu, relu and sigmoid function. The two stage hyperparameter tuning method given in [57] is employed in experiments. Following [73], we have also chosen the hyper-parameters as $\mu = 4$ and $\omega = 2$ in the feature generation method.

5.1.5.3 Experiments analysis over UCI datasets

We conducted the experiments over 46 UCI datasets. The classification performances of the compared models, i.e., Self Normalizing Neural Network (SNN), MSRAinit (MS), WeightNorm (WN) Network, Highway (HW) Network, LayerNorm (LN) Network, Batchnorm (BN) Network, ResNet Network, RVFL, Deep RVFL(dRVFL), Ensemble of deep RVFL (edRVFL) and the proposed efRVFL and en-efRVFL models are shown in Table 5.3. One can observe that the proposed en-efRVFL model has win-

Table 5.3: Classification accuracy (%) of the models on UCI datasets.

Dataset	MS* [356]	HW* [45]	ResNet* [357]	BN* [358]	WN* [359]	LN* [360]	SNN* [268]	RVFL* [11]	dRVFL* [57]	edRVFL* [57]	efRVFL†	en-efRVFL†
abalone	62.84	64.27	64.66	63.03	63.51	61.78	**66.57**	66.12	66.33	65.81	66.38	66.52
adult	84.87	84.53	84.84	84.99	84.53	85.17	84.76	**85.25**	85.12	**85.25**	85.09	85.21
bank	88.76	88.85	87.96	88.23	88.5	89.2	89.03	89.71	89.87	89.6	**90.09**	90.02
car	98.61	95.6	92.82	96.06	97.69	**99.07**	98.38	97.74	97.97	98.04	98.03	98.38
cardiotocography-10clases	84.18	84.56	81.73	79.1	**86.06**	83.62	83.99	82.82	83.62	83.24	83.76	84.75
cardiotocography-3clases	89.64	91.71	90.21	90.96	89.45	90.21	91.53	92.37	92.84	92.66	92.7	**93.08**
chess-krvk	86.06	52.55	85.43	87.81	76.73	**89.38**	88.05	57.82	68.39	70.07	67	67.65
chess-krvkp	99	99	99.12	98.62	99.12	98.75	99.12	98.84	99.03	99.12	98.72	**99.41**
connect-4	88.31	85.99	87.16	87.29	88.33	**88.56**	88.07	81.15	83.94	84.77	85.52	86.81
contrac	51.36	50.54	51.36	45.38	47.55	45.92	51.9	52.85	**55.37**	54.01	52.85	53.13
hill-valley	51.16	50	53.96	50.5	49.34	50.5	52.48	**69.64**	58.75	67	65.84	63.7
image-segmentation	90.9	90.24	89.19	84.81	89.38	88.38	**91.14**	88.33	89.1	88.52	90.52	54.05
led-display	72	70.4	71.6	62.8	69.2	64.8	**76.4**	74.7	73.9	74.4	74.6	74.9
letter	97.12	89.84	97.62	**97.96**	95.8	97.42	97.26	94.55	97.23	97.46	97.15	97.69
magic	86.29	86.73	**87.23**	87.13	86.9	86.2	86.92	86.17	86.55	86.81	86.81	86.97
minibone	92.5	92.7	92.54	92.62	92.72	93.13	93.07	92.14	92.33	92.72	93.81	**94.29**
molec-biol-splice	84.82	88.33	85.57	85.19	84.94	86.07	**90.09**	82.56	82.34	84	89.15	90.06
mushroom	**100**	**100**	**100**	99.9	99.5	99.5	**100**	**100**	**100**	**100**	**100**	**100**
musk-2	99.45	99.15	99.64	**99.82**	99.27	99.51	98.91	98.29	99.04	98.57	98.64	99.39
nursery	99.88	**100**	99.4	99.4	99.66	99.66	99.78	97.92	98.32	98.7	99	99.24
om nucleus_4d	81.96	71.76	80	80.78	80.78	76.86	82.35	85.1	82.75	84.41	84.61	**85.98**
om_states_2f	94.9	94.9	93.73	93.33	90.2	94.12	**95.29**	91.96	91.86	93.63	93.04	93.63
optical	96.66	96.44	96.27	97.16	96.38	97.55	97.11	98.05	98.16	**98.27**	96.99	**98.27**
ozone	97.32	97.16	96.69	96.69	**97.48**	97.16	97	97.04	97.12	97.2	97.08	97
page-blocks	97.08	96.56	96.05	96.13	**97.3**	97.08	95.83	96.49	96.6	96.56	96.29	96.27
pendigits	97.14	96.71	97.08	97.34	96.2	96.57	97.06	97.23	97.77	97.46	97.6	**98**
pt-margin	81.25	83.75	79.75	76	81.75	84.25	81.25	81.81	81.88	81.88	82.81	**84.56**
pt-shape	63.5	63.25	51.5	28.5	65.75	67.75	**72.75**	67.63	71.75	72.31	69.56	71.81
pt-texture	79	79	80	82	81.75	84.25	81.25	83.88	84.06	85.25	83.94	**85.69**
ringnorm	98.43	96.92	98.11	98.43	97.19	98.27	97.51	97.14	98.26	97.97	98.24	**98.49**
semeion	92.96	**94.47**	91.46	93.72	93.22	**94.47**	91.96	92.9	93.47	92.96	91.71	93.91
spambase	94.61	94.35	94.61	94.26	95.04	**95.13**	94.09	93.39	93.61	93.87	93	93.98
statlog-german-credit	72.8	77.6	77.2	75.2	74	74	75.6	**78.8**	76.2	77.7	77.8	78.4
statlog-image	**97.57**	95.84	95.84	96.71	95.15	**97.57**	95.49	96.71	96.79	96.84	96.71	97.18
statlog-landsat	90.75	91.1	90.55	90.4	89.25	90.4	91	90.25	90.7	**91.2**	90.2	90.95
statlog-shuttle	99.83	99.77	99.92	99.88	99.88	99.87	99.9	99.87	99.88	99.91	99.92	**99.93**
steel plates	75.67	76.08	76.29	70.31	**78.56**	75.88	78.35	75.52	58.66	76.44	75.98	75.98
thyroid	97.7	97.08	97.99	97.78	98.07	97.52	**98.16**	95.45	95.92	95.65	95.51	96.18
titanic	79.09	**79.27**	77.27	78	78.18	78.91	78.36	78.82	78.82	78.82	78.55	78.55
twonorm	97.78	97.08	97.35	97.57	97.3	97.24	**98.05**	97.65	97.78	97.81	97.73	97.81
wall-following	90.76	92.3	92.23	**93.33**	92.74	91.28	90.98	88.56	90.71	90.3	91.75	**93.33**
waveform	83.12	83.2	83.6	83.6	83.76	84.48	84.8	**86.7**	86.34	86.44	86.06	86.58
waveform-noise	83.28	**86.96**	85.84	84.8	86.4	85.04	86.08	86.38	86.32	85.7	86.26	86.86
wq-red	62.5	56.25	61.5	54.5	55.75	61	63	63	61.75	**65.63**	62.63	63.69
wq-white	64.79	55.64	63.07	53.35	54.82	**65.44**	63.73	58.48	62.36	63.3	57.9	62.64
yeast	61.73	60.65	54.99	49.06	58.76	60.92	**63.07**	61.73	60.24	61.66	60.58	60.18

The results of * models are taken from [57].
†denotes the proposed models and bold face denotes best results.

ning performance over cardiotocography-3clases (93.08%), chess-krvkp (99.41%), minibone (94.29%), oocytes_merluccius_nucleus_4d (85.98%), pendigits (98%), plant-margin (84.56%), plant-texture (85.69%), ringnorm (98.49%), statlog-shuttle (99.93%) datasets. The best classification performances on each data are highlighted in bold.

Furthermore, we perform the Friedman rank test [269] employed in [267] to assess the performance of each model.

The average rank of compared models, the proposed efRVFL and en-efRVFL models are given in Table 5.4. It can be observed that the proposed en-efRVFL model has the smallest rank that is equal to 3.91. The lower rank indicates that the proposed en-efRVFL model has the winning performance among all the compared models over maximum datasets. The ensemble of deep RVFL (edRVFL) model has the second lowest rank that is equal to 5.3 and then followed by SNN that has a rank 5.4. The proposed efRVFL model has the fourth position with a rank 6.23 among 12 models. It shows that the proposed shallow efRVFL

Table 5.4: Statistical comparison (on UCI datasets) between en-efRVFL and other baseline classification models.

Model	Average rank	p-value
en-efRVFL†	3.91	
edRVFL [57]	5.3	0.07346
SNN [268]	5.4	0.13622
efRVFL†	6.23	0.0001
LN [360]	6.26	0.02382
dRVFL [57]	6.45	0.00001
MS [356]	6.52	0.0164
HW [45]	7.28	0.00012
RVFL [11]	7.38	0.00001
ResNet [357]	7.53	0.00018
WN [359]	7.6	0.00064
BN [358]	8.13	0.0001

The p-value is calculated from paired Wilcoxon test.
Low rank shows better performance.
† denotes the proposed models.

Table 5.5: Statistical evaluation of RVFL based classification models on UCI datasets.

Model	Average rank	p-value
en-efRVFL†	1.88	
edRVFL [57]	2.65	0.07346
dRVFL [57]	3.25	0.00001
efRVFL†	3.34	0.0001
RVFL [11]	3.88	0.00001

The p-value is calculated from paired Wilcoxon test.
Low rank shows better performance.
† denotes the proposed models.

model with extended features is performing better than many other deep models such as ResNet, HW, LN, MS, WN, BN and dRVFL, and standard RVFL as well. The experimental results show that the extended feature space gives more information to the efRVFL model as compared to the original feature space.

We also perform the Wilcoxon test and calculate p-values to discover the fact that whether there exists any significant difference between the best-performing model and other given models. It shows that except edRVFL and SNN, the proposed en-efRVFL model significantly outperforms the other compared models, i.e., HW, LN, BN, MS, RVFL, ResNet, dRVFL, and WN.

Table 5.6: Statistical evaluation of classification models on UCI datasets.

	en-efRVFL†	edRVFL [57]	SNN [268]	efRVFL†	LN [360]	dRVFL [57]	MS [356]	HW [45]	RVFL [11]	ResNet [357]	WN [359]	BN [358]
en-efRVFL†	0	0	0	0	0	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$
edRVFL	0	0	0	0	0	0	0	0	0	0	0	$s+$
SNN	0	0	0	0	0	0	0	0	0	0	0	$s+$
efRVFL†	0	0	0	0	0	0	0	0	0	0	0	0
LN	0	0	0	0	0	0	0	0	0	0	0	0
dRVFL	$s-$	0	0	0	0	0	0	0	0	0	0	0
MS	$s-$	0	0	0	0	0	0	0	0	0	0	0
HW	$s-$	0	0	0	0	0	0	0	0	0	0	0
RVFL	$s-$	0	0	0	0	0	0	0	0	0	0	0
ResNet	$s-$	0	0	0	0	0	0	0	0	0	0	0
WN	$s-$	0	0	0	0	0	0	0	0	0	0	0
BN	$s-$	$s-$	$s-$	0	0	0	0	0	0	0	0	0

Here, $s+$ means the model in the corresponding row is statistically better than the model in the corresponding column and $s-$ mean the model in the corresponding row is statistically worse than the model in the corresponding column.
† denotes the proposed models.

We also compared RVFL-based models and calculated their Friedman rank and p-value. The corresponding results are shown in Table 5.5. One can observe that in this family, the proposed en-efRVFL model has the smallest average rank that is equal to 1.88 which means that, en-efRVFL has winning performance in this family. The edRVFL model with rank 2.65 is followed by dRVFL, the proposed efRVFL and RVFL with rank 3.25, 3.34, and 3.88, respectively. Moreover, Table 5.6 gives the pairwise statistical comparison of the proposed efRVFL and en-efRVFL models and the other baseline models. Here, zero entries denote that no significant difference exists between the models in the row and column of the corresponding cell, $s+$ and $s-$ denote that a significant difference exists among the models in the row and column of the corresponding cell with the row model performing better and worse compared to the column model, respectively. The proposed en-efRVFL model is significantly better than dRVFL, MS, HW, RVFL, ResNet, WN and BN deep models.

5.1.5.4 Experiments analysis over sparse datasets

As SNN and RVFL-based models perform well on UCI datasets, we compare these models with the proposed en-efRVFL and efRVFL model over 12 sparse datasets. The classification performances of these models are shown in Table 5.7. One can observe that the proposed en-efRVFL model outperforms on 9 out of 12 datasets among all compared models. The proposed en-efRVFL model has highest average accuracy that is equal to 99.39 $\%$ over musk-2 dataset and second highest performance with average accuracy 97.56 $\%$ over bbcsport dataset. As compare to SNN model, the proposed en-efRVFL model wins over 11 datasets out of 12 datasets. Similar to previous calculations, we employ the Friedman rank

test and Wilcoxon test to check whether there is any significant difference between the proposed en-efRVFL model and other models.

Table 5.7: Evaluation of FNN-based classification models over the sparse datasets.

Dataset	SNN [268]	RVFL [11]	dRVFL [57]	edRVFL [57]	efRVFL†	en-efRVFL†
arrhythmia	65.49	**71.46**	70.58	69.91	70.8	70.8
BASEHOCK	**98.03**	96.59	96.39	97.04	94.63	96.39
bbc	96.06	96.63	96.49	96.81	96.23	**97.08**
bbcsport	92.43	97.28	97.28	97.15	97.15	**97.56**
hill-valley	52.48	**69.64**	58.75	67	63.37	63.7
low-res-spect	85.71	89.85	88.35	89.66	90.6	**91.54**
musk-1	87.39	80.67	83.82	88.24	88.66	**89.71**
musk-2	98.91	98.29	99.04	98.57	98.82	**99.39**
RCV1	69.62	93.21	93.08	93.18	92.73	**94.02**
RELATHE	84.07	86.62	86.05	86.33	85.7	**88.51**
semeion	91.96	92.9	93.47	92.96	91.33	**93.91**
TDT2	78.07	96.33	96.22	96.37	96.07	**96.56**

Table 5.8: Statistical comparison (on sparse datasets) of en-efRVFL and other RVFL models.

Model	Average rank	p-value
en-efRVFL†	1.58	
RVFL [11]	3.04	0.08364
edRVFL [57]	3.21	0.07186
dRVFL [57]	3.83	0.00338
efRVFL†	4.25	0.0038
SNN [268]	5.08	0.0048

The p-value is calculated from paired Wilcoxon test.
Low rank shows better performance.
† denotes the proposed models.

Table 5.8 shows the average rank and p values of the compared models. The proposed en-efRVFL model has the smallest rank, i.e., 1.58, among all the compared models. The proposed efRVFL with average rank 4.25 is also performing better than deep SNN model that has the average rank 5.08. Except RVFL and edRVFL, the proposed en-efRVFL model outperforms significantly than dRVFL, efRVFL and SNN models. The proposed en-efRVFL model has the average accuracy, 97.56 % and 91.54 % over small sparse dataset such as bbcsport, low-res-spect, respectively. The proposed efRVFL and en-efRVFL models have overall good performance on small and large datasets.

5.2 Proposed rotated random vector functional link neural network (RoF-RVFL) classifier

In this section, we discuss the proposed rotated random vector functional link neural network (RoF-RVFL) ensemble classifier. Here, RVFL is employed as base classifier in rotation forest [42]. Let F be the attribute set of the original data set and it is split randomly into K subsets of attributes, where every subset contains ($M = \frac{d}{K}$) features (let d is divisible by K). Now, PCA is employed to each subset of attributes and constructs the full attribute set for every model in the ensemble. All principal components derived through PCA are taken to keep all the information of the dataset. Using these principal components make a sparse rotation matrix R_i as follows:

$$R_i = \begin{bmatrix} b_{i,1}^{(1)}, \ldots, b_{i,1}^{(M_1)} & 0 & \cdots & 0 \\ 0 & b_{i,2}^{(1)}, \ldots, b_{i,2}^{(M_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_{i,K}^{(1)}, \ldots, b_{i,K}^{(M_K)} \end{bmatrix}. \tag{5.10}$$

Correspond to the original features, columns of R_i are rearranged to R_i^*. Let $f_1, f_2, ..., f_{\bar{T}}$ be the RVFL classifiers in the proposed RoF-RVFL model. Calculate the hidden-layer output matrix H for each base model f_i as:

$$H = \begin{bmatrix} \phi(w_1.(x_1 R_i^*) + z_1) & \ldots & \phi(w_L.(x_1 R_i^*) + z_L) & x_1 R_i^* \\ \vdots & \ddots & \vdots & \vdots \\ \phi(w_1.(x_N R_i^*) + z_1) & \ldots & \phi(w_L.(x_N R_i^*) + z_L) & x_N R_i^* \end{bmatrix}. \tag{5.11}$$

where x_i and $\phi(.)$ denote the input pattern vector and the activation function, respectively; and w_i and z_i are the hidden layer weights and biases, respectively, with $i = 1, ..., L$. Each f_i assign probability $f_{i,\sigma}$ for every class, $\sigma = 1, ..., m$. Let x be a test sample, using the

averaged method, measure the confidence for every class A_σ as:

$$F^*_\sigma(x) = \frac{1}{\bar{T}} \sum_{i=1}^{\bar{T}} f_{i,\sigma}(xR^*_i), \sigma = 1, ..., m. \tag{5.12}$$

Finally, x will be assigned a class with the maximum confidence. The pseudocode of the proposed RoF-RVFL model is given in Algorithm 5.3.

Algorithm 5.3 RoF-RVFL

Training Phase

- X: The training dataset $(N \times d$ matrix); Y: Target vector containing the class labels; $\bar{T}$: The number of classifiers in the ensemble; K: The number of attribute subsets
- $\{A_1, ..., A_m\}$: the set of class labels

For $i = 1, ..., \bar{T}$

- Prepare the rotation matrix R^*_i:
  - Split the feature set F into K subsets $F_{i,t}$ $(t = 1, ..., K)$, randomly.
  - **For** $t = 1, ..., K$
    - Choose the columns of X according to the features in $F_{i,t}$ to compose a submatrix $X_{i,t}$.
    - Draw a bootstrap sample $X'_{i,t}$ (the size of this set is generally smaller than that of $X_{i,t}$) from $X_{i,t}$.
    - Apply PCA on $X'_{i,t}$ to get a matrix $B_{i,t}$ whose j^{th} column contains the coefficients of the j^{th} principal component.
  - Arrange the matrices $B_{i,t}$ $(t = 1, ..., K)$ in a rotation matrix R_i as in Eq. (5.10).
  - Rearrange the columns of R_i to construct R^*_i corresponding to the original features in the set F.
- Use (XR^*_i, Y) as the training set to train RVFL classifier (f_i).

Classification phase

- For a new sample x, measure the confidence for every class as in Eq. (5.12) and refer x to the class with the maximum confidence.

Table 5.9: Accuracies (%) of the classification models corresponding to different datasets.

Datasets	RVFL [114]	RaF [40]	RoF [42]	RoF-RVFL
abalone	64.61	64.2	64.92	65.01
balance-scale	89.74	86.86	86.86	90.38
blood	77.67	76.07	76.34	77.94
breast-cancer-wisc	97.29	97.29	97.29	97.29
breast-cancer-wisc-prog	80.1	76.02	82.65	81.12
congressional-voting	61.7	61.01	59.86	61.7
echocardiogram	84.85	83.33	84.09	85.61
ecoli	87.2	87.5	87.8	88.69
fertility	85	88	87	87
haberman-survival	74.34	71.71	67.11	75
heart-hungarian	83.22	82.88	83.9	85.27
ilpd-indian-liver	73.29	70.89	74.32	73.46
iris	95.95	95.95	96.62	96.62
led-display	72.2	73	71.5	73.9
libras	84.44	77.5	86.39	85.28
lung-cancer	46.88	46.88	53.13	50
mammographic	81.98	80.83	78.96	83.13
molec-biol-promoter	81.73	83.65	91.35	87.5
oocytes_merluccius_nucleus_4d	83.43	77.94	83.14	84.71
oocytes_merluccius_states_2f	91.57	91.47	92.55	92.06
oocytes_trisopterus_nucleus_2f	80.7	80.59	82.35	82.79
ozone	97.08	97.12	97.12	97.16
parkinsons	88.27	87.24	90.82	88.78
pima	75.65	75.52	75.78	75.91
pittsburg-bridges-MATERIAL	90.38	91.35	92.31	92.31
pittsburg-bridges-SPAN	69.57	63.04	66.3	70.65
pittsburg-bridges-TYPE	67.31	65.38	68.27	67.31
planning	71.11	71.11	70	71.11
post-operative	72.73	71.59	67.05	72.73
seeds	96.15	92.31	93.27	95.19
statlog-australian-credit	65.26	65.26	64.1	67.73
statlog-german-credit	76.9	78.4	77	78.2
statlog-heart	88.06	85.82	84.33	88.06
statlog-vehicle	77.61	75.83	78.79	80.21
synthetic-control	96	98.33	98.83	98.33
trains	75	87.5	87.5	87.5
twonorm	97.85	96.66	97.77	97.85
vertebral-column-2clases	83.44	83.44	84.74	85.71
vertebral-column-3clases	82.79	85.06	82.14	83.77
waveform	84.92	84.1	85.66	85.98
waveform-noise	86.52	84.94	85.4	86.46
zoo	96	98	99	98
Average Accuracy	81.34	80.75	81.77	82.7
Average Rank	2.8	3.21	2.44	1.55

5.2.1 Experimental results and analysis

In this subsection, experimental analysis is done to illustrate the performance of the proposed RoF-RVFL. All the experiments are conducted on MATLAB-2017b on a PC with 8 GB RAM Intel(R) Core(TM) i7-6700 CPU 3.41GHz. We conduct the experiments over UCI [342] datasets and the detailed information about all datasets are given in [267]. In Table 5.9, the average accuracy and the average rank of the proposed RoF-RVFL model, RVFL, RaF and RoF are shown.

5.2.2 Experimental setup

The parameters range in RVFL is, $L = 3 : 203$ and $\lambda = (\frac{1}{2})^C$, where $C = -5 : 14$. The sigmoid function is used as an activation function. The random parameters (Weights and biases) are chosen randomly from [-1,1] and [0,1], respectively. And we fix the ensemble size $\bar{T} = 50$. We use the same parameter setting for rotation forest (RoF) as in [361].

5.2.3 Statistical tests

To illustrate the statistical significance among the proposed RoF-RVFL and the existing algorithms, i.e., RaF, RoF and RVFL, we use the Friedman test [362].

Table 5.10: Nemenyi post hoc test: Here, numbers mean there is a significant difference between the row algorithm and the column algorithm. The row models are better than the column models.

	RVFL	RaF	RoF	RoF-RVFL
RVFL				
RaF				
RoF		0.77		
RoF-RVFL	1.25	1.67	0.89	

With simple calculation, we obtain $\chi_F^2 = 38.0357$, and $F_F = 17.7284$. With $q = 4$ and $n = 42$ datasets, F_F is distributed with 3 and 123 degrees of freedom. With $\alpha = 0.05$, the critical value of $F_{(3,123)} = 2.67$. Hence, there are significance difference among the models

as the null hypothesis is rejected. After calculation,

$$\begin{aligned} CD &= 2.5690 * \sqrt{(3*4)/(6*42)}, \\ &= 0.7237. \end{aligned} \tag{5.13}$$

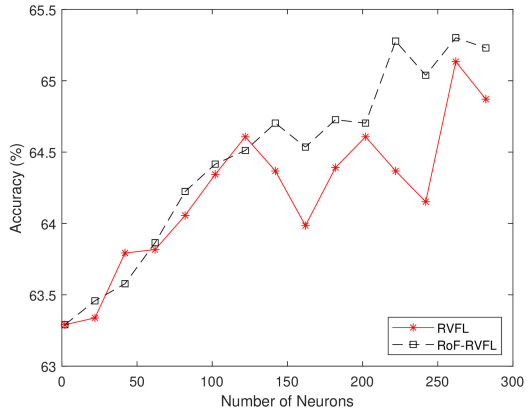
Moreover, one can see the detailed information about the Nemenyi test in Table 5.10. To evaluate the performance of the existing models and the proposed RoF-RVFL model in pairwise manner, we use pairwise sign test. From Table 5.11, one can notice that RoF-RVFL is significantly better than RVFL, RaF and RoF algothims. RoF is statistically better than RaF model.

Table 5.11: Pairwise win tie loss: x-y-z means that row model win x-times, ties y-times and loses z-times with respect to the column model.

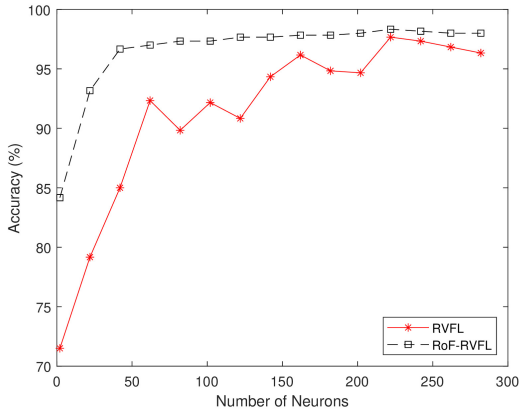
	RVFL	RaF	RoF	RoF-RVFL
RaF	11-6-25			
RoF	24-1-17	27-3-12		
RoF-RVFL	34-6-2	34-5-3	27-5-10	

5.2.4 Parameter sensitivity

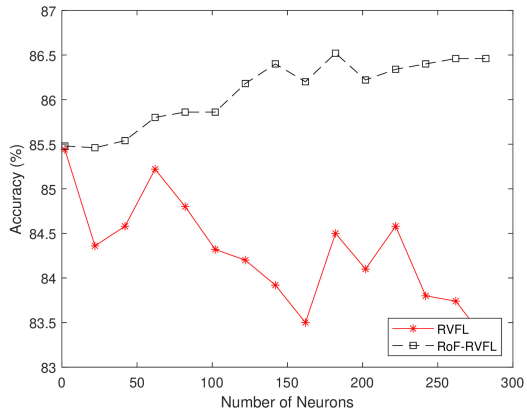
In this section, we analyze the effect of parameter L (the number of neurons in the hidden layer) over the performance of the proposed model (RoF-RVFL) and RVFL. In Figure 5.2a and 5.2b, one can observe that as the number of neurons is increasing, the performance of RVFL and RoF-RVFL increases. From Figure 5.2c and 5.2d, we can observe the performance of RVFL and RoF-RVFL with number of neurons are reversed, i.e. for large enough neurons, the performance of RVFL is worse but RoF-RVFL has better performance.
In Figure 5.2e, we can notice that the performance of RVFL is fluctuating with neurons and the performance of RoF-RVFL is smooth. In Figure 5.2f, RVFL outperforms RoF-RVFL over the statlog-heart dataset. As one can see the performance of the models is sensitive to the parameters so one needs to select the appropriate value of the parameters.



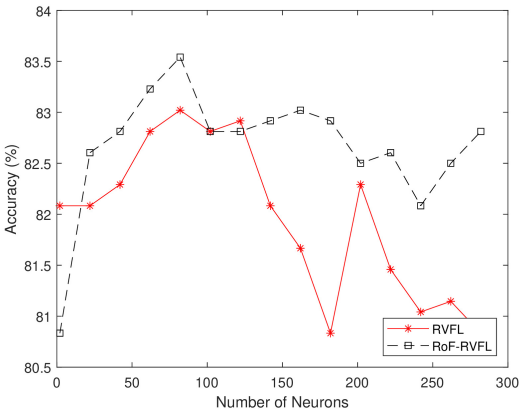
(a) Abalone



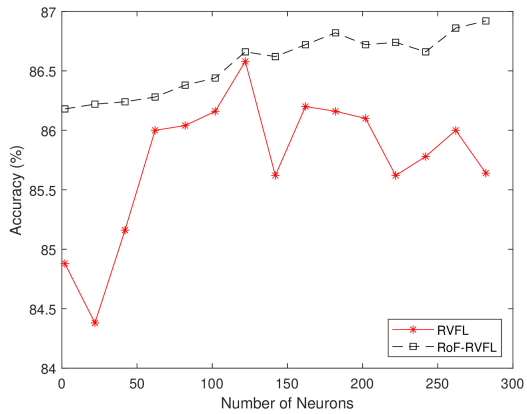
(b) Synthetic-control



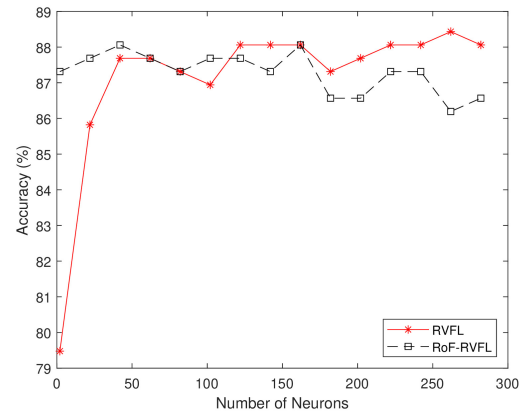
(c) Waveform



(d) Mammographic



(e) Waveform-noise



(f) Statlog-heart

Figure 5.2: The classification performance of the proposed RoF-RVFL and RVFL models with varying number of hidden neurons.

5.3 Experimental results analysis of the proposed models on ADNI dataset

To further evaluate the performance of the proposed models in Chapter 5, we conducted the experiment with the ADNI dataset, and the corresponding results are shown in Table 5.12. The results demonstrate that the proposed en-efRVFL has superior performance at 78.47%, followed by RoF-RVFL at 75.23%. The results demonstrate that the proposed ensemble en-efRVFL model has 3 wins with 0 losses and the lowest average rank at 1. Apart from these results, we compare the models proposed in Chapter 5 with the models proposed in Chapters 3 and 4. The corresponding results are Table 5.13. One can observe that among all proposed shallow models, i.e. IFRVFL, GE-IFWRVFL, CP-FRVFL, ext-LSTSVM, ext-RELS-TSVM, NPRVFL, and efRVFL, the proposed NPRVFL has the best performance at 76.78% and the proposed efRVFL has the worst performance at 70.59%. The proposed CP-FRVFL has the second position among shallow models. Moreover, the proposed ensemble en-efRVFL model has the highest average accuracy at 78.47 and the lowest average rank at 1.33. One can also observe that all the proposed models except efRVFL and ext-LSTSVM have superior performance to the existing RVFL model on the ADNI dataset.

Table 5.12: Experimental results of the RVFL and the proposed models, i.e. efRVFL, Rof-RVFL, and en-efRVFL on the ADNI dataset.

	RVFL [110]	efRVFL	RoF-RVFL	en-efRVFL
CN_vs_AD	0.8903	0.8831	0.8968	0.9127
CN_vs_MCI	0.6295	0.6135	0.6878	0.7407
MCI_vs_AD	0.6621	0.6211	0.6723	0.7006
Average AUC	0.7273	0.7059	0.7523	0.7847
Average Rank	3	4	2	1
Overall win tie loss	$[0, 0, 0]$	$[0, 0, 3]$	$[0, 0, 0]$	$[3, 0, 0]$

Table 5.13: Experimental results of the RVFL and the proposed models, i.e. IFRVFL, GE-IFWRVFL, CP-FRVFL, ext-LSTSVM, ext-RELS-TSVM, NPRVFL, efRVFL, RoF-RVFL, en-efRVFL on the ADNI dataset.

	RVFL [110]	IFRVFL	GE-IFWRVFL	CP-FRVFL	ext-LSTSVM	ext-RELS-TSVM	NPRVFL	efRVFL	RoF-RVFL	en-efRVFL
CN_vs_AD	0.8903	0.898	0.8834	0.8843	0.8492	0.8818	0.9117	0.8831	0.8968	0.9127
CN_vs_MCI	0.6295	0.6763	0.7001	0.7188	0.6353	0.6728	0.6883	0.6135	0.6878	0.7407
MCI_vs_AD	0.6621	0.6813	0.6773	0.6705	0.6777	0.6321	0.7033	0.6211	0.6723	0.7006
Average Accuracy	0.7273	0.7519	0.7536	0.7579	0.7207	0.7289	0.7678	0.7059	0.7523	0.7847
Average Rank	7.33	4	5	5	7.33	8.33	2.33	9.33	5	1.33
Overall win tie loss	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 1]	[0, 0, 0]	[1, 0, 0]	[0, 0, 2]	[0, 0, 0]	[2, 0, 0]

5.4 Summary

In this chapter, we developed ensemble variants of the standard RVFL. Ensembles usually improve generalization by leveraging the advantages of many models. They have the ability to reduce overfitting and capture a wide range of patterns in the data. Ensembles tend to be more robust in the face of noisy or uncertain data. They can smooth out individual model errors, leading to more reliable predictions. We proposed extended feature RVFL (efRVFL) and its ensemble, i.e., en-efRVFL, where supervised randomized features (new features) are generated analytically from the original features to get an extended feature space. Three types of features, i.e., original features, supervised randomized features (newly generated features) and (unsupervised) randomized features participate in the proposed efRVFL model for classification. Generally, randomization-based NNs are unstable classifiers, and thus, ensemble learning approach develops stable and more accurate models compared to the single model. Therefore, we proposed an ensemble of extended feature RVFL (en-efRVFL) model. In the proposed en-efRVFL model, each efRVFL model is trained over different extended feature space so that accurate and diverse base models can be generated which leads to better generalization performance. Experiments over 46 UCI datasets demonstrate that the proposed efRVFL and en-efRVFL models have better generalization performance than the standard RVFL and other compared deep models. The proposed efRVFL model performs better than several deep networks, i.e., ResNet, LN, MS, HW, WN, BN, and dRVFL networks. The proposed en-efRVFL has the lowest rank (3.91), among all the compared models over 46 UCI datasets that shows the superiority of the proposed en-efRVFL. Moreover, among the RVFL-based models, the proposed en-efRVFL model has the lowest rank (1.88). To further check the efficiency of the proposed models,

we conducted the experiments over 12 sparse datasets. The proposed en-efRVFL model has the lowest rank (1.58) over sparse datasets. The overall empirical evaluation of the proposed efRVFL and en-efRVFL model shows that the proposed en-efRVFL model has winning performance among the compared deep FNN models.

In another approach, we proposed a novel NN-based ensemble model, i.e., RoF-RVFL. The proposed RoF-RVFL model employs an unsupervised feature extraction technique (PCA) to extract new features from the original datasets. To evaluate the performance of the proposed RoF-RVFL model, extensive experiments have been conducted over 42 benchmark datasets and the experimental results illustrate that RoF-RVFL performs better than other algorithms, i.e., RaF, RoF, and RVFL, in terms of classification accuracy. Results show that feature extraction techniques can boost the performance of the ensemble classifier.

In this chapter, we studied RVFL using different ensemble learning approaches. The proposed ensemble models show better generalization performance than standard RVFL. The proposed efRVFL and en-efRVFL models are suitable for datasets with an insufficient amount of features, as they operate effectively on an extended feature space. One can use the proposed RoF-RVFL with datasets where features are highly correlated (collinear), PCA can decorrelate these features by transforming them into a new set of uncorrelated variables (principal components). This can be beneficial in cases where multicollinearity poses challenges for certain statistical models. However, the proposed models have less feature representation learning as they use shallow RVFL. The standard RVFL is a shallow model that restricts its ability to learn deep feature representations. Deep architectures, with multiple hidden layers, have a higher capacity for learning hierarchical and complex representations from the data. This allows them to capture intricate patterns and relationships, potentially leading to improved performance. The next chapter discusses this issue of RVFL using more advanced theory, i.e., ensemble deep learning.

Chapter 6

Random Vector functional Link Network with Ensemble Deep Learning

Chapter 3 to Chapter 5 developed novel models that use shallow RVFL that have limited capacity to automatically learn hierarchical and abstract features from raw data. The single hidden layer in a shallow architecture struggles to capture complex relationships present in the data. Deep architectures with multiple hidden layers can automatically learn deep feature representations, which may be beyond the capability of a shallow model. Randomized shallow/deep neural networks with closed-form solutions avoid the shortcomings that exist in the BP-based trained NNs. Ensemble deep random vector functional link (edRVFL) network utilizes the strength of two growing fields, i.e., deep learning and ensemble learning. However, the edRVFL model doesn't consider the geometrical relationship of the data while calculating the final output parameters corresponding to each layer considered as base model. In the literature, GE frameworks have been successfully used to describe the geometrical relationship within data. Inspired by the works [73] and [57], in this chapter, we propose extended GE RVFL (EGERVFL) [81] model trained over extended feature space that is generated from original feature space. The proposed EGERVFL model is more capable of capturing the nonlinear hidden relationship within dataset compared to standard RVFL. The proposed EGERVFL model has more information about the data due to additional features. The graph regularization term based on SL criteria is also used in the optimization problem of the proposed EGERVFL model that handles the geometrical properties

of the data. The proposed EGERVFL model is different from standard RVFL as the former model employs graph embedding regularization term in the optimization problem while the later ignores them. The EGERVFL model has only a single hidden layer and hence, has less representation learning. Therefore, we extend the proposed shallow EGERVFL model to an ensemble deep framework known as ensemble deep EGERVFL (edEGERVFL) [81] model. The proposed edEGERVFL model trains EGERVFL in each layer whereas the edRVFL trains standard RVFL. The proposed edEGERVFL model is also different from the dRVFL model as the later one is a deep RVFL network in which all hidden layers outputs are concatenated in a single randomized feature matrix with original features for calculating the final output parameter, whereas the proposed edEGERVFL model follows implicit ensemble learning approach in which each layer is considered as base model and the final output parameters are calculated via averaging/voting method.

The main highlights of this chapter are as follows:

- We proposed a novel extended GE RVFL model that is trained over new extended feature space generated analytically from the original features. The proposed EGERVFL model, unlike standard RVFL, incorporates the SL criteria under the GE framework in its optimization problem to calculate the final output parameters.
- The proposed shallow EGERVFL model has a single hidden layer and hence, has less capability to capture the complex hidden relationships within the dataset. Therefore, we further propose a novel ensemble deep EGERVFL (edEGERVFL) model that follows the deep learning and implicit ensemble learning approaches. The proposed edEGERVFL model has rich representation learning compared to the proposed shallow EGERVFL model.
- We employ the proposed EGERVFL and edEGERVFL models for the diagnosis of AD. The outcomes of the experimental results demonstrate that the proposed models are effective and have better generalization performance compared to the baseline models.
- The proposed EGERVFL and edEGERVFL models have best performance compared to the baseline models for CN vs MCI and CN vs AD cases.

6.1 Proposed models

In this section, we present the shallow extended graph embedded RVFL (EGERVFL) and its extension to an ensemble deep framework, i.e., edEGERVFL model.

6.1.1 Extended graph embedded RVFL (EGERVFL) model

Unlike standard RVFL, the proposed EGERVFL considers the geometrical aspect of the data while training and hence, has good generalization performance. The optimization problem of the EGERVFL model is defined as follows:

$$\min_{\beta} \frac{1}{2} \left\| \hat{H}\beta - Y \right\|^2 + \frac{1}{2}\lambda \mathrm{tr}(\beta^{\mathrm{T}} \mathrm{S} \beta) + \frac{1}{2}\eta \left\| \beta \right\|^2, \tag{6.1}$$

where first term represents the empirical error, second term represents the graph regularization term and the last term represents the regularization term with l_2 norm. β is the output weights matrix. Moreover, tr(.) represents the trace of a matrix and S has the same meaning as defined in Chapter 2 (Section 2.6.3). $\hat{H}$ is the concatenated matrix that contains, original features, newly generated features, and randomized features, as calculated in Chapter 5. Both regularization parameters λ and η handle the impact of these two regularization terms. The problem (6.1) is a convex quadratic problem. Therefore, this problem has a unique global solution. Taking the derivative of (6.1) with respect to β and set it equal to zero, we get,

$$\frac{\partial L}{\partial \beta} = \hat{H}^T(\hat{H}\beta - Y) + \lambda S\beta + \eta\beta = 0, \tag{6.2}$$

after simple calculations, we get the final parameter,

$$\beta = (\hat{H}^T\hat{H} + \lambda S + \eta I)^{-1}\hat{H}^T Y. \tag{6.3}$$

After calculating the output weights matrix β, the output of the proposed GE RVFL model for a new given sample x is calculated similarly to standard RVFL.

There are two choices for choosing matrix S. First, when $S_p = I$, in this case, the matrix

S considers only intrinsic training data relationship. In the proposed EGERVFL model, randomized feature space is used for making the intrinsic graph $G = \{\hat{H}, \Omega\}$. In this case, the matrix S can be defined by $S = \hat{H}^T L \hat{H}$, where $L \in \mathbb{R}^{N \times N}$ as defined in Chapter 2 (Section 2.6.3). The matrix S is calculated in the randomized feature space $(\hat{H})$. Second, the matrix S considers both intrinsic and penalty training data relationships. In this case, the randomized feature space is used to generate both graphs, i.e., $G = \{\hat{H}, \Omega\}$ and $G^p = \{\hat{H}, \Omega^p\}$, Ω and $\Omega^p \in \mathbb{R}^{N \times N}$. The intrinsic and penalty matrices are defined as: $S_i = \hat{H}^T L \hat{H}$ and $S_p = \hat{H}^T L^p \hat{H}$, here, L and L^p are graph Laplacian matrices of G and G^p, respectively. Finally, S is defined as: $S = S_p^{-1} S_i$. When S_p is a singular matrix, we calculate the pseudoinverse of the matrix. To apply the graph embedding approach [74], we use three methods, i.e., linear discriminant analysis (LDA) [347], mixture discriminant analysis (MDA) [74] and local Fisher discriminant analysis (LFDA) [363]. For LDA model, it uses the following graph weights for intrinsic and penalty graphs:

$$\Omega_{ij} = \begin{cases} \frac{1}{N_{c_i}}, & c_j = c_i, \\ 0, & \text{otherwise}, \end{cases} \tag{6.4}$$

$$\Omega^p_{ij} = \begin{cases} \frac{1}{N} - \frac{1}{N_{c_i}}, & c_j = c_i, \\ \frac{1}{N}, & \text{otherwise}, \end{cases} \tag{6.5}$$

respectively. Similarly, the MDA model defines these graph weights as follows:

$$\Omega_{ij} = \begin{cases} 1, & c_i = c_j \text{ and } h_j \in N_i, \\ 1, & c_i = c_j \text{ and } h_i \in N_j, \\ 0, & \text{otherwise}, \end{cases} \tag{6.6}$$

$$\Omega^p_{ij} = \begin{cases} 1, & c_i \neq c_j \text{ and } h_j \in N_i, \\ 1, & c_i \neq c_j \text{ and } h_i \in N_j, \\ 0, & \text{otherwise}. \end{cases} \tag{6.7}$$

Finally, the LFDA model defines these graph weights as:

$$\Omega_{ij} = \begin{cases} \frac{\gamma_{ij}}{N_{c_i}}, & c_j = c_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6.8}$$

$$\Omega^p_{ij} = \begin{cases} \gamma_{ij}\left(\frac{1}{N} - \frac{1}{N_{c_i}}\right), & c_j = c_i, \\ \frac{1}{N}, & \text{otherwise.} \end{cases} \tag{6.9}$$

Here, γ_{ij} measures the similarity between h_i and h_j in the randomized feature space $(\hat{H})$. Also, c_i represents the classes and N_{c_i} represents the number of sample in class (c_i). We employ the heat kernel function for calculating the similarity measure, i.e., $\gamma_{ij} = e^{\frac{-\left\|h_i - h_j\right\|^2}{2\sigma^2}}$, where σ is a scaling parameter. Figure 6.1 shows flowchart of the proposed EGERVFL model.

Algorithm 6.1 Extended Graph embedded RVFL (EGERVFL)

1: Let $X \in \mathbb{R}^{N\times d}$ be the given training data set. $\Phi(.)$ is the activation function. Initialize τ and ω parameters used for generating extended feature space (see Chapter 5). Hidden weights and biases (b) are generated randomly.
2: Calculate the extended feature space (E) (given in Chapter 5) and randomized feature space $(\hat{H})$.
3: Calculate intrinsic and penalty graph weights Ω and Ω^p from (6.4) to (6.8).
4: Calculate the graph Laplacian matrices $L = D - \Omega$ and $L^p = D^p - \Omega^p$.
5: Calculate: $S_i = \hat{H}^T L \hat{H}$, $S_p = \hat{H}^T L^p \hat{H}$ and $S = S_p^{-1} S_i$.
6: Calculate the final output weights of the proposed model by (6.3).

6.1.2 Ensemble deep EGERVFL (edEGERVFL) model

The proposed EGERVFL model has only one hidden layer and hence, it has less representation learning. Deep models are successful because of their rich representation learning and have better generalization performance compared to shallow models. Therefore, we propose ensemble deep variant of EGERVFL model, i.e., ensemble deep extended GE RVFL (edEGERVFL). Similar to standard edRVFL, the proposed edEGERVFL model has several hidden layers and each layer constitutes a base model in the ensemble learning framework.

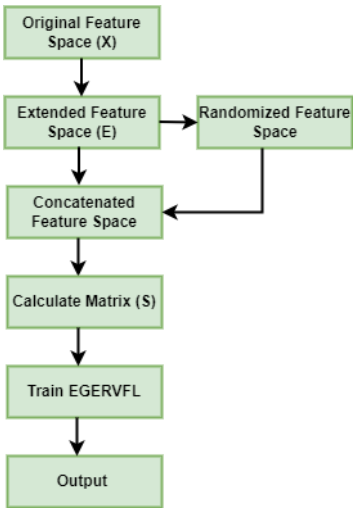


Figure 6.1: The flowchart of the proposed EGERVFL model.

The first hidden layer's output in the proposed edEGERVFL model is defined as:

$$\hat{L}^{(1)} = \Phi([X\ \bar{X}]W^{(1)}), \tag{6.10}$$

and for higher hidden layers ($i > 1$) is defined as:

$$\hat{L}^{(i)} = \Phi([\hat{L}^{(i-1)}X\ \bar{X}]W^{(i)}). \tag{6.11}$$

Where X and $\bar{X}$ have the same meaning as given in Chapter 5. In the proposed edEGERVFL model, each base model considers the geometrical relationship of the data while training. Unlike edRVFL, that trains RVFL at each layer, the proposed edEGERVFL model trains EGERVFL at each layer.

The output weights corresponding to each layer are calculated by Eq. (6.3). The final decision of the proposed edEGERVFL model is taken by the averaging method (similar to [57]).

6.2 Computational complexity

Let (X, Y) be the training set with $X \in \mathbb{R}^{N\times m}$ and $Y \in \mathbb{R}^{N\times c}$. Here, N and m are the number of samples and number of features, respectively and c is the number of classes.

Suppose h_l is the number of hidden nodes in each layer and k represents the number of layers considered in deep RVFL models. In all the RVFL-based models, the optimization of output layer weights involves the computation of matrix inverses. Hence, the complexity of the models is determined by the size of the matrices to be inverted. By the standard procedure, the matrix inversion of size $N \times N$ require $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ time [337]. The standard RVFL model solves the output layer weights either in the primal or dual space, which results in reducing the complexity of the model by choosing among $min(N, m+h_l)$. Thus, the time complexity of RVFL model is either $\mathcal{O}(N^3)$ or $\mathcal{O}((m+h_l)^3)$. The complexity of dRVFL is either $\mathcal{O}(N^3)$ or $\mathcal{O}((m+kh_l)^3)$ whichever results in lower computation. In the edRVFL model, the matrix inversions are required which are of the size $N \times N$ or $((h_l+m) \times (h_l+m))$ for the first layer and $((2h_l+m) \times (2h_l+m))$ for the higher layers. In the proposed models, we need to calculate the extended feature space wherein atmost d new features can be generated. For each new feature, we need to take the inverse of a square matrix of order d. Therefore, the computational complexity of calculating m features is equal to $\mathcal{O}(md^3)$. Then, we need to calculate the matrix (S) which involves the inverse of a square matrix of order $(2m+h_l)$ so the corresponding complexity is $\mathcal{O}((2m+h_l)^3)$. The computational complexity of the proposed EGERVFL model is $\mathcal{O}(md^3)+2\mathcal{O}((2m+h_l)^3)$ as the EGERVFL model works in feature space. In the proposed edEGERVFL model, we have k number of hidden layers. Therefore, it has $\mathcal{O}(md^3)+\mathcal{O}((2m+h_l)^3)+\mathcal{O}(k(2m+2h_l)^3)$ complexity.

6.3 Experiments

In this section, we illustrate the experimental results of the proposed EGERVFL, edEGERVFL and compared models.

6.3.1 Experimental setup

All the experiments are conducted over MATLAB R2017b and the workstation with Intel Xenon(R) CPU E5-2697 v4 2.30 GHZ and 128 GB RAM. The datasets are randomly partitioned into $70:30$ ratio. Here, 70% of the samples are used for training whereas 30%

samples are reserved for testing the models. We use 5 fold cross validation on the training set for optimizing the hyperparameters corresponding to different classification models.

The hyperparameters corresponding to different classification models are obtained via grid search approach. The regularization parameters ($\gamma = \eta$) for the proposed models are set to be $\frac{1}{C}$ where C is chosen from the following range $\{2^{-6}, 2^{-4}, \cdots, 2^{12}\}$ and the number of hidden neurons are taken from $\{1028, 2048, 4096\}$. We used three activation functions, i.e., sigmoid, relu and selu in our experiments. We followed 2-stage tuning method [57] for obtaining the optimal hyperparameters. In the first stage, we optimize the parameters and get optimal hidden nodes h_l^* and regularization parameter η^* wherein we fixed the hidden layers to two. In the second stage, number of layers and other parameters are tuned in the neighbourhood of h_l^* and η^*.

6.3.2 Evaluation on ADNI dataset

The scans from the ADNI repository (adni.loni.usc.edu) are used in this study. In 2003, the Principal Investigator of ADNI project, Michael W. Weiner, launched the project. The aim of this project is to analyze the neuroimages like PET (positron emission tomography), MRI (magnetic resonance imaging) and other tests for the early diagnosis of the AD from the MCI stage. For detailed information, we refer the interested readers to www.adni-info.org. In this study, Volume-based Morphometry (VolBM) based features are used. The feature extraction pipeline followed is the same as in [338]. The classification models are analyzed via generalization performance over three cases, i.e., CN versus MCI, CN versus AD, and MCI versus AD cases.

For VolBM analysis, Freesurfer's recon-all pipeline (version 6.0.1) [364, 365] is used on structural MRI images. Out of 150 MRI images, 1 MCI image failed to process in Freesurfer. So, feature selection was performed on 149 images. We extracted 23 subcortical tissue volumes (SCV), 34 WM tissue volumes (WMV), and 34 cortical thickness (CT) measures of every subject. To check the performance of our model on an independent dataset, we downloaded 817 sMRI images from ADNI baseline dataset [366, 367], out of which 4 images failed to process through the Freesurfer pipeline. Thus, our baseline dataset

includes 228 CN, 398 MCI, and 187 AD images. For more information one can visit [328].

6.3.3 Experimental results and discussion

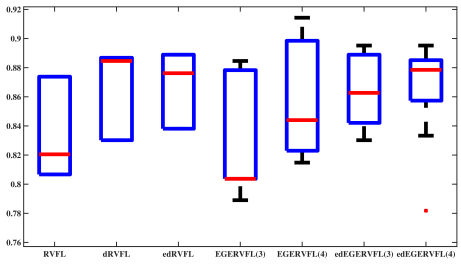
In this section, we discuss the experimental results based on the proposed EGERVFL and edEGERVFL models and the compared models, i.e., standard RVFL, deep RVFL (dRVFL) and ensemble deep RVFL (edRVFL) models. To evaluate the strength of randomized feature space and its effects on the models, we used three different activation functions, i.e., selu, relu and sigmoid functions. We also used some metrics such as Accuracy (AUC), Sensitivity (Sens.), Specificity (Spec.), F-measure, and Precision (Prec.) to evaluate the classification performances of the models. In the feature generation method, τ is the randomly selected feature parameter. These additional features give the strength to the proposed models to capture the nonlinear hidden relationship within dataset. So we conducted the experiments with $\tau = 3, 4$. The EGERVFL(3) or edEGERVFL(3) with MDA represent the results of the proposed EGERVFL or edEGERVFL model with $\tau = 3$ and MDA-based GE approach. A similar meaning is followed for LDA and LFDA also. The performance of the models to classify the different stages of AD is analyzed in this study.

Table 6.1: Experimental results of the algorithms on CN vs AD case.

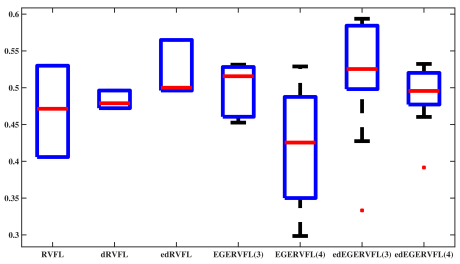
Act.	RVFL	dRVFL	edRVFL	EGERVFL(3)	EGERVFL(4)	edEGERVFL(3)	edEGERVFL(4)
				MDA			
	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)
Selu	(0.8432, 0.9057) (0.7808, 0.75)	(**0.8997**, 0.8679) (0.9315, 0.902)	(0.8929, 0.8679) (0.9178, 0.8846)	(0.8303, 0.8113) (0.8493, 0.7963)	(0.8398, 0.8302) (0.8493, 0.8)	(**0.9023**, 0.8868) (0.9178, 0.8868)	(**0.8997**, 0.8679) (0.9315, 0.902)
Relu	(0.8295, 0.9057) (0.7534, 0.7273)	(0.8535, 0.8302) (0.8767, 0.8302)	(**0.9049**, 0.9057) (0.9041, 0.8727)	(0.8303, 0.8113) (0.8493, 0.7963)	(0.8655, 0.8679) (0.863, 0.8214)	(**0.9091**, 0.8868) (0.9315, 0.9038)	(0.8955, 0.8868) (0.9041, 0.8704)
Sigmoid	(0.8903, 0.8491) (0.9315, 0.9)	(**0.9023**, 0.8868) (0.9178, 0.8868)	(0.8603, 0.8302) (0.8904, 0.8462)	(0.8997, 0.8679) (0.9315, 0.902)	(**0.9254**, 0.9057) (0.9452, 0.9231)	(0.8834, 0.8491) (0.9178, 0.8824)	(0.8877, 0.8302) (0.9452, 0.9167)
				LFDA			
Selu	(0.8432, 0.9057) (0.7808, 0.75)	(**0.8997**, 0.8679) (0.9315, 0.902)	(0.8929, 0.8679) (0.9178, 0.8846)	(0.8303, 0.8113) (0.8493, 0.7963)	(0.8398, 0.8302) (0.8493, 0.8)	(0.8586, 0.8679) (0.8493, 0.807)	(**0.9091**, 0.8868) (0.9315, 0.9038)
Relu	(0.8295, 0.9057) (0.7534, 0.7273)	(0.8535, 0.8302) (0.8767, 0.8302)	(**0.9049**, 0.9057) (0.9041, 0.8727)	(0.8303, 0.8113) (0.8493, 0.7963)	(0.8655, 0.8679) (0.863, 0.8214)	(**0.9091**, 0.8868) (0.9315, 0.9038)	(0.8955, 0.8868) (0.9041, 0.8704)
Sigmoid	(0.8903, 0.8491) (0.9315, 0.9)	(**0.9023**, 0.8868) (0.9178, 0.8868)	(0.8603, 0.8302) (0.8904, 0.8462)	(0.8997, 0.8679) (0.9315, 0.902)	(**0.9254**, 0.9057) (0.9452, 0.9231)	(0.8808, 0.8302) (0.9315, 0.898)	(0.8834, 0.8491) (0.9178, 0.8824)
				LDA			
Selu	(0.8432, 0.9057) (0.7808, 0.75)	(**0.8997**, 0.8679) (0.9315, 0.902)	(**0.8929**, 0.8679) (0.9178, 0.8846)	(0.8166, 0.8113) (0.8219, 0.7679)	(0.8492, 0.8491) (0.8493, 0.8036)	(0.8655, 0.8679) (0.863, 0.8214)	(0.856, 0.8491) (0.863, 0.8182)
Relu	(0.8295, 0.9057) (0.7534, 0.7273)	(0.8535, 0.8302) (0.8767, 0.8302)	(**0.9049**, 0.9057) (0.9041, 0.8727)	(0.8303, 0.8113) (0.8493, 0.7963)	(0.8586, 0.8679) (0.8493, 0.807)	(0.8655, 0.8679) (0.863, 0.8214)	(**0.9023**, 0.8868) (0.9178, 0.8868)
Sigmoid	(0.8903, 0.8491) (0.9315, 0.9)	(**0.9023**, 0.8868) (0.9178, 0.8868)	(0.8603, 0.8302) (0.8904, 0.8462)	(0.8929, 0.8679) (0.9178, 0.8846)	(**0.9066**, 0.8679) (0.9452, 0.92)	(0.8535, 0.8302) (0.8767, 0.8302)	(0.8098, 0.8113) (0.8082, 0.7544)

The performances of the top two models are shown in bold face.

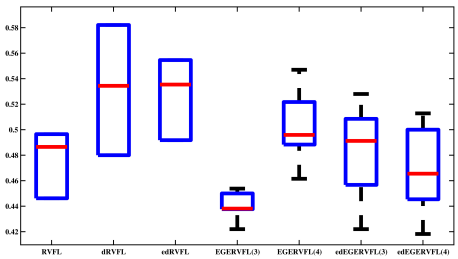
[1] **CN vs AD case:** The classification performances of the models are shown in Table 6.1. **For MDA case**, one can see that the proposed edEGERVFL(3) has highest performance with AUC equal to 90.23%, the proposed edEGERVFL(4) and dRVFL got second position with AUC equal to 89.97% for selu activation function. For other activation functions, i.e., relu and sigmoid, the proposed edEGERVFL(3) and EGERVFL(4) are the best performing models with AUC equal to 90.91% and 92.54%, respectively. It shows that edEGERVFL(3) performs better with selu and relu function as compared to sigmoid function. One can observe that the features space generated by $\tau = 3$ is more informative than the space generated by $\tau = 4$. The proposed EGERVFL(4) model with sigmoid function has almost 6% and 2.5% better accuracy than edRVFL and dRVFL model, respectively. These increments in accuracy show the efficiency of the proposed models. **For LFDA case**, both proposed edEGERVFL(3) and edEGERVFL(4) have better performance with AUC equal to 90.91% and second highest models are dRVFL and edRVFL with AUC equal to 89.97% and 90.49% with selu and relu activation functions, respectively. Moreover, proposed EGERVFL(4) has highest AUC equal to 92.54% with sigmoid function. **For LDA case**, dRVFL model is the winner with AUC equal to 89.97% with selu function and the proposed edEGERVFL(4) has the second position with AUC equal to 90.23% with relu function and finally, the proposed EGERVFL(4) has highest AUC equal to 90.66% among the compared models with sigmoid function. Over all conclusion is that the proposed edEGERVFL(3) or (4) has the best performance with selu and and relu activation functions and with MDA and LFDA cases. For the LDA case, dRVFL and edRVFL models are performing better than the proposed models. One can observe from Figure 6.2a that the higher F-measure values are achieved by the proposed EGERVFL model. The proposed edEGERVFL model has competitive performance with the existing dRVFL and edRVFL models. A higher F-measure indicates high precision and recall which means that our proposed model has less false positive and false negative, that is very important for a classifier. Figure 6.3a shows the model's performances with different activation functions and LDA, LFDA and MDA cases as well. Here, F-measures corresponding to EGERVFL(3) or (4) are higher compared to other models.



(a) CN vs AD



(b) CN vs MCI

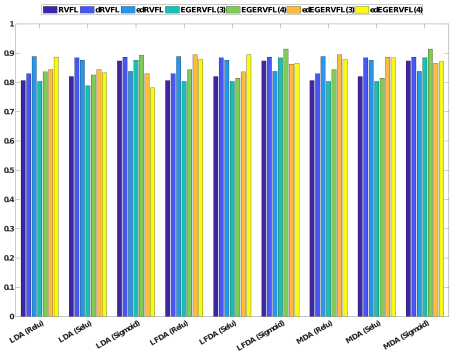


(c) MCI vs AD

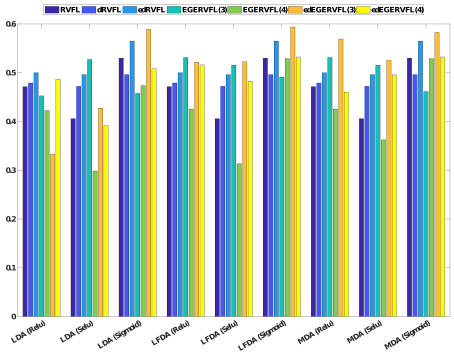
Figure 6.2: F-measure box plot analysis of the models for AD.

[2] **CN vs MCI case:** Table 6.2 shows the classification performances of the models for CN vs MCI case. **For MDA case**, among all compared models, the proposed edEGERVFL(3) has the highest AUC equal to 65.25%, 68.14% and 69% with selu, relu and sigmoid activation functions, respectively. Also, one can observe that the features space generated by $\tau = 3$ is more informative than the space generated by $\tau = 4$. The proposed edEGERVFL(3) with sigmoid function has almost 2% better accuracy than edRVFL model. **For the LFDA case**, the proposed edEGERVFL(3) with AUC equal to 65.57% and 65.22% corresponding to selu and relu functions has better performance whereas, it has highest performance with AUC equal to 69.82% with sigmoid function.

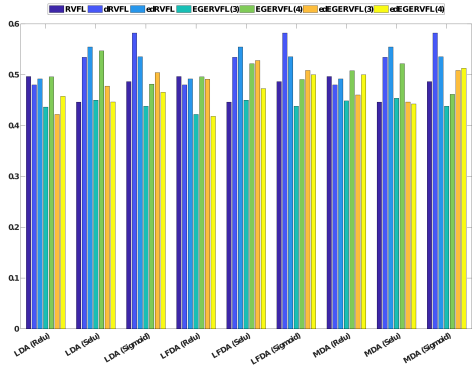
The proposed EGERVFL(3) with AUC equal to 63.77% and 64.98% has second position with selu and relu functions, respectively. **For the LDA case**, the EGERVFL(3) model has winning performance that has AUC equal to 64.59% with selu function. One can see that edRVFL is performing well compared to the proposed models with



(a) CN vs AD



(b) CN vs MCI



(c) MCI vs AD

Figure 6.3: F-measure analysis of the models for AD.

Table 6.2: Experimental results of the algorithms on CN vs MCI case.

Act.	RVFL	dRVFL	edRVFL	EGERVFL(3)	EGERVFL(4)	edEGERVFL(3)	edEGERVFL(4)
				MDA			
	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)
Selu	(0.5381, 0.459) (0.6172, 0.3636)	(0.5873, 0.5574) (0.6172, 0.4096)	(0.6181, 0.541) (0.6953, 0.4583)	(**0.6377**, 0.541) (0.7344, 0.4925)	(0.5018, 0.4098) (0.5938, 0.3247)	(**0.6525**, 0.5082) (0.7969, 0.5439)	(0.6358, 0.459) (0.8125, 0.5385)
Relu	(0.5908, 0.541) (0.6406, 0.4177)	(0.5951, 0.5574) (0.6328, 0.4198)	(0.615, 0.5738) (0.6563, 0.443)	(**0.6498**, 0.5574) (0.7422, 0.5075)	(0.5506, 0.4918) (0.6094, 0.375)	(**0.6814**, 0.5738) (0.7891, 0.5645)	(0.5971, 0.4754) (0.7188, 0.4462)
Sigmoid	(0.6564, 0.5082) (0.8047, 0.5536)	(0.6181, 0.541) (0.6953, 0.4583)	(**0.6744**, 0.6066) (0.7422, 0.5286)	(0.608, 0.4426) (0.7734, 0.4821)	(0.6529, 0.5246) (0.7813, 0.5333)	(**0.69**, 0.6066) (0.7734, 0.5606)	(0.6533, 0.541) (0.7656, 0.5238)
				LFDA			
Selu	(0.5381, 0.459) (0.6172, 0.3636)	(0.5873, 0.5574) (0.6172, 0.4096)	(0.6181, 0.541) (0.6953, 0.4583)	(**0.6377**, 0.541) (0.7344, 0.4925)	(0.469, 0.3443) (0.5938, 0.2877)	(**0.6557**, 0.4754) (0.8359, 0.58)	(0.624, 0.459) (0.7891, 0.5091)
Relu	(0.5908, 0.541) (0.6406, 0.4177)	(0.5951, 0.5574) (0.6328, 0.4198)	(0.615, 0.5738) (0.6563, 0.443)	(**0.6498**, 0.5574) (0.7422, 0.5075)	(0.5506, 0.4918) (0.6094, 0.375)	(**0.6522**, 0.4918) (0.8125, 0.5556)	(0.6412, 0.5246) (0.7578, 0.5079)
Sigmoid	(0.6564, 0.5082) (0.8047, 0.5536)	(0.6181, 0.541) (0.6953, 0.4583)	(**0.6744**, 0.6066) (0.7422, 0.5286)	(0.6319, 0.459) (0.8047, 0.5283)	(0.6529, 0.5246) (0.7813, 0.5333)	(**0.6982**, 0.623) (0.7734, 0.5672)	(0.6533, 0.541) (0.7656, 0.5238)
				LDA			
Selu	(0.5381, 0.459) (0.6172, 0.3636)	(0.5873, 0.5574) (0.6172, 0.4096)	(**0.6181**, 0.541) (0.6953, 0.4583)	(**0.6459**, 0.5574) (0.7344, 0.5)	(0.4569, 0.3279) (0.5859, 0.274)	(0.5838, 0.4098) (0.7578, 0.4464)	(0.5186, 0.459) (0.5781, 0.3415)
Relu	(0.5908, 0.541) (0.6406, 0.4177)	(**0.5951**, 0.5574) (0.6328, 0.4198)	(**0.615**, 0.5738) (0.6563, 0.443)	(0.5783, 0.5082) (0.6484, 0.4079)	(0.5467, 0.4918) (0.6016, 0.3704)	(0.4663, 0.3934) (0.5391, 0.2892)	(0.5923, 0.6066) (0.5781, 0.4066)
Sigmoid	(0.6564, 0.5082) (0.8047, 0.5536)	(0.6181, 0.541) (0.6953, 0.4583)	(**0.6744**, 0.6066) (0.7422, 0.5286)	(0.6041, 0.4426) (0.7656, 0.4737)	(0.6197, 0.4426) (0.7969, 0.5094)	(**0.6943**, 0.623) (0.7656, 0.5588)	(0.6369, 0.5082) (0.7656, 0.5082)

The performances of the top two models are shown in bold face.

selu and relu activation functions. However, the proposed edEGERVFL(3) has overall best accuracy (for LDA case) 69.43% with sigmoid function whereas RVFL, dRVFL and edRVFL has accuracy 65.64%, 61.81% and 67.44%, respectively. Here, the proposed edEGERVFL(3) model with MDA and LFDA cases is the best-performing model. Figure 6.2b shows the box plot diagram among the compared models. It can be observed that the proposed edEGERVFL model has higher F-measure compared to other models. F-measure is a harmonic mean of precision and recall. Therefore, higher F-measure indicates high both precision and recall. As can be observed from Figure 6.3b that edEGERVFL with LDA (sigmoid), LFDA (sigmoid), MDA (relu) and MDA (sigmoid) shows winning performance with higher F-measure. The proposed edEGERVFL(3) achieved 69.82% accuracy (with sigmoid) which is better compared to the baseline models. Here, edRVFL achieved 67.44% which is second highest. Hence, the proposed model is at least 2% more accurate compared to the baseline models.

[3] **MCI vs AD case:** The classification performances of the models for MCI vs AD case are given in Table 6.3. For the MDA case, edRVFL and the proposed EGERVFL(4) have a first and second positions with AUC equal to 66.01% and 64.15% with selu function, respectively. The edEGERVFL(4) with AUC equal to 63.06% is performing

better than edRVFL with AUC equal to 61.02%. As one can see that our proposed model is at least 2% more accurate than the edRVFL model with relu function. The dRVFL model with AUC equal to 66.61% has the winning performance with sigmoid function. For LFDA and LDA case, both dRVFL and edRVFL models are performing better than the proposed models. Finally, one can observe that the proposed EGERVFL(3) or (4) is performing better than the standard RVFL model with all three activation functions and with MDA, LFDA and LDA cases. It can be observed from Figure 6.2c that the proposed models don't perform well compared to the existing models except RVFL. Of course, in this case, our proposed models are not performing well as can be seen in Figure 6.3c. Here, overall the dRVFL model is performing very well.

Table 6.3: Experimental results of the algorithms on MCI vs AD case.

Act.	RVFL	dRVFL	edRVFL	EGERVFL(3)	EGERVFL(4)	edEGERVFL(3)	edEGERVFL(4)
				MDA			
	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)	(AUC, Sens.) (Spec., Prec.)
Selu	(0.5624, 0.4462) (0.6786, 0.4462)	(0.6308, 0.5385) (0.7232, 0.5303)	(**0.6601**, 0.5077) (0.8125, 0.6111)	(0.5872, 0.4154) (0.7589, 0.5)	(**0.6415**, 0.4615) (0.8214, 0.6)	(0.5782, 0.4154) (0.7411, 0.4821)	(0.5896, 0.3846) (0.7946, 0.5208)
Relu	(0.5862, 0.5385) (0.6339, 0.4605)	(0.5968, 0.4615) (0.7321, 0.5)	(0.6102, 0.4615) (0.7589, 0.5263)	(0.6043, 0.3692) (0.8393, 0.5714)	(**0.6167**, 0.4923) (0.7411, 0.5246)	(0.6018, 0.4) (0.8036, 0.5417)	(**0.6306**, 0.4308) (0.8304, 0.5957)
Sigmoid	(0.6229, 0.4154) (0.8304, 0.587)	(**0.6661**, 0.6) (0.7321, 0.5652)	(**0.6365**, 0.5231) (0.75, 0.5484)	(0.601, 0.3538) (0.8482, 0.575)	(0.6177, 0.3692) (0.8661, 0.6154)	(0.6281, 0.4615) (0.7946, 0.566)	(0.6326, 0.4615) (0.8036, 0.5769)
				LFDA			
Selu	(0.5624, 0.4462) (0.6786, 0.4462)	(0.6308, 0.5385) (0.7232, 0.5303)	(**0.6601**, 0.5077) (0.8125, 0.6111)	(0.5827, 0.4154) (0.75, 0.4909)	(**0.6415**, 0.4615) (0.8214, 0.6)	(0.6333, 0.5077) (0.7589, 0.55)	(0.6152, 0.4) (0.8304, 0.5778)
Relu	(0.5862, 0.5385) (0.6339, 0.4605)	(0.5968, 0.4615) (0.7321, 0.5)	(0.6102, 0.4615) (0.7589, 0.5263)	(0.5832, 0.3538) (0.8125, 0.5227)	(**0.6147**, 0.4615) (0.7679, 0.5357)	(**0.6216**, 0.4308) (0.8125, 0.5714)	(0.5787, 0.3538) (0.8036, 0.5111)
Sigmoid	(0.6229, 0.4154) (0.8304, 0.587)	(**0.6661**, 0.6) (0.7321, 0.5652)	(**0.6365**, 0.5231) (0.75, 0.5484)	(0.601, 0.3538) (0.8482, 0.575)	(0.633, 0.4) (0.8661, 0.6341)	(0.6281, 0.4615) (0.7946, 0.566)	(0.6249, 0.4462) (0.8036, 0.5686)
				LDA			
Selu	(0.5624, 0.4462) (0.6786, 0.4462)	(0.6308, 0.5385) (0.7232, 0.5303)	(**0.6601**, 0.5077) (0.8125, 0.6111)	(0.5827, 0.4154) (0.75, 0.4909)	(**0.6569**, 0.4923) (0.8214, 0.6154)	(0.581, 0.4923) (0.6696, 0.4638)	(0.5941, 0.3846) (0.8036, 0.5319)
Relu	(0.5862, 0.5385) (0.6339, 0.4605)	(0.5968, 0.4615) (0.7321, 0.5)	(**0.6102**, 0.4615) (0.7589, 0.5263)	(0.5909, 0.3692) (0.8125, 0.5333)	(**0.6147**, 0.4615) (0.7679, 0.5357)	(0.5832, 0.3538) (0.8125, 0.5227)	(0.5916, 0.4154) (0.7679, 0.5094)
Sigmoid	(0.6229, 0.4154) (0.8304, 0.587)	(**0.6661**, 0.6) (0.7321, 0.5652)	(**0.6365**, 0.5231) (0.75, 0.5484)	(0.601, 0.3538) (0.8482, 0.575)	(0.6241, 0.4) (0.8482, 0.6047)	(0.6236, 0.4615) (0.7857, 0.5556)	(0.6005, 0.4154) (0.7857, 0.5294)

The performances of the top two models are shown in bold face.

6.3.4 Results analysis on UCI datasets

Apart from AD datasets, we conducted the experiments on 14 publicly available UCI datasets [267] to validate the performance of the proposed models. Table 6.4 shows the corresponding results. Here, the results of RVFL, dRVFL and edRVFL models are taken from paper [57]. The results show that the proposed edEGERVFL (LFDA) has the highest

Table 6.4: Experimental results of the models on the UCI datasets.

Datasets	RVFL	dRVFL†	edRVFL†	EGERVFL(MDA)	EGERVFL(LFDA)	EGERVFL(LDA)	edEGERVFL(MDA)	edEGERVFL(LFDA)	edEGERVFL(LDA)
car	97.74	97.97	98.04	98.32	98.26	98.03	97.04	97.41	95.87
cardiotocography_10clases	82.82	83.62	83.24	83.33	83.33	83.29	85.59	85.57	86.3
cardiotocography_3clases	92.37	92.84	92.66	93.08	92.84	92.56	85.97	84.44	85.13
image_segmentation	88.33	89.1	88.52	88.71	89.29	89.24	95.5	95.28	94.19
led_display	74.7	73.9	74.4	74.6	74.5	74.8	73.41	76.99	74.07
oocytes_merluccius_nucleus_4d	85.1	82.75	84.41	85.88	85.39	84.8	81.47	82.07	78.7
oocytes_merluccius_states_2f	91.96	91.86	93.63	92.94	93.73	93.33	94.17	94.11	90.47
plant_margin	81.81	81.88	81.88	83.44	83.88	83.69	89.8	87.91	85.57
plant_shape	67.63	71.75	72.31	70.5	70.38	70.31	84.94	81.47	74.4
plant_texture	83.88	84.06	85.25	85.06	84.94	84.56	91.79	91.67	91.38
semeion	92.9	93.47	92.96	90.33	90.33	90.58	94.48	93.43	90.23
statlog_image	96.71	96.79	96.84	96.19	96.19	96.53	98.13	98.08	97.83
steel_plates	75.52	58.66	76.44	76.24	75.98	75.67	63.11	82.51	81.58
yeast	61.73	60.24	61.66	60.71	60.92	60.78	61.56	61.03	62.11
Average Accuracy	83.8	82.78	84.45	84.24	84.28	84.16	85.5	86.57	84.85
Average Rank	6.43	6.07	4.82	5.04	4.86	5.57	3.79	3.43	5

86.57% accuracy. The other two proposed edEGERVFL (MDA) and edEGERVFL (LDA) models have 85.5%, and 84.85% accuracy, respectively. The existing edRVFL model has 84.45% accuracy that is almost 2% less accurate than the proposed edEGERVFL model. One can observe that the proposed shallow EGERVFL with MDA, LFDA, LDA has accuracy, 84.24%, 84.28% and 84.16%, respectively. All these three variants (of RVFL) are performing better than standard RVFL that has 83.8% accuracy. We have also conducted the Friedman rank test [362] that gives a lower rank to the best-performing model and the highest rank is given to the least-performing model. Here, the proposed edEGERVFL with LFDA and MDA has lowest (3.43) and second lowest (3.79) ranks, respectively and hence, the proposed ensemble deep models have better generalization performance than the existing baseline models. The reason behind that why edEGERVFL (LFDA) is the best-performing model because of its weighting mechanism that uses kernel function for calculating the similarity measure while the others, LDA and MDA don't use the same. The similar observation can be seen from shallow EGERVFL (LFDA). It has better generalization performance than the proposed EGERVFL with LDA and MDA. The proposed edEGERVFL (MDA) is performing better than the proposed edEGERVFL (LDA) model. Similarly, one can observe that EGERVFL (MDA) has better generalization performance than EGERVFL (LDA).

6.4 Experimental results of the proposed models on ADNI dataset

To check the combined performance of the proposed models, we conducted the experiments with the shallow and deep RVFL models on the ADNI dataset, and the corresponding results are shown in Table 6.5. It shows the experimental results of a shallow extended graph embedded RVFL (EGERVFL) and its deep variant (edEGERVFL) with LDA, LFDA, and MDA techniques. All deep proposed models, i.e. edEGERVFL (LFDA), edEGERVFL (MDA), and edEGERVFL (LDA) have superior performance with average accuracy at 77.64%, 77.46%, and 76.84%, respectively. One can observe that all proposed shallow models (EGERVFL (LDA), EGERVFL (LFDA), EGERVFL (MDA)) have better generalization performance than the standard RVFL model. Among deep models, edEGERVFL (MDA) has the lowest average rank at 2.33, followed by edEGERVFL (LDA) at 3. The proposed deep models have almost 4% more average accuracy than standard RVFL. Among shallow models, the proposed EGERVFL (MDA) has the highest average accuracy (74.93%) and a small average rank (3.33). The last row in Table 6.5 shows overall win-tie-loss which demonstrates that the edEGERVFL with LFDA and MDA has 1-1 winning with no loss. Moreover, Table 6.6 shows the performance of the models proposed in Chapter 3 to Chapter 6. The results show that the proposed en-efRVFL has a first position at 78.24%, followed by edEGERVFL at 77.64%, NPRVFL at 76.78%.

Table 6.5: Experimental results of the RVFL and the proposed shallow EGERVFL and deep EGERVFL models on the ADNI dataset.

	RVFL [110]	EGERVFL (LDA)	EGERVFL (LFDA)	EGERVFL (MDA)	edEGERVFL (LDA)	edEGERVFL (LFDA)	edEGERVFL (MDA)
CN_vs_AD	0.8903	0.8997	0.9091	0.9091	0.912698	0.904762	0.920635
CN_vs_MCI	0.6295	0.6974	0.6455	0.6455	0.708995	0.740741	0.719577
MCI_vs_AD	0.6621	0.6298	0.6857	0.6934	0.683616	0.683616	0.683616
Avearge AUC	0.7273	0.7423	0.7468	0.7493	0.7684	0.7764	0.7746
Average Rank	6.67	5.67	3.67	3.33	3	3.33	2.33
Overall win-tie-loss	$[0, 0, 2]$	$[0, 0, 1]$	$[0, 1, 0]$	$[1, 2, 0]$	$[0, 0, 0]$	$[1, 0, 0]$	$[1, 0, 0]$

6.5 Summary

In this chapter, we developed shallow and ensemble deep variants of the RVFL. Ensemble Deep models can automatically learn hierarchical and abstract features from raw data.

Table 6.6: Experimental results of the RVFL and the proposed models, i.e. IFRVFL, GE-IFWRVFL, CP-FRVFL, ext-LSTSVM, ext-RELS-TSVM, NPRVFL, efRVFL, RoF-RVFL, en-efRVFL, EGERVFL, and edEGERVFL on the ADNI dataset.

	RVFL [110]	IFRVFL	GE-IFWRVFL	CP-FRVFL	ext-LSTSVM	ext-RELSTSVM	NPRVFL	efRVFL	RoF-RVFL	en-efRVFL	EGERVFL (MDA)	edEGERVFL (LFDA)
CN_vs_AD	0.8903	0.898	0.8834	0.8843	0.8492	0.8818	0.9117	0.8831	0.8968	0.9127	0.9091	0.9048
CN_vs_MCI	0.6295	0.6763	0.7001	0.7188	0.6353	0.6728	0.6883	0.6135	0.6878	0.7407	0.6455	0.7407
MCI_vs_AD	0.6621	0.6813	0.6773	0.6705	0.6777	0.6321	0.7033	0.6211	0.6723	0.7006	0.6934	0.6836
Average Accuracy	0.7273	0.7519	0.7536	0.7579	0.7207	0.7289	0.7678	0.7059	0.7523	0.7847	0.7493	.7764
Average Rank	9.33	5.67	6.66	6.69	9.33	10	2.67	11.33	6.67	1.5	5	3.17
Overall win-tie- loss	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 1]	[0, 0, 0]	[1, 0, 0]	[0, 0, 2]	[0, 0, 0]	[1, 1, 0]	[0, 0, 0]	[0, 1, 0]

The depth of the ensemble deep architecture allows for the extraction of features at multiple levels, capturing complex patterns and relationships that a shallow model may struggle to represent. We proposed extended graph embedded RVFL (EGERVFL) model that can be considered a variant of standard RVFL. Unlike standard RVFL, the proposed EGERVFL model is trained over extended feature space and hence, possesses more capability to capture the nonlinear hidden relationships within the dataset. Moreover, we used the graph regularization term in the optimization problem of the proposed EGERVFL model under the GE framework that incorporates the geometrical relationship within data. Shallow networks are not good for representation learning. Therefore, we extended the proposed shallow EGERVFL model to an ensemble deep framework namely, the ensemble deep EGERVFL (edEGERVFL) model. The proposed edEGERVFL model solves GE-based optimization problem in each layer and hence has better generalization performance. To evaluate the proposed models, we employed them for the diagnosis process of AD. The experimental results demonstrate the strength and efficiency of the proposed models. The proposed EGERVFL model (with MDA, LFDA, and LDA) has better classification performance compared to the standard RVFL model for all three cases. The proposed edEGERVFL model (with MDA and LFDA) has the overall best performance among the compared models on CN vs AD and CN vs MCI. Moreover, the proposed models are implemented over UCI datasets and the results demonstrate that the proposed EGERVFL and edEGERVFL with LFDA have better generalization performance compared to the baseline models. The proposed deep model, i.e. edEGERVFL is suitable for the problems where one needs deep feature representation learning, fast training speed, and better generalization performance.

Chapter 7

Conclusions and Future Works

Randomized neural networks have shown their strength among machine learning models. In this thesis, we focused on a special kind of randomized network, i.e. RVFL, that has emerged as a very successful randomized network. We developed novel variants of RVFL for classification problems and their application in the diagnosis of Alzheimer's disease. The proposed works use a wide range of approaches including feature engineering, fuzzy and intuitionistic fuzzy theory, graph embedding theory, hyperplane-based learning, deep representations, and ensemble learning to develop novel RVFL variants. The brief summary of this thesis is as follows:

7.1 Conclusions

First, we presented a thorough survey on the developments of the RVFL model in different aspects, i.e., shallow RVFL, ensemble algorithms based on the RVFL, deep RVFL variants, etc. Also, we presented applications of the RVFL models that show their applicability in the real world. The presented comprehensive survey of RVFL can serve as an extensive summary for beginners as well as practitioners.

Second, we proposed three variants of RVFL, i.e., intuitionistic fuzzy RVFL (IFRVFL), class probability-based fuzzy RVFL (CP-FRVFL) and graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) network. Unlike standard RVFL, which uses a uniform weighting approach, the proposed models use fuzzy or intuitionistic fuzzy weighting ap-

proach to decide the contributions of the samples in the training process and hence, get better generalization performance than standard RVFL. The proposed models are robust to noise and outliers and give special attention to each sample based on their importance in classification problems. The proposed IFRVFL model is employed for the diagnosis of AD and has shown the highest performance in MCI versus AD cases. As per the literature, discrimination between MCI versus AD cases is hard compared to other cases of AD. The proposed CP-FRVFL model uses a novel score function to determine the importance of each sample in the training process and the experimental results on KEEL and UCI datasets demonstrate that the proposed CP-FRVFL model has superior performance than the compared models. Moreover, the proposed GE-IFWRVFL model employs graph embedding approach that uses intrinsic and penalty graphs to describe the geometrical properties of the data. Unlike RVFL and IFRVFL models, the proposed GE-IFWRVFL model considers the geometrical relationship of the data while calculating the final output parameters. The experimental results demonstrate that the proposed GE-IFWRVFL model has better generalization performance with the highest average accuracy and lowest average rank compared to the baseline models.

Third, we proposed two novel models that combine hyperplane-based approaches with RdNN techniques. In one approach, we integrate NPSVM with the RVFL network and hence, the hybrid model, i.e., NPRVFL, has better generalization performance than the baseline models. Unlike NPSVM, the proposed NPRVFL model gets the hyperplanes in randomized features space and hence, takes the benefits of RVFL's features space to classify the data. The proposed NPRVFL's hyperplanes pass through the origin. The results show that the proposed NPRVFL has superior performance than the NPSVM and RVFL models. In another approach, we incorporate hybrid features (original features and autoencoder-based features) into LSTSVM and RELS-TSVM, and propose extended LSTSVM and extended RELS-TSVM models. Experimental results and statistical tests demonstrate that the proposed models show better generalization performance.

Fourth, we proposed an extended feature RVFL (efRVFL) model that is trained on higher dimensional space (extended feature space) generated from the original feature space analytically. The proposed efRVFL model consists of additional non-linear transformed features

and hence, is capable of capturing the highly non-linear patterns within datasets. Instead of stacking the several hidden layers (for higher representation learning) in the RVFL model, we employed a simple linear regression method for generating supervised randomized features (new features) from original features and hence, the proposed efRVFL model has better representative features. Ensemble learning develops more stable and robust model that has better generalization performance than a single model. The proposed efRVFL model is an unstable classifier. Therefore, we developed an ensemble of extended feature RVFL (en-efRVFL) model. The proposed en-efRVFL model trains each base model (efRVFL) over different feature spaces so that accurate and diverse base models can be generated which leads to better accuracy. The outcomes of base models are integrated via the average voting scheme. Empirical results show that the proposed efRVFL and en-efRVFL models perform better than the standard RVFL model and have competitive performance with several compared deep models. The proposed en-efRVFL model has the lower rank among all the compared state-of-the-art deep feed-forward neural networks. Ensemble learning is more beneficial with unstable models (low bias, high variance) such as decision trees and neural networks. RVFL having randomized features is an unstable classifier. Therefore, a novel neural network-based ensemble model (RoF-RVFL) has been proposed wherein the unsupervised feature extraction technique PCA is employed to extract new features from the original datasets. The proposed RoF-RVFL model uses RVFL as its base model since it is sensitive to changes in the data set. To evaluate the performance of the proposed RoF-RVFL model, extensive experiments have been conducted over 42 benchmark datasets and the experimental results illustrate that RoF-RVFL performs better than baseline algorithms (that is RaF, RoF, and RVFL) in terms of classification accuracy.

Fifth, we proposed extended graph embedded RVFL (EGERVFL) model, that can be considered as a variant of standard RVFL. Unlike standard RVFL, the proposed EGERVFL model is trained over extended feature space and hence, possesses more capability to capture the nonlinear hidden relationships within dataset. Moreover, we used the graph regularization term in the optimization problem of the proposed EGERVFL model under the graph embedded framework, that incorporates the geometrical relationship within data. Shallow networks are not good in representation learning. Therefore, we extended the proposed shallow

EGERVFL model to ensemble deep framework and named it, the ensemble deep EGERVFL (edEGERVFL) model. The proposed edEGERVFL model solves graph embedded-based optimization problem in each layer and hence, has better generalization performance. To evaluate the proposed models, we employed them for the diagnosis of AD. The experimental results demonstrate good generalization of the proposed models. The proposed EGERVFL model (with MDA, LFDA, and LDA) has better classification performance compared to the standard RVFL. The proposed edEGERVFL model (with MDA and LFDA) has the overall best performance among the compared models on CN vs AD and CN vs MCI cases. Moreover, the proposed models are implemented over UCI datasets and the results demonstrate that the proposed EGERVFL and its ensemble deep variant edEGERVFL with LFDA have better generalization performance compared to the other proposed models. Table 7.1 and Table 7.2 discuss the advantages and limitations of the proposed models in this thesis.

Table 7.1: Advantages and limitations of the models developed in Chapter 3 and Chapter 4.

	IFRVFL	GE-IFWRVFL	CP-FRVFL	ext-LSTSVM	ext-RELS-TSVM	NPRVFL
Advantage	Uses closed-form solution approach. Efficient and robust to noise and outliers	Effectively handles geometrical aspects of the data as compared to RVFL. Efficient and robust to noise and outliers	Efficient and robust to noise and outliers	Unlike LSTSVM, it uses original and autoencoder-based features and hence captures linear and nonlinear patterns simultaneously	Unlike RELS-TSVM, it uses original and autoencoder-based features and hence captures linear and nonlinear patterns simultaneously	Unlike RVFL and TWSVM, it doesn't need to calculate the inverse of matrices to calculate the final output parameters
Limitations	Unstable classifier. Applicable only for binary classification problems and not suitable for large datasets.	Not suitable for large datasets	Not suitable for large datasets	Not applicable when number of samples and features are large enough, sensitive to noise and outliers	Not applicable when number of samples and features are large enough, sensitive to noise and outliers	Applicable only for binary classification problems, Sensitive to outliers

This thesis improved the RVFL model in multiple aspects. However, there is still room to grow and much more research can be done. Next, we give potential future research directions.

7.2 Future research directions

While reviewing the RVFL's papers in the literature, we found some potential research directions that the researchers in the future should explore.

Table 7.2: Advantages and limitations of the models developed in Chapter 5 and Chapter 6.

	efRVFL	RoF-RVFL	en-efRVFL	EGERVFL	edEGERVFL
Advantage	Efficient and uses closed-form solution approach	Stable and generates uncorrelated features	Stable and robust	Efficient and handle topological properties of data effectively	Handle topological properties of data, deep architecture, deep feature representation
Limitations	Sensitive to noise and outliers	Sensitive to outliers	Not applicable when number of samples and features are large enough	Shallow architecture and applicable to only binary classification	Sensitive to noise and outliers

- Weights initialization techniques (WITs) have significant impact on the performance of RVFL models. A few research [12, 121, 136, 137] suggest that this topic needs to be discussed further with mathematical justifications. Moreover, several other strategies [368] can be explored with RVFL model such as interval-based initialization [369], variance scaling based initialization [370], data-driven initialization [371], hybrid initialization [372, 373], cluster-based initialization [374] and data statistics based initialization [375].

- Outliers or noisy samples influence the modeling capability of standard RVFL and hence, lead to poor performance. Kernalized RVFL models are robust but can't be employed for large-scale dataset. Therefore, for large scale, different techniques such as random Fourier features [376] can be used to handle the same [377]. Moreover, RVFL with fuzzy neural networks [378], ensemble learning [38], or other advanced techniques can be employed to develop robust RVFL variants.

- Ensemble learning and deep learning are two individual growing fields [132]. Researchers have recently combined them to develop a more accurate and efficient model that can perform well on real-world data. The RVFL model has fast training speed and good generalization performance and has been employed successfully in various engineering domains. Therefore, this can be a hot topic for researchers to explore RVFL in these research directions.

- In the literature [43, 127], RVFL and decision tree have been employed together to

develop a model with better performance. Recently, deep forest [379, 380] with better interpretability and less tunable parameters as compared to deep neural networks (DNNs) is a growing research field. Studying the RVFL model and decision tree with deep forest architecture can be a new research field.

- To increase the generalization performance of machine learning models, learning with global/local data consistency (topological properties of data) has shown its importance among the machine learning community. The RVFL model transforms the original features into randomized features in an unsupervised manner. Hence, the randomization process might ruin the original feature space's topological properties and lead to an inefficient model. Therefore, works [143, 191] indicate that incorporating the idea of manifold learning into the RVFL model can develop more accurate models.

- The standard RVFL handles balanced data effectively. The imbalance learning problem seriously deteriorates the performance of the RVFL model. In general, techniques addressing imbalance data are divided into two categories, i.e., data-level approach [187] and algorithm approaches [381] can be used with RVFL model to classify imbalance data, effectively. Therefore, it is an opportunity for researchers to develop other techniques to explore this research direction.

- Developments in the RVFL model have been focused on supervised problems, i.e., classification and regression problems. In the real-world, unlabeled data consists of only a few labeled samples but many unlabeled samples. There is very less work with RVFL to handle semi-supervised problems. Therefore, it is desirable to develop variants of the RVFL model that can be employed for semi-supervised problems effectively.

- The model pool of ensemble RVFL mainly consists of learning algorithms. However, statistical methods can be included and improve the performance. For instance, statistical forecasting methods, such as ARIMA and exponential smoothing can be included in the model pool for forecasting tasks. Therefore, the models' diversity is increased significantly.

- Most deep RVFLs networks are designed based on the conventional feed-forward architecture. However, the deep learning community has proposed various advanced architectures, such as the Transformer [382] and graph convolution neural network, etc. Combining the advanced architectures and the idea of deep RVFL may maintain high performance and reduce training time simultaneously.

- The existing literature utilizes static aggregation for the ensemble block of edRVFL. However, such a static ensemble does not consider the evolving characteristic or the concept drift problem. Recently, a dynamic ensemble algorithm that computes the ensemble weights based on the latest accuracy is proposed for forecasting by Liang et al. [383]. The output layers of edRVFL can be considered as different models. Hence, the dynamic ensemble can be applied to combine all output layers' forecasts. Therefore, a dynamic ensemble that assigns evolving weights can be combined with edRVFL to boost the performance further.

- The tuning process of edRVFL imposes a significant effect on the performance. A layer-wise tuning algorithm is proposed for time series forecasting in [253]. Such a tuning procedure benefits the diverse and optimal architecture of the edRVFL. However, Gao et al. [253] only implements a layer-wise tuning algorithm with Bayesian optimization. In the future, more advanced optimization algorithms can be combined with layer-wise tunings, such as evolutionary algorithms [384]. The marriage of layer-wise tuning and advanced optimization algorithms will develop the RVFLs into auto-ML in the future.

- Although the random features offer non-linearity and fast computation, the random nature carries redundant information. Therefore, an intelligent selection of the random features owns the strong potential to increase the performance [59, 254]. The inferior features of random layers may deteriorate the performance. The existing works only consider linear feature selection, and pruning techniques [59, 254]. However, there are more advanced feature selection algorithms [385]. The effects of applying different feature selection algorithms on the random features must be studied.

- Although RVFL and its variants show superior forecasting, there are still directions worth exploring. For instance, the augmentation of RVFL's random features is not mature yet, although signal decomposition shows its effectiveness. If the decomposition is done correctly, the elements generated are always high-dimensional. In RVFL and deep RVFL, the direct links are connected to the linear output layers. Therefore, effective treatment of such high-dimensional features is critical. Potential solutions can be dimensionality reduction, feature selection and double regularizations. Dimensionality reduction algorithms can be utilized to transform the huge input feature matrix into a low-dimensional space. Then, a linear layer is trained in the low-dimensional space. Feature selection only selects a few best features for the linear layer to learn. As for the double regularizations, different regularizations are imposed on the direct link and random features. If the direct link is of high dimension, its regularization would prefer sparsity.

- RVFL's forecasting ability on spatial-temporal time series is not investigated yet. The spatial-temporal time series is a temporal sequence of graph signals. However, the conventional version of RVFL is not suitable for graph data. Therefore, it is a promising direction to develop RVFL for graph data.

- Recently, multi-label learning has been emerging as an exciting research domain. Therefore, researchers may develop randomized neural networks to handle multi-label data. RVFL model doesn't have enough work to manage the multi-label datasets. The efficient and effective variants of the RVFL model should be developed for multi-label tasks.

- Unlike supervised learning, unsupervised learning problems doesn't have target variable information. Standard RVFL model needs target information while calculating the final parameters. Unlabeled data are clustered (or grouped) by considering their topological properties or other properties of the data. Therefore, the needful works are required to handle unlabeled data via RVFL model.

- The community lacks a thorough investigation that compares the performance of ran-

domised neural networks on datasets that are openly accessible, utilising standardised metrics, evaluation procedures, and several datasets. A benchmark for comparing the various randomised architectures is thus needed in the field. This will encourage future efforts to improve a randomised model from various angles, including precision, trustworthiness, and training/inference efficiency.

- Most of the RVFL-based architectures are based on the offline training wherein all the data is available for the training at once. However, in online scenarios the sequential streaming data needs to be processed. RVFL models can be adapted to handle such scenarios. Moreover, one can also focus on the development of deep RVFL architectures for the online learning process. As for the edRVFL, the output layers can be trained in an online fashion and the ensemble can be online, too.

Bibliography

[1] C. Zhang, Y. Lu, Study on artificial intelligence: The state of the art and future prospects, Journal of Industrial Information Integration 23 (2021) 100224.

[2] A. K. Jain, J. Mao, K. M. Mohiuddin, Artificial neural networks: A tutorial, Computer 29 (1996) 31–44.

[3] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (2018).

[4] L. Zhang, P. N. Suganthan, A survey of randomized algorithms for training neural networks, Information Sciences 364 (2016) 146–155.

[5] P. N. Suganthan, On non-iterative learning algorithms with closed-form solution, Applied Soft Computing 70 (2018) 1078–1082.

[6] W. Cao, X. Wang, Z. Ming, J. Gao, A review on neural networks with random weights, Neurocomputing 275 (2018) 278–287.

[7] B. Widrow, A. Greenblatt, Y. Kim, D. Park, The no-prop algorithm: A new learning algorithm for multilayer neural networks, Neural Networks 37 (2013) 182–188.

[8] J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, Neural Computation 3 (1991) 246–257.

[9] F. Scarselli, A. C. Tsoi, Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results, Neural Networks 11 (1998) 15–37.

[10] F. M. A. Acosta, Radial basis function and related models: an overview, Signal Processing 45 (1995) 37–58.

[11] Y.-H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (1992) 76–79.

[12] L. Zhang, P. N. Suganthan, A comprehensive evaluation of random vector functional link networks, Information Sciences 367 (2016) 1094–1105.

[13] N. Vuković, M. Petrović, Z. Miljković, A comprehensive experimental evaluation of orthogonal polynomial expanded random vector functional link neural networks for regression, Applied Soft Computing 70 (2018) 1083–1096.

[14] G. Dudek, Are direct links necessary in random vector functional link networks for regression?, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2020, pp. 60–70.

[15] D. S. Broomhead, D. Lowe, Radial basis functions, multi-variable functional interpolation and adaptive networks, Technical Report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[16] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.

[17] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[18] V. Vapnik, The nature of statistical learning theory, Springer science & business media, 1999.

[19] Z. Gao, Y. Wang, M. Huang, J. Luo, S. Tang, A kernel-free fuzzy reduced quadratic surface ν-support vector machine with applications, Applied Soft Computing 127 (2022) 109390.

[20] B. Gaonkar, C. Davatzikos, Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification, Neuroimage 78 (2013) 270–283.

[21] E. Byvatov, G. Schneider, Support vector machine applications in bioinformatics., Applied bioinformatics 2 (2003) 67–77.

[22] R. Khemchandani, S. Chandra, et al., Twin support vector machines for pattern classification, IEEE Transactions on pattern analysis and machine intelligence 29 (2007) 905–910.

[23] M. A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert systems with applications 36 (2009) 7535–7543.

[24] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, N.-Y. Deng, Improvements on twin support vector machines, IEEE transactions on neural networks 22 (2011) 962–968.

[25] J. A. Nasiri, N. M. Charkari, K. Mozafari, Energy-based model of least squares twin support vector machines for human action recognition, Signal Processing 104 (2014) 248–257.

[26] M. Tanveer, M. A. Khan, S.-S. Ho, Robust energy-based least squares twin support vector machines, Applied Intelligence 45 (2016) 174–186.

[27] S. Rezvani, X. Wang, F. Pourpanah, Intuitionistic fuzzy twin support vector machines, IEEE Transactions on Fuzzy Systems 27 (2019) 2140–2151.

[28] H. Huang, X. Wei, Y. Zhou, Twin support vector machines: A survey, Neurocomputing 300 (2018) 34–43.

[29] M. Tanveer, T. Rajani, R. Rastogi, Y.-H. Shao, M. Ganaie, Comprehensive review on twin support vector machines, Annals of Operations Research (2022) 1–46.

[30] M. Tanveer, C. Gautam, P. N. Suganthan, Comprehensive evaluation of twin svm based classifiers on uci datasets, Applied Soft Computing 83 (2019) 105617.

[31] Y. Wan, S. Song, G. Huang, S. Li, Twin extreme learning machines for pattern classification, Neurocomputing 260 (2017) 235–244.

[32] P. Borah, D. Gupta, Unconstrained convex minimization based implicit lagrangian twin random vector functional-link networks for binary classification (ultrvflc), Applied Soft Computing 81 (2019) 105534.

[33] X. Huang, L. Shi, J. A. Suykens, Support vector machine classifier with pinball loss, IEEE transactions on pattern analysis and machine intelligence 36 (2013) 984–997.

[34] N. Deng, Y. Tian, C. Zhang, Support vector machines: optimization based theory, algorithms, and extensions, CRC press, 2012.

[35] Y. Tian, Z. Qi, X. Ju, Y. Shi, X. Liu, Nonparallel support vector machines for pattern classification, IEEE transactions on cybernetics 44 (2013) 1067–1079.

[36] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Frontiers of Computer Science 14 (2020) 241–258.

[37] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2018) e1249.

[38] Y. Ren, L. Zhang, P. N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions, IEEE Computational Intelligence Magazine 11 (2016) 41–53.

[39] Y. Zhao, J. Gao, X. Yang, A survey of neural network ensembles, in: 2005 International Conference on Neural Networks and Brain, volume 1, IEEE, 2005, pp. 438–442.

[40] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[41] T. K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 832–844.

[42] J. J. Rodriguez, L. I. Kuncheva, C. J. Alonso, Rotation forest: A new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1619–1630.

[43] R. Katuwal, P. N. Suganthan, L. Zhang, An ensemble of decision trees with random vector functional link networks for multi-class classification, Applied Soft Computing 70 (2018) 1146–1153.

[44] R. Katuwal, P. N. Suganthan, Dropout and dropconnect based ensemble of random vector functional link neural network, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 1772–1778.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, B. Mele, G. Altarelli, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (2014) 1929–1958. doi:10.1016/0370-2693(93)90272-J.

[46] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, R. Fergus, Regularization of neural networks using dropconnect, in: International Conference on Machine Learning, 2013, pp. 1058–1066.

[47] Q.-H. Ling, Y.-Q. Song, F. Han, D. Yang, D.-S. Huang, An improved ensemble of random vector functional link networks based on particle swarm optimization with double optimization strategy, PloS one 11 (2016) e0165803.

[48] M. Hu, Q. Shi, P. N. Suganthan, M. Tanveer, Adaptive ensemble variants of random vector functional link networks, in: International Conference on Neural Information Processing, Springer, 2020, pp. 30–37.

[49] Z. Zhu, S. Wang, Y. Zhang, Roenet: A resnet-based output ensemble for malaria parasite classification, Electronics 11 (2022) 2040.

[50] Y. Ren, P. N. Suganthan, N. Srikanth, G. Amaratunga, Random vector functional link network for short-term electricity load demand forecasting, Information Sciences 367 (2016) 1078–1093.

[51] R. Sharma, T. Goel, M. Tanveer, P. N. Suganthan, I. Razzak, R. Murugan, Conv-eRVFL: Convolutional neural network based ensemble RVFL classifier for alzheimer's disease diagnosis, IEEE Journal of Biomedical and Health Informatics (2022).

[52] Y. Cao, T. A. Geddes, J. Y. H. Yang, P. Yang, Ensemble deep learning in bioinformatics, Nature Machine Intelligence 2 (2020) 500–508.

[53] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117.

[54] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[55] I. Goodfellow, Y. Bengio, A. Courville, Deep learning. book in preparation for MIT press, URL¡ http://www. deeplearningbook. org (2016).

[56] C. Gallicchio, S. Scardapane, Deep randomized neural networks, Recent Trends in Learning From Data (2020) 43–68.

[57] Q. Shi, R. Katuwal, P. N. Suganthan, M. Tanveer, Random vector functional link neural network based ensemble deep learning, Pattern Recognition 117 (2021) 107978.

[58] R. Katuwal, P. N. Suganthan, Stacked autoencoder based deep random vector functional link neural network for classification, Applied Soft Computing 85 (2019) 105854.

[59] Q. Shi, M. Hu, P. N. Suganthan, R. Katuwal, Weighting and pruning based ensemble deep random vector functional link network for tabular data classification, Pattern Recognition (2022) 108879.

[60] J. Tang, C. Deng, G.-B. Huang, Extreme learning machine for multilayer perceptron, IEEE Transactions on Neural Networks and Learning Systems 27 (2015) 809–821.

[61] R. Sharma, T. Goel, M. Tanveer, S. Dwivedi, R. Murugan, FAF-DRVFL: fuzzy activation function based deep random vector functional links network for early diagnosis of Alzheimer disease, Applied Soft Computing 106 (2021) 107371.

[62] W. X. Cheng, P. N. Suganthan, R. Katuwal, Time series classification using diversified ensemble deep random vector functional link and resnet features, Applied Soft Computing 112 (2021) 107826.

[63] D. R. Nayak, R. Dash, B. Majhi, R. B. Pachori, Y. Zhang, A deep stacked random vector functional link network autoencoder for diagnosis of brain abnormalities and breast cancer, Biomedical Signal Processing and Control 58 (2020) 101860.

[64] R. Gao, L. Du, K. F. Yuen, P. N. Suganthan, Walk-forward empirical wavelet random vector functional link for time series forecasting, Applied Soft Computing 108 (2021) 107450.

[65] J. Del Ser, D. Casillas-Perez, L. Cornejo-Bueno, L. Prieto-Godino, J. Sanz-Justo, C. Casanova-Mateo, S. Salcedo-Sanz, Randomization-based machine learning in renewable energy prediction problems: critical literature review, new results and perspectives, arXiv preprint arXiv:2103.14624 (2021).

[66] P. Zhou, Y. Jiang, C. Wen, X. Dai, Improved incremental RVFL with compact structure and its application in quality prediction of blast furnace, IEEE Transactions on Industrial Informatics (2021).

[67] T. Chakravorti, P. Satyanarayana, Non linear system identification using kernel based exponentially extended random vector functional link network, Applied Soft Computing 89 (2020) 106117.

[68] Y. Zhang, R. Guo, Y. Peng, W. Kong, F. Nie, B.-L. Lu, An auto-weighting incremental random vector functional link network for eeg-based driving fatigue detection, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–14.

[69] M. Ganaie, M. Tanveer, Lstsvm classifier with enhanced features from pre-trained functional link network, Applied Soft Computing 93 (2020) 106305.

[70] Y. Hu, B. Qu, J. Wang, J. Liang, Y. Wang, K. Yu, Y. Li, K. Qiao, Short-term load forecasting using multimodal evolutionary algorithm and random vector functional link network based ensemble learning, Applied Energy 285 (2021) 116415.

[71] L. Yu, Y. Wu, L. Tang, H. Yin, K. K. Lai, Investigation of diversity strategies in RVFL network ensemble learning for crude oil price forecasting, Soft Computing 25 (2021) 3609–3622.

[72] W. Wang, Y. Peng, W. Kong, EEG-based emotion recognition via joint domain adaptation and semi-supervised RVFL network, in: International Conference on Intelligent Automation and Soft Computing, Springer, 2021, pp. 413–422.

[73] M. F. Amasyali, Improved space forest: A meta ensemble method, IEEE Transactions on Cybernetics 49 (2018) 816–826.

[74] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2006) 40–51.

[75] A. K. Malik, R. Gao, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Random vector functional link network: recent developments, applications, and future directions, arXiv:2203.11316 (2022).

[76] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, A. D. N. I. Initiative, et al., Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network, IEEE Transactions on Computational Social Systems (2022).

[77] A. K. Malik, M. A. Ganaie, M. Tanveer, Graph embedded intuitionistic fuzzy weighted random vector functional link network, in: 2022 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2022, pp. 293–299.

[78] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Support vector machine based models with sparse auto-encoder based features for classification problem, in: International Conference on Neural Information Processing, Springer, 2022, pp. 248–259.

[79] A. K. Malik, M. Ganaie, M. Tanveer, P. N. Suganthan, Extended features based random vector functional link network for classification problem, IEEE Transactions on Computational Social Systems (2022).

[80] A. K. Malik, M. Ganaie, M. Tanveer, P. N. Suganthan, A novel ensemble method of rvfl for classification problem, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[81] A. K. Malik, M. Tanveer, Graph embedded ensemble deep randomized network for diagnosis of alzheimer's disease, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2022).

[82] S. Russell, Artificial intelligence: The future is superintelligent, Nature 548 (2017) 520–521.

[83] R. Gao, W. X. Cheng, P. Suganthan, K. F. Yuen, Inpatient discharges forecasting for Singapore hospitals by machine learning, IEEE Journal of Biomedical and Health Informatics (2022).

[84] M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 4189–4196.

[85] P. Giudici, E. Raffinetti, Shapley-lorenz explainable artificial intelligence, Expert Systems with Applications 167 (2021) 114104.

[86] R. Gao, R. Li, M. Hu, P. N. Suganthan, K. F. Yuen, Dynamic ensemble deep echo state network for significant wave height forecasting, Applied Energy 329 (2023) 120261.

[87] P. J. Lisboa, A review of evidence of health benefit from artificial neural networks in medical intervention, Neural Networks 15 (2002) 11–39.

[88] D. M. Himmelblau, Applications of artificial neural networks in chemical engineering, Korean Journal of Chemical Engineering 17 (2000) 373–392.

[89] S.-Y. King, J.-N. Hwang, Neural network architectures for robotic applications, IEEE Transactions on Robotics and Automation 5 (1989) 641–657.

[90] K. J. Hunt, D. Sbarbaro, R. Żbikowski, P. J. Gawthrop, Neural networks for control systems—a survey, Automatica 28 (1992) 1083–1112.

[91] S. Chen, S. A. Billings, P. Grant, Non-linear system identification using neural networks, International Journal of Control 51 (1990) 1191–1214.

[92] B. C. Csáji, Approximation with artificial neural networks, Faculty of Sciences, Etvs Lornd University, Hungary 24 (2001) 7.

[93] I. A. Basheer, M. Hajmeer, Artificial neural networks: fundamentals, computing, design, and application, Journal of Microbiological Methods 43 (2000) 3–31.

[94] M. Leshno, V. Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Networks 6 (1993) 861–867.

[95] Y. Chauvin, D. E. Rumelhart, Backpropagation: theory, architectures, and applications, Psychology Press, 2013.

[96] R. A. Jacobs, Increased rates of convergence through learning rate adaptation, Neural Networks 1 (1988) 295–307.

[97] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 76–86.

[98] G. D. Magoulas, M. N. Vrahatis, G. S. Androulakis, Improving the convergence of the backpropagation algorithm using learning rate adaptation methods, Neural Computation 11 (1999) 1769–1796.

[99] S. Scardapane, D. Wang, Randomness in neural networks: an overview, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 7 (2017) e1200.

[100] K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural networks 3 (1990) 551–560.

[101] K. Hornik, Some new results on neural network approximation, Neural networks 6 (1993) 1069–1072.

[102] J. Del Ser, D. Casillas-Perez, L. Cornejo-Bueno, L. Prieto-Godino, J. Sanz-Justo, C. Casanova-Mateo, S. Salcedo-Sanz, Randomization-based machine learning in renewable energy prediction problems: critical literature review, new results and perspectives, Applied Soft Computing (2022) 108526.

[103] P. N. Suganthan, R. Katuwal, On the origins of randomization-based feedforward neural networks, Applied Soft Computing 105 (2021) 107239.

[104] J. Del Ser, E. L. Manibardo, I. Laña, On the potential of randomization-based neural networks for driving behavior classification, in: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2022, pp. 2991–2997.

[105] R. Gao, R. Li, M. Hu, P. N. Suganthan, K. F. Yuen, Significant wave height forecasting using hybrid ensemble deep randomized networks with neurons pruning, Engineering Applications of Artificial Intelligence 117 (2023) 105535.

[106] J. Long, S. Zhang, C. Li, Evolving deep echo state networks for intelligent fault diagnosis, IEEE Transactions on Industrial Informatics 16 (2019) 4928–4937.

[107] W. Liu, C. Ren, Y. Xu, Missing-data tolerant hybrid learning method for solar power forecasting, IEEE Transactions on Sustainable Energy 13 (2022) 1843–1852.

[108] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.

[109] J. Zhang, G. G. Walter, Y. Miao, W. N. W. Lee, Wavelet neural networks for function learning, IEEE Transactions on Signal Processing 43 (1995) 1485–1497.

[110] Y.-H. Pao, G.-H. Park, D. J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (1994) 163–180.

[111] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (2009) 183–202.

[112] C. Huang, M. Li, F. Cao, H. Fujita, Z. Li, X. Wu, Are graph convolutional networks with random weights feasible?, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

[113] W. F. Schmidt, M. A. Kraaijveld, R. P. Duin, et al., Feed forward neural networks with random weights, in: International Conference on Pattern Recognition, IEEE Computer Society Press, 1992, pp. 1–1.

[114] Y.-H. Pao, S. M. Phillips, D. J. Sobajic, Neural-net computing and the intelligent control of systems, International Journal of Control 56 (1992) 263–289.

[115] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE transactions on systems, man, and cybernetics (1985) 116–132.

[116] C. G. Looney, Radial basis functional link nets and fuzzy reasoning, Neurocomputing 48 (2002) 489–509.

[117] B. Igelnik, Y.-H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE transactions on Neural Networks 6 (1995) 1320–1329.

[118] D. Needell, A. A. Nelson, R. Saab, P. Salanevich, Random vector functional link networks for function approximation on manifolds, arXiv preprint arXiv:2007.15776 (2020).

[119] R. Bisoi, P. Dash, S. Mishra, Modes decomposition method in fusion with robust random vector functional link network for crude oil price forecasting, Applied Soft Computing 80 (2019) 475–493.

[120] Y. Zhang, J. Wu, Z. Cai, B. Du, S. Y. Philip, An unsupervised parameter learning model for rvfl neural network, Neural Networks 112 (2019) 85–97.

[121] W. Cao, J. Gao, Z. Ming, S. Cai, H. Zheng, Impact of probability distribution selection on RVFL performance, in: International Conference on Smart Computing and Communication, Springer, 2017, pp. 114–124.

[122] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[123] Y. Freund, R. E. Schapire, Experiments with a new boosting algorithm, in: icml, volume 96, Citeseer, 1996, pp. 148–156.

[124] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[125] C. Zhang, F. Jiang, S. Wang, W. Shang, A novel hybrid approach with a decomposition method and the RVFL model for crude oil price prediction, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 4446–4449.

[126] J. Shi, Z. Xue, Y. Dai, B. Peng, Y. Dong, Q. Zhang, Y. Zhang, Cascaded multi-column RVFL+ classifier for single-modal neuroimaging-based diagnosis of Parkinson's disease, IEEE Transactions on Biomedical Engineering 66 (2018) 2362–2371.

[127] R. Katuwal, P. N. Suganthan, Enhancing multi-class classification of random forest using random vector functional neural network and oblique decision surfaces, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[128] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, Computational Intelligence and Neuroscience 2018 (2018).

[129] Y. Li, C. Huang, L. Ding, Z. Li, Y. Pan, X. Gao, Deep learning in bioinformatics: Introduction, application, and perspective in the big data era, Methods 166 (2019) 4–21.

[130] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, S. Kasaei, Deep learning for visual tracking: A comprehensive survey, IEEE Transactions on Intelligent Transportation Systems (2021).

[131] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: A systematic review, IEEE Access 7 (2019) 19143–19165.

[132] M. A. Ganaie, M. Hu, M. Tanveer, P. N. Suganthan, Ensemble deep learning: A review, arXiv preprint arXiv:2104.02395 (2021).

[133] C. R. Rao, S. K. Mitra, Further contributions to the theory of generalized inverse of matrices and its applications, Sankhyā: The Indian Journal of Statistics, Series A (1971) 289–300.

[134] W. Cao, L. Hu, J. Gao, X. Wang, Z. Ming, A study on the relationship between the rank of input data and the performance of random weight neural network, Neural Computing and Applications 32 (2020) 12685–12696.

[135] A. Rasheed, A. Adebisi, K. C. Veluvolu, Respiratory motion prediction with random vector functional link (RVFL) based neural networks, in: Journal of Physics: Conference Series, volume 1626, IOP Publishing, 2020, p. 012022.

[136] W. Cao, M. J. Patwary, P. Yang, X. Wang, Z. Ming, An initial study on the relationship between meta features of dataset and the initialization of NNRW, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[137] M. Tanveer, M. A. Ganaie, P. N. Suganthan, Ensemble of classification models with weighted functional link network, Applied Soft Computing 107 (2021) 107322.

[138] L. Fan, W. Sun, G. Feng, Image steganalysis via random subspace fisher linear discriminant vector functional link network and feature mapping, Mobile Networks and Applications 24 (2019) 1269–1278.

[139] J. Pan, J. Fan, A. Dong, Y. Li, Random vector functional link network optimized by jaya algorithm for transient stability assessment of power systems, Mathematical Problems in Engineering 2020 (2020).

[140] R. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, International Journal of Industrial Engineering Computations 7 (2016) 19–34.

[141] S.-Y. Lu, S.-H. Wang, Y.-D. Zhang, A classification method for brain MRI via mobilenet and feedforward network with random weights, Pattern Recognition Letters 140 (2020) 252–260.

[142] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[143] X. Li, Y. Yang, N. Hu, Z. Cheng, J. Cheng, Discriminative manifold random vector functional link neural network for rolling bearing fault diagnosis, Knowledge-Based Systems 211 (2021) 106507.

[144] X. Guo, W. Zhou, Q. Lu, A. Du, Y. Cai, Y. Ding, Assessing dry weight of hemodialysis patients via sparse Laplacian regularized RVFL neural network with $L_{2,1}$-norm, BioMed Research International 2021 (2021).

[145] M. A. Ganaie, M. Tanveer, P. N. Suganthan, Minimum variance embedded random vector functional link network, in: International Conference on Neural Information Processing, Springer, 2020, pp. 412–419.

[146] S. Parija, R. Bisoi, P. Dash, M. Sahani, Deep long short term memory based minimum variance kernel random vector functional link network for epileptic EEG signal classification, Engineering Applications of Artificial Intelligence 105 (2021) 104426.

[147] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

[148] M. A. Ganaie, M. Tanveer, P. N. Suganthan, Co-trained random vector functional link network, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[149] P. Dai, F. Gwadry-Sridhar, M. Bauer, M. Borrie, X. Teng, Healthy cognitive aging: a hybrid random vector functional-link model for the analysis of Alzheimer's disease, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 4567–4573.

[150] W. Cui, L. Zhang, B. Li, J. Guo, W. Meng, H. Wang, L. Xie, Received signal strength based indoor positioning using a random vector functional link network, IEEE Transactions on Industrial Informatics 14 (2017) 1846–1855.

[151] D. Samal, P. K. Dash, R. Bisoi, Modified added activation function based exponential robust random vector functional link network with expanded version for nonlinear system identification, Applied Intelligence (2021) 1–27.

[152] W. Dai, Q. Chen, F. Chu, X. Ma, T. Chai, Robust regularized random vector functional link network and its industrial application, IEEE Access 5 (2017) 16162–16172.

[153] J. Chen, S. Yang, D. Zhang, Y. Nanehkaran, A turning point prediction method of stock price based on RVFL-GMDH and chaotic time series analysis, Knowledge and Information Systems (2021) 1–26.

[154] J.-A. Mueller, A. Ivachnenko, F. Lemke, GMDH algorithms for complex systems modelling, Mathematical and Computer Modelling of Dynamical Systems 4 (1998) 275–316.

[155] P. Zhou, Y. Lv, H. Wang, T. Chai, Data-driven robust RVFLNs modeling of a blast furnace iron-making process using cauchy distribution weighted m-estimation, IEEE Transactions on Industrial Electronics 64 (2017) 7141–7151.

[156] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, for the Alzheimer's Disease Neuroimaging Initiative, Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network, IEEE Transactions on Computational Social Systems (2022). doi:10.1109/TCSS.2022.3146974.

[157] A. K. Malik, M. A. Ganaie, M. Tanveer, Graph embedded intuitionistic fuzzy weighted random vector functional link network, in: IEEE Symposium Series on Computational Intelligence, IEEE, 2022.

[158] N. Ahmad, M. A. Ganaie, A. K. Malik, K. T. Lai, M. Tanveer, Minimum variance

embedded intuitionistic fuzzy weighted random vector functional link network, in: International Conference on Neural Information Processing, IEEE, 2022.

[159] B. B. Hazarika, D. Gupta, 1-norm random vector functional link networks for classification problems, Complex & Intelligent Systems (2022) 1–17.

[160] P.-B. Zhang, Z.-X. Yang, A new learning paradigm for random vector functional-link network: RVFL+, Neural Networks 122 (2020) 94–105.

[161] D. R. Nayak, R. Dash, B. Majhi, U. R. Acharya, Application of fast curvelet tsallis entropy and kernel random vector functional link network for automated detection of multiclass brain abnormalities, Computerized Medical Imaging and Graphics 77 (2019) 101656.

[162] S. Parija, P. K. Dash, R. Bisoi, Multi-kernel-based random vector functional link network with decomposed features for epileptic EEG signal classification, IET Signal Processing 14 (2020) 162–174.

[163] S. Scardapane, D. Wang, A. Uncini, Bayesian random vector functional-link networks for robust data modeling, IEEE Transactions on Cybernetics 48 (2017) 2049–2059.

[164] H. Ye, F. Cao, D. Wang, A hybrid regularization approach for random vector functional-link networks, Expert Systems with Applications 140 (2020) 112912.

[165] A. Alalimi, L. Pan, M. A. Al-Qaness, A. A. Ewees, X. Wang, M. Abd Elaziz, Optimized random vector functional link network to predict oil production from tahe oil field in china, Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles 76 (2021) 3.

[166] W. Dai, Y. Ao, L. Zhou, P. Zhou, X. Wang, Incremental learning paradigm with privileged information for random vector functional-link networks: IRVFL+, Neural Computing and Applications (2022) 1–13.

[167] W. Jiao, S. Song, H. Han, W. Wang, Q. Zhang, Artificially intelligent differential diagnosis of enlarged lymph nodes with random vector functional link network plus, Medical Engineering & Physics (2022) 103939.

[168] V. H. A. Ribeiro, R. Santana, G. Reynoso-Meza, Random vector functional link forests and extreme learning forests applied to uav automatic target recognition, Engineering Applications of Artificial Intelligence 117 (2023) 105538.

[169] M. Abd Elaziz, S. Senthilraja, M. E. Zayed, A. H. Elsheikh, R. R. Mostafa, S. Lu, A new random vector functional link integrated with mayfly optimization algorithm for performance prediction of solar photovoltaic thermal collector combined with electrolytic hydrogen production system, Applied Thermal Engineering 193 (2021) 117055.

[170] A. H. Elsheikh, T. A. Shehabeldeen, J. Zhou, E. Showaib, M. Abd Elaziz, Prediction of laser cutting parameters for polymethylmethacrylate sheets using random vector functional link network integrated with equilibrium optimizer, Journal of Intelligent Manufacturing 32 (2021) 1377–1388.

[171] G. A. Ruz, P. A. Henríquez, Random vector functional link with naive bayes for classification problems of mixed data, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2019, pp. 1749–1752.

[172] P. A. Henríquez, G. A. Ruz, A non-iterative method for pruning hidden neurons in neural networks with random weights, Applied Soft Computing 70 (2018) 1109–1121.

[173] M. Pratama, P. P. Angelov, E. Lughofer, M. J. Er, Parsimonious random vector functional link network for data streams, Information Sciences 430 (2018) 519–537.

[174] Y. Wang, D. Gong, L. Pang, D. Yang, RVFL-based optical fiber intrusion signal recognition with multi-level wavelet decomposition as feature, Photonic Sensors 8 (2018) 234–241.

[175] E. M. El-Said, M. Abd Elaziz, A. H. Elsheikh, Machine learning algorithms for improving the prediction of air injection effect on the thermohydraulic performance of shell and tube heat exchanger, Applied Thermal Engineering 185 (2021) 116471.

[176] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (1999) 293–300.

[177] M.-L. Zhang, Z.-H. Zhou, A k-nearest neighbor based algorithm for multi-label classification, in: 2005 IEEE International Conference on Granular Computing, volume 2, IEEE, 2005, pp. 718–721.

[178] D. Candel, R. Ñanculef, C. Concha, H. Allende, A sequential minimal optimization algorithm for the all-distances support vector machine, in: Iberoamerican Congress on Pattern Recognition, Springer, 2010, pp. 484–491.

[179] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (2004) 199–222.

[180] L. Tang, C. Zhang, L. Li, S. Wang, A multi-scale method for forecasting oil price with multi-factor search engine data, Applied Energy 257 (2020) 114033.

[181] S. S. S. Mary, A. Sasithradevi, S. M. M. Roomi, J. J. Immanuvel, A random vector functional link network based content based image retrieval, in: 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), volume 1, IEEE, 2019, pp. 486–492.

[182] Z. Zhou, B. Wu, Adaptive sliding mode control of manipulators based on fuzzy random vector function links for friction compensation, Optik 227 (2021) 166055.

[183] E. T. Elkabbash, R. R. Mostafa, S. I. Barakat, Android malware classification based on random vector functional link and artificial jellyfish search optimizer, PloS One 16 (2021) e0260232.

[184] Z. Zhou, Z. Ma, Y. Wang, Z. Zhu, Fabric wrinkle rating model based on resnet18 and optimized random vector functional-link network, Textile Research Journal 93 (2023) 172–193.

[185] J. Huang, C.-M. Vong, W. Qian, Q. Huang, Y. Zhou, Online label distribution learning using random vector functional-link network, IEEE Transactions on Emerging Topics in Computational Intelligence (2022).

[186] W. Cao, P. Yang, Z. Ming, S. Cai, J. Zhang, An improved fuzziness based random vector functional link network for liver disease detection, in: 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, 2020, pp. 42–48.

[187] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[188] W. Cao, J. Gao, Z. Ming, S. Cai, Z. Shan, Fuzziness based random vector functional-link network for semi-supervised learning, in: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, 2017, pp. 782–786.

[189] M. Sahani, P. K. Dash, FPGA-based online power quality disturbances monitoring using reduced-sample HHT and class-specific weighted RVFLN, IEEE Transactions on Industrial Informatics 15 (2019) 4614–4623.

[190] V. Chauhan, A. Tiwari, Randomized neural networks for multilabel classification, Applied Soft Computing (2021) 108184.

[191] M. Ganaie, M. Tanveer, A. Malik, P. N. Suganthan, Minimum variance embedded random vector functional link network with privileged information, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022.

[192] C. Zhang, S.-K. Oh, W. Pedrycz, Z. Fu, S. Lu, Reinforced fuzzy clustering-based rule model constructed with the aid of exponentially weighted L2 regularization strategy and augmented random vector functional link network, Fuzzy Sets and Systems (2021). doi:10.1016/j.fss.2021.09.022.

[193] M. E. Zayed, J. Zhao, W. Li, A. H. Elsheikh, M. Abd Elaziz, D. Yousri, S. Zhong, Z. Mingxi, Predicting the performance of solar dish stirling power plant using a hybrid random vector functional link/chimp optimization model, Solar Energy 222 (2021) 1–17.

[194] D. R. Dash, P. Dash, R. Bisoi, Short term solar power forecasting using hybrid minimum variance expanded RVFLN and sine-cosine levy flight PSO algorithm, Renewable Energy 174 (2021) 513–537.

[195] B. B. Hazarika, D. Gupta, Modelling and forecasting of Covid-19 spread using wavelet-coupled random vector functional link networks, Applied Soft Computing 96 (2020) 106626.

[196] M. Abd Elaziz, T. A. Shehabeldeen, A. H. Elsheikh, J. Zhou, A. A. Ewees, M. A. Alqaness, Utilization of random vector functional link integrated with marine predators algorithm for tensile behavior prediction of dissimilar friction stir welded aluminum alloy joints, Journal of Materials Research and Technology 9 (2020) 11370–11381.

[197] F. Essa, M. Abd Elaziz, A. H. Elsheikh, Prediction of power consumption and water productivity of seawater greenhouse system using random vector functional link network integrated with artificial ecosystem-based optimization, Process Safety and Environmental Protection 144 (2020) 322–329.

[198] S. W. Sharshir, M. Abd Elaziz, M. Elkadeem, Enhancing thermal performance and modeling prediction of developed pyramid solar still utilizing a modified random vector functional link, Solar Energy 198 (2020) 399–409.

[199] A. M. Hussein, M. Abd Elaziz, M. S. A. Wahed, M. Sillanpää, A new approach to predict the missing values of algae during water quality monitoring programs based on a hybrid moth search algorithm and the random vector functional link network, Journal of Hydrology 575 (2019) 852–863.

[200] P. Zhou, Y. Jiang, C. Wen, T. Chai, Data modeling for quality prediction using im-

proved orthogonal incremental random vector functional-link networks, Neurocomputing 365 (2019) 1–9.

[201] Y. Dash, S. K. Mishra, S. Sahany, B. K. Panigrahi, Indian summer monsoon rainfall prediction: a comparison of iterative and non-iterative approaches, Applied Soft Computing 70 (2018) 1122–1134.

[202] M. F. Nhabangue, G. Pillai, Wind speed forecasting using improved random vector functional link network, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 1744–1750.

[203] K.-K. Xu, H.-X. Li, H.-D. Yang, Kernel-based random vector functional-link network for fast learning of spatiotemporal dynamic processes, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (2017) 1016–1026.

[204] L. Zhang, P. Zhou, M. Yuan, T.-y. Chai, Multivariable dynamic modeling for molten iron quality using incremental random vector functional-link networks, Journal of Iron and Steel Research International 23 (2016) 1151–1159.

[205] P. Zhou, M. Yuan, H. Wang, Z. Wang, T.-Y. Chai, Multivariable dynamic modeling for molten iron quality using online sequential random vector functional-link networks with self-feedback connections, Information Sciences 325 (2015) 237–255.

[206] Y. Ren, X. Qiu, P. N. Suganthan, G. Amaratunga, Detecting wind power ramp with random vector functional link (RVFL) network, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE, 2015, pp. 687–694.

[207] Y. Peng, Q. Li, W. Kong, F. Qin, J. Zhang, A. Cichocki, A joint optimization framework to semi-supervised RVFL and ELM networks for efficient data classification, Applied Soft Computing 97 (2020) 106756.

[208] J. Wang, X. Shen, W. Pan, On transductive support vector machines, Contemporary Mathematics 443 (2007) 7–20.

[209] S. Scardapane, D. Comminiello, M. Scarpiniti, A. Uncini, A semi-supervised random vector functional-link network based on the transductive framework, Information Sciences 364 (2016) 156–166.

[210] J. Xie, S. Liu, H. Dai, Y. Rong, Distributed semi-supervised learning algorithms for random vector functional-link networks with distributed data splitting across samples and features, Knowledge-Based Systems 195 (2020) 105577.

[211] A. K. Jain, R. C. Dubes, Algorithms for clustering data, Prentice-Hall, Inc., 1988.

[212] M. Tanveer, T. Gupta, M. Shah, A. D. N. Initiative, Pinball loss twin support vector clustering, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 17 (2021) 1–23.

[213] B. Richhariya, M. Tanveer, A. D. N. Initiative, et al., Least squares projection twin support vector clustering (lsptsvc), Information Sciences (2020).

[214] Y. Zhang, Q. Zhu, Y. Peng, W. Kong, An unsupervised discriminative random vector functional link network for efficient data clustering, in: 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), IEEE, 2021, pp. 347–352.

[215] Z.-H. Zhou, Ensemble learning, in: Machine Learning, Springer, 2021, pp. 181–210.

[216] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, A novel ensemble method of RVFL for classification problem, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[217] L. Zhang, X. Zhang, H. Chen, H. Tang, A robust temperature prediction model of shuttle kiln based on ensemble random vector functional link network, Applied Thermal Engineering 150 (2019) 99–110.

[218] A. Tahir, G. Morison, D. A. Skelton, R. M. Gibson, A novel functional link network stacking ensemble with fractal features for multichannel fall detection, Cognitive Computation 12 (2020) 1024–1042.

[219] X. Qiu, P. N. Suganthan, A. G. Amaratunga, Ensemble incremental random vector functional link network for short-term crude oil price forecasting, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 1758–1763.

[220] X. Qiu, P. N. Suganthan, G. A. Amaratunga, Ensemble incremental learning random vector functional link network for short-term electric load forecasting, Knowledge-Based Systems 145 (2018) 182–196.

[221] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, H. H. Liu, The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454 (1998) 903–995.

[222] G. Rilling, P. Flandrin, P. Gonçalves, J. M. Lilly, Bivariate empirical mode decomposition, IEEE Signal Processing Letters 14 (2007) 936–939.

[223] Z. Wu, N. E. Huang, Ensemble empirical mode decomposition: a noise-assisted data analysis method, Advances in Adaptive Data Analysis 1 (2009) 1–41.

[224] M. E. Torres, M. A. Colominas, G. Schlotthauer, P. Flandrin, A complete ensemble empirical mode decomposition with adaptive noise, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 4144–4147.

[225] X. Qiu, P. N. Suganthan, G. A. Amaratunga, Fusion of multiple indicators with ensemble incremental learning techniques for stock price forecasting, Journal of Banking and Financial Technology 3 (2019) 33–42.

[226] W. X. Cheng, P. N. Suganthan, X. Qiu, R. Katuwal, Classification of stock market trends with confidence-based selective predictions, in: Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing, Springer, 2019, pp. 93–104.

[227] J. Gilles, Empirical wavelet transform, IEEE Transactions on Signal Processing 61 (2013) 3999–4010.

[228] K. Dragomiretskiy, D. Zosso, Variational mode decomposition, IEEE Transactions on Signal Processing 62 (2013) 531–544.

[229] X. Qiu, P. N. Suganthan, G. A. Amaratunga, Electricity load demand time series forecasting with empirical mode decomposition based random vector functional link network, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2016, pp. 001394–001399.

[230] L. Tang, Y. Wu, L. Yu, A randomized-algorithm-based decomposition-ensemble learning methodology for energy price forecasting, Energy 157 (2018) 526–538.

[231] L. Tang, Y. Wu, L. Yu, A non-iterative decomposition-ensemble learning paradigm using RVFL network for crude oil price forecasting, Applied Soft Computing 70 (2018) 1097–1108.

[232] L. Li, X. Qu, J. Zhang, H. Li, B. Ran, Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network, Applied Soft Computing 73 (2018) 921–932.

[233] H. Sun, W. Zhai, Y. Wang, L. Yin, F. Zhou, Privileged information-driven random network based non-iterative integration model for building energy consumption prediction, Applied Soft Computing 108 (2021) 107438.

[234] X. Qiu, Y. Ren, P. N. Suganthan, G. A. Amaratunga, Short-term wind power ramp forecasting with empirical mode decomposition based ensemble learning techniques, in: 2017 IEEE Symposium Series on computational intelligence (SSCI), IEEE, 2017, pp. 1–8.

[235] J. Wu, T. Zhou, T. Li, A hybrid approach integrating multiple ICEEMDANs, WOA, and RVFL networks for economic and financial time series forecasting, Complexity 2020 (2020).

[236] J. Wu, F. Miu, T. Li, Daily crude oil price forecasting based on improved CEEMDAN, SCA, and RVFL: a case study in WTI oil market, Energies 13 (2020) 1852.

[237] J. Lu, J. Ding, C. Liu, Y. Jin, Prediction of physical properties of crude oil based on ensemble random weights neural network, IFAC-PapersOnLine 51 (2018) 655–660.

[238] C. Lian, L. Zhu, Z. Zeng, Y. Su, W. Yao, H. Tang, Constructing prediction intervals for landslide displacement using bootstrapping random vector functional link networks selective ensemble with neural networks switched, Neurocomputing 291 (2018) 1–10.

[239] J. Lu, C. Li, J. Ding, Ensemble random weights neural network based online prediction model of the production rate for mineral beneficiation process, IFAC-PapersOnLine 51 (2018) 1–6.

[240] B. Miskony, D. Wang, A randomized algorithm for prediction interval using RVFL networks ensemble, in: International Conference on Neural Information Processing, Springer, 2017, pp. 51–60.

[241] D. P. Mesquita, J. P. P. Gomes, L. R. Rodrigues, S. A. Oliveira, R. K. Galvão, Building selective ensembles of randomization based neural networks with the successive projections algorithm, Applied Soft Computing 70 (2018) 1135–1145.

[242] C. Ren, Y. Xu, Y. Zhang, R. Zhang, A hybrid randomized learning system for temporal-adaptive voltage stability assessment of power systems, IEEE Transactions on Industrial Informatics 16 (2019) 3672–3684.

[243] Y. Xia, Y. Xu, B. Gou, A data-driven method for IGBT open-circuit fault diagnosis based on hybrid ensemble learning and sliding-window classification, IEEE Transactions on Industrial Informatics 16 (2019) 5223–5233.

[244] Y. Liu, W. Cao, Z. Ming, Q. Wang, J. Zhang, Z. Xu, Ensemble neural networks with random weights for classification problems, in: 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, 2020, pp. 1–5.

[245] L. Zhang, P. N. Suganthan, Benchmarking ensemble classifiers with novel co-trained kernel ridge regression and random vector functional link ensembles [research frontier], IEEE Computational Intelligence Magazine 12 (2017) 61–72.

[246] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Extended features based random vector functional link network for classification problem, IEEE Transactions on Computational Social Systems (2022) 1–10. doi:10.1109/TCSS.2022.3187461.

[247] S.-Y. Lu, D. R. Nayak, S.-H. Wang, Y.-D. Zhang, A cerebral microbleed diagnosis method via featurenet and ensembled randomized neural networks, Applied Soft Computing (2021) 107567.

[248] S. Chen, J. Gao, R. Zhao, H. Fu, Selective ensemble modeling method based on random vector functional link network and game theory, in: 2019 6th International Conference on Systems and Informatics (ICSAI), IEEE, 2019, pp. 584–588.

[249] P. Musikawan, K. Sunat, Y. Kongsorot, P. Horata, S. Chiewchanwattana, Parallelized metaheuristic-ensemble of heterogeneous feedforward neural networks for regression problems, IEEE Access 7 (2019) 26909–26932.

[250] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions, Journal of big Data 8 (2021) 1–74.

[251] L. Zhang, P. N. Suganthan, Visual tracking with convolutional random vector functional link network, IEEE Transactions on Cybernetics 47 (2017) 3243–3253.

[252] P. A. Henríquez, G. A. Ruz, Twitter sentiment classification based on deep random vector functional link, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–6.

[253] R. Gao, L. Du, P. N. Suganthan, Q. Zhou, K. F. Yuen, Random vector functional link

neural network based ensemble deep learning for short-term load forecasting, Expert Systems with Applications 206 (2022) 117784.

[254] W. X. Cheng, P. N. Suganthan, R. Katuwal, Time series classification using diversified ensemble deep random vector functional link and resnet features, Applied Soft Computing (2021) 107826. doi:https://doi.org/10.1016/j.asoc.2021.107826.

[255] M. Hu, J. H. Chion, P. N. Suganthan, R. K. Katuwal, Ensemble deep random vector functional link neural network for regression, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2022) 1–12. doi:10.1109/TSMC.2022.3213628.

[256] L. Du, R. Gao, P. N. Suganthan, D. Z. Wang, Time series forecasting using online performance-based ensemble deep random vector functional link neural network, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–7.

[257] X. Yu, C. Lian, Y. Su, B. Xu, X. Wang, W. Yao, H. Tang, Selective ensemble deep bidirectional RVFLN for landslide displacement prediction, Natural Hazards (2022) 1–21.

[258] M. Hu, R. Gao, P. N. Suganthan, M. Tanveer, Automated layer-wise solution for ensemble deep randomized feed-forward neural network, Neurocomputing 514 (2022) 137–147.

[259] Q. Shi, P. N. Suganthan, J. Del Ser, Jointly optimized ensemble deep random vector functional link network for semi-supervised classification, Engineering Applications of Artificial Intelligence 115 (2022) 105214.

[260] M. A. Ganaie, M. Tanveer, Ensemble deep random vector functional link network using privileged information for Alzheimer's disease diagnosis, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2022).

[261] A. K. Malik, M. Tanveer, Graph embedded ensemble deep randomized network

for diagnosis of alzheimer's disease, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2022).

[262] Q. Dai, G. Zhang, Z. Fang, B. Xue, SAR target recognition with modified convolutional random vector functional link network, IEEE Geoscience and Remote Sensing Letters (2021).

[263] M. Hu, R. Gao, P. Suganthan, Deep reservoir computing based random vector functional link for non-sequential classification, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–8.

[264] S. Yang, R. Gao, L. Li, W. T. Ang, Deep randomized feed-forward networks based prediction of human joint angles using wearable inertial measurement unit: Performance comparison, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 01–08.

[265] M. Hu, P. N. Suganthan, Representation learning using deep random vector functional link networks for clustering, Pattern Recognition 129 (2022) 108744.

[266] K. Maeda, S. Takahashi, T. Ogawa, M. Haseyama, Convolutional sparse coding-based deep random vector functional link network for distress classification of road structures, Computer-Aided Civil and Infrastructure Engineering 34 (2019) 654–676.

[267] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems?, The Journal of Machine Learning Research 15 (2014) 3133–3181.

[268] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 972–981.

[269] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[270] J. Carrasco, S. García, M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm and Evolutionary Computation 54 (2020) 100665.

[271] A. Aggarwal, M. Tripathi, Short-term solar power forecasting using random vector functional link (RVFL) network, in: Ambient Communications and Computer Systems, Springer, 2018, pp. 29–39.

[272] I. Majumder, P. Dash, S. Dhar, Real-time energy management for PV–battery–wind based microgrid using on-line sequential kernel based robust random vector functional link network, Applied Soft Computing 101 (2021) 107059.

[273] E. L. Manibardo, I. Laña, J. Del Ser, Random vector functional link networks for road traffic forecasting: Performance comparison and stability analysis, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[274] Z. Cheng, J. Wang, A new combined model based on multi-objective salp swarm optimization for wind speed forecasting, Applied Soft Computing 92 (2020) 106294.

[275] X. Zhang, Y. Zou, S. Li, Enhancing incremental deep learning for FCCU end-point quality prediction, Information Sciences 530 (2020) 95–107.

[276] V. Kushwaha, N. M. Pindoriya, A SARIMA-RVFL hybrid model assisted by wavelet decomposition for very short-term solar PV power generation forecast, Renewable Energy 140 (2019) 124–139.

[277] I. Majumder, P. K. Dash, R. Bisoi, Short-term solar power prediction using multi-kernel-based random vector functional link with water cycle algorithm-based parameter optimization, Neural Computing and Applications 32 (2020) 8011–8029.

[278] T. Moudiki, F. Planchet, A. Cousin, Multiple time series forecasting using quasi-randomized functional link neural networks, Risks 6 (2018) 22.

[279] C. Lian, L. Zhu, Z. Zeng, Y. Su, W. Yao, H. Tang, Constructing prediction intervals for landslide displacement using bootstrapping random vector functional link networks selective ensemble with neural networks switched, Neurocomputing 291 (2018) 1–10. doi:https://doi.org/10.1016/j.neucom.2018.02.046.

[280] D. Gupta, N. Natarajan, M. Berlin, Short-term wind speed prediction using hybrid machine learning techniques, Environmental Science and Pollution Research (2021) 1–19.

[281] J. Li, L. Tang, S. Wang, Forecasting crude oil price with multilingual search engine data, Physica A: Statistical Mechanics and its Applications 551 (2020) 124178.

[282] X. Xue, W. Sun, J. Wang, Q. Li, G. Luo, K. Yu, RVFL-LQP: RVFL-based link quality prediction of wireless sensor networks in smart grid, IEEE Access 8 (2020) 7829–7841.

[283] Y. Zhang, B. Li, Y. Wang, Q. Tian, The forecast of the temperature in subway station based on RVFL neural network, in: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE, 2018, pp. 1–5.

[284] C. Lian, Z. Zeng, X. Wang, W. Yao, Y. Su, H. Tang, Landslide displacement interval prediction using lower upper bound estimation method with pre-trained random vector functional link network initialization, Neural Networks 130 (2020) 286–296.

[285] C. P. Chen, S. R. LeClair, Y.-H. Pao, An incremental adaptive implementation of functional-link processing for function approximation, time-series prediction, and system identification, Neurocomputing 18 (1998) 11–31.

[286] H. Qu, S. Fu, L. Pang, C. Ding, H. Zhang, Rapid temperature prediction method for electronic equipment cabin, Applied Thermal Engineering 138 (2018) 83–93.

[287] P. Zhou, Y. Jiang, C. Wen, T. Chai, Data modeling for quality prediction using improved orthogonal incremental random vector functional-link networks, Neurocomputing 365 (2019) 1–9.

[288] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., Psychological Review 65 (1958) 386.

[289] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Technical Report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[290] A. Elisseeff, H. Paugam-Moisy, Jnn, a randomized algorithm for training multilayer networks in polynomial time, Neurocomputing 29 (1999) 3–24.

[291] L. Wang, Support vector machines: theory and applications, volume 177, Springer Science & Business Media, 2005.

[292] C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE transactions on neural networks 13 (2002) 464–471.

[293] Y. Tian, Y. Shi, X. Liu, Recent advances on support vector machines research, Technological and economic development of Economy 18 (2012) 5–33.

[294] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing 408 (2020) 189–215.

[295] O. L. Mangasarian, E. W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE transactions on pattern analysis and machine intelligence 28 (2005) 69–74.

[296] M. Tanveer, M. Ganaie, A. Bhattacharjee, C. Lin, Intuitionistic fuzzy weighted least squares twin svms, IEEE Transactions on Cybernetics (2022).

[297] D. Chen, Y. Tian, X. Liu, Structural nonparallel support vector machine for pattern recognition, Pattern Recognition 60 (2016) 296–305.

[298] L. Liu, P. Li, M. Chu, Z. Zhai, L2-loss nonparallel bounded support vector machine for robust classification and its dcd-type solver, Applied Soft Computing 126 (2022) 109125.

[299] C.-N. Li, Y.-H. Shao, N.-Y. Deng, Robust l1-norm non-parallel proximal support vector machine, Optimization 65 (2016) 169–183.

[300] L. Liu, M. Chu, R. Gong, X. Qi, A nonparallel support vector machine with pinball loss for pattern classification, Journal of Intelligent & Fuzzy Systems 39 (2020) 911–923.

[301] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000.

[302] V. N. Vapnik, An overview of statistical learning theory, IEEE Transactions on Neural Networks 10 (1999) 988–999.

[303] M. Tanveer, M. A. Khan, S.-S. Ho, Robust energy-based least squares twin support vector machines, Applied Intelligence 45 (2016) 174–186.

[304] L. A. Zadeh, Fuzzy sets, Information and control 8 (1965) 338–353.

[305] H.-J. Zimmermann, Fuzzy set theory, Wiley interdisciplinary reviews: computational statistics 2 (2010) 317–332.

[306] D. Dubois, H. Prade, Fundamentals of fuzzy sets, volume 7, Springer Science & Business Media, 2012.

[307] C. Kahraman, Fuzzy applications in industrial engineering, volume 201, Springer, 2006.

[308] M. Yong, X. Zheng, Y. Zheng, S. Youxian, W. Zheng, Fault diagnosis based on fuzzy support vector machine with parameter tuning and feature selection, Chinese Journal of Chemical Engineering 15 (2007) 233–239.

[309] V. Khatibi, G. A. Montazer, Intuitionistic fuzzy set vs. fuzzy set application in medical pattern recognition, Artificial intelligence in medicine 47 (2009) 43–52.

[310] B. K. Sriwastava, S. Basu, U. Maulik, Predicting protein-protein interaction sites with a novel membership based fuzzy svm classifier, IEEE/ACM transactions on computational biology and bioinformatics 12 (2015) 1394–1404.

[311] E. Hüllermeier, Fuzzy sets in machine learning and data mining, Applied Soft Computing 11 (2011) 1493–1505.

[312] K. T. Atanassov, S. Stoeva, Intuitionistic fuzzy sets, Fuzzy sets and Systems 20 (1986) 87–96.

[313] Z. Xu, N. Zhao, Information fusion for intuitionistic fuzzy decision making: an overview, Information Fusion 28 (2016) 10–23.

[314] S. Laxmi, S. Gupta, S. Kumar, Intuitionistic fuzzy least square twin support vector machines for pattern classification, Annals of Operations Research (2022) 1–50.

[315] M. Ganaie, A. Kumari, A. K. Malik, M. Tanveer, Eeg signal classification using improved intuitionistic fuzzy twin support vector machines, Neural Computing and Applications (2022) 1–17.

[316] D. Hooda, R. Kumari, D. Sharma, Intuitionistic fuzzy soft set theory and its application in medical diagnosis, International Journal 7 (2018) 71.

[317] M. Ha, C. Wang, J. Chen, The support vector machine based on intuitionistic fuzzy number and kernel function, Soft Computing 17 (2013) 635–641.

[318] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence 35 (2013) 1798–1828.

[319] L. Yang, S. Song, S. Li, Y. Chen, G. Huang, Graph embedding-based dimension reduction with extreme learning machine, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (2019) 4262–4273.

[320] A. Iosifidis, A. Tefas, I. Pitas, Graph embedded extreme learning machine, IEEE transactions on cybernetics 46 (2015) 311–324.

[321] F. R. Chung, Spectral graph theory, volume 92, American Mathematical Soc., 1997.

[322] S. B. Kotsiantis, Decision trees: a recent overview, Artificial Intelligence Review 39 (2013) 261–283.

[323] I. D. Mienye, Y. Sun, A survey of ensemble learning: Concepts, algorithms, applications, and prospects, IEEE Access 10 (2022) 99129–99149.

[324] P. Pintelas, I. E. Livieris, Special issue on ensemble learning and applications, 2020.

[325] J. J. Rodriguez, L. I. Kuncheva, C. J. Alonso, Rotation forest: A new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1619–1630.

[326] G. E. Berrios, Alzheimer's disease: A conceptual history., International journal of geriatric psychiatry (1990).

[327] S. Gauthier, P. Rosa-Neto, J. Morais, C. Webster, World alzheimer report 2021: Journey through the diagnosis of dementia, Alzheimer's Disease International (2021).

[328] B. Richhariya, M. Tanveer, A. Rashid, A. D. N. Initiative, Diagnosis of alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE), Biomedical Signal Processing and Control 59 (2020) 101903.

[329] E. Pellegrini, L. Ballerini, M. d. C. V. Hernandez, F. M. Chappell, V. González-Castro, D. Anblagan, S. Danso, S. Muñoz-Maniega, D. Job, C. Pernet, Machine learning of neuroimaging for assisted diagnosis of cognitive impairment and dementia: a systematic review, Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring 10 (2018) 519–535.

[330] M. Orouskhani, S. Rostamian, F. S. Zadeh, M. Shafiei, Y. Orouskhani, Alzheimer's disease detection from structural mri using conditional deep triplet network, Neuroscience Informatics (2022) 100066.

[331] M. Amini, M. Pedram, A. Moradi, M. Ouchani, Diagnosis of Alzheimer's disease severity with fmri images using robust multitask feature extraction method and convolutional neural network (CNN), Computational and Mathematical Methods in Medicine 2021 (2021).

[332] S. Afzal, M. Maqsood, U. Khan, I. Mehmood, H. Nawaz, F. Aadil, O.-Y. Song, Y. Nam, Alzheimer disease detection techniques and methods: A review., International Journal of Interactive Multimedia & Artificial Intelligence 6 (2021).

[333] E. E. De Roeck, P. P. De Deyn, E. Dierckx, S. Engelborghs, Brief cognitive screening instruments for early detection of Alzheimer's disease: a systematic review, Alzheimer's research & therapy 11 (2019) 1–14.

[334] M. Tanveer, B. Richhariya, R. U. Khan, A. H. Rashid, P. Khanna, M. Prasad, C. Lin, Machine learning techniques for the diagnosis of alzheimer's disease: A review, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16 (2020) 1–35.

[335] L. I. Kuncheva, J. J. Rodriguez, Classifier ensembles with a random linear oracle, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 500–508.

[336] P. B. Nemenyi, Distribution-free multiple comparisons., Princeton University, 1963.

[337] Y. Zhang, J. Duchi, M. Wainwright, Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates, The Journal of Machine Learning Research 16 (2015) 3299–3340.

[338] M. A. Ganaie, M. Tanveer, A. D. N. Initiative, Fuzzy least squares projection twin support vector machines for class imbalance learning, Applied Soft Computing 113 (2021) 107933.

[339] C. Saunders, A. Gammerman, V. Vovk, Ridge regression learning algorithm in dual variables, ICML-1998 Proceedings of the 15th International Conference on Machine Learning (1999).

[340] B. B. Hazarika, D. Gupta, P. Borah, An intuitionistic fuzzy kernel ridge regression classifier for binary classification, Applied Soft Computing 112 (2021) 107816.

[341] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2011) 513–529.

[342] D. Dua, C. Graff, UCI machine learning repository, 2017. URL: http://archive.ics.uci.edu/ml.

[343] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework., Journal of Multiple-Valued Logic & Soft Computing 17 (2011).

[344] B. Chen, Y. Fan, W. Lan, J. Liu, C. Cao, Y. Gao, Fuzzy support vector machine with graph for classifying imbalanced datasets, Neurocomputing 514 (2022) 296–312.

[345] H. Yu, C. Sun, X. Yang, S. Zheng, H. Zou, Fuzzy support vector machine with relative density information for classifying imbalanced data, IEEE transactions on fuzzy systems 27 (2019) 2353–2367.

[346] P. Cunningham, S. J. Delany, k-nearest neighbour classifiers-a tutorial, ACM computing surveys (CSUR) 54 (2021) 1–25.

[347] A. Tharwat, T. Gaber, A. Ibrahim, A. E. Hassanien, Linear discriminant analysis: A detailed tutorial, AI communications 30 (2017) 169–190.

[348] Z. Liang, L. Zhang, Intuitionistic fuzzy twin support vector machines with the insensitive pinball loss, Applied Soft Computing 115 (2022) 108231.

[349] M. Tanveer, A. Sharma, P. N. Suganthan, General twin support vector machine with pinball loss function, Information Sciences 494 (2019) 311–327.

[350] L. Liu, M. Chu, R. Gong, L. Zhang, An improved nonparallel support vector machine, IEEE Transactions on Neural Networks and Learning Systems 32 (2020) 5129–5143.

[351] Y. Zhang, J. Duchi, M. Wainwright, Divide and conquer kernel ridge regression, in: Conference on Learning Theory, 2013, pp. 592–617.

[352] A. Frank, UCI machine learning repository, http://archive. ics. uci. edu/ml (2010).

[353] M. F. Amasyali, O. K. Ersoy, Classifier ensembles with the extended space forest, IEEE Transactions on Knowledge and Data Engineering 26 (2013) 549–562.

[354] T. G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.

[355] D. Musicant, NDC: normally distributed clustered datasets, Computer Sciences Department, University of Wisconsin, Madison (1998).

[356] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of The IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[357] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[358] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).

[359] T. Salimans, D. P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks, Advances in Neural Information Processing Systems 29 (2016) 901–909.

[360] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[361] L. Zhang, P. N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE Transactions on Cybernetics 45 (2014) 2165–2176.

[362] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, The Annals of Mathematical Statistics 11 (1940) 86–92.

[363] M. Sugiyama, Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis., Journal of Machine Learning Research 8 (2007).

[364] E. Westman, J.-S. Muehlboeck, A. Simmons, Combining mri and csf measures for classification of Alzheimer's disease and prediction of mild cognitive impairment conversion, Neuroimage 62 (2012) 229–238.

[365] M. Reuter, N. J. Schmansky, H. D. Rosas, B. Fischl, Within-subject template estimation for unbiased longitudinal image analysis, Neuroimage 61 (2012) 1402–1418.

[366] C. Lian, M. Liu, J. Zhang, D. Shen, Hierarchical fully convolutional network for joint atrophy localization and Alzheimer's disease diagnosis using structural mri, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2018) 880–893.

[367] R. Y. Lo, W. J. Jagust, Predicting missing biomarker data in a longitudinal study of Alzheimer disease, Neurology 78 (2012) 1376–1382.

[368] M. V. Narkhede, P. P. Bartakke, M. S. Sutaone, A review on weight initialization strategies for neural networks, Artificial Intelligence Review 55 (2022) 291–322.

[369] S. S. Sodhi, P. Chandra, S. Tanwar, A new weight initialization method for sigmoidal feedforward artificial neural networks, in: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 291–298.

[370] H. Yang, X. Ding, R. Chan, H. Hu, Y. Peng, T. Zeng, A new initialization method based on normed statistical spaces in deep networks, Inverse Problems & Imaging 15 (2021) 147.

[371] S. Li, Z. Zhao, T. Liu, R. Hu, X. Du, Initializing convolutional filters with semantic features for text classification, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1884–1889.

[372] D. Mishkin, J. Matas, All you need is a good init, arXiv preprint arXiv:1511.06422 (2015).

[373] D. Aguirre, O. Fuentes, Improving weight initialization of relu and output layers, in: International Conference on Artificial Neural Networks, Springer, 2019, pp. 170–184.

[374] P. Steiner, A. Jalalvand, P. Birkholz, Cluster-based input weight initialization for echo state networks, IEEE Transactions on Neural Networks and Learning Systems (2022).

[375] S. Koturwar, S. Merchant, Weight initialization of deep neural networks (dnns) using data statistics, arXiv preprint arXiv:1710.10570 (2017).

[376] A. Rahimi, B. Recht, Random features for large-scale kernel machines., in: NIPS, volume 3, Citeseer, 2007, p. 5.

[377] S. Mehrkanoon, J. A. Suykens, Deep hybrid neural-kernel networks using random fourier features, Neurocomputing 298 (2018) 46–54.

[378] P. V. de Campos Souza, Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature, Applied soft computing 92 (2020) 106275.

[379] Z.-H. Zhou, J. Feng, Deep forest, arXiv preprint arXiv:1702.08835 (2017).

[380] Z.-H. Zhou, J. Feng, Deep forest, National Science Review 6 (2019) 74–86.

[381] V. López, A. Fernández, J. G. Moreno-Torres, F. Herrera, Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics, Expert Systems with Applications 39 (2012) 6585–6608.

[382] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[383] D. Liang, G. Ruobin, P. N. Suganthan, D. Z. Wang, Bayesian optimization based dynamic ensemble for time series forecasting, Information Sciences (2022). doi:https://doi.org/10.1016/j.ins.2022.01.010.

[384] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Transactions on Evolutionary Computation 15 (2010) 4–31.

[385] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering 40 (2014) 16–28.