Advancing Randomized Neural Networks: Interpretability, Robustness, Expressivity and Scalability

Ph.D. Thesis

By
Md Sajid

under the supervision of
Prof. M. Tanveer



Department of Mathematics
INDIAN INSTITUTE OF TECHNOLOGY INDORE

July 2026

Advancing Randomized Neural Networks: Interpretability, Robustness, Expressivity and Scalability

A THESIS

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

By
Md Sajid

under the supervision of
Prof. M. Tanveer



Department of Mathematics
INDIAN INSTITUTE OF TECHNOLOGY INDORE
July 2026

INDIAN INSTITUTE OF TECHNOLOGY INDORE

I hereby certify that the work which is being presented in the thesis entitled **Advancing Randomized Neural Networks: Interpretability, Robustness, Expressivity and Scalability** 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 2021** to **July 2026** under the supervision of **Prof. M. Tanveer, 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.

Md Sajid 27/07/2026

Signature of the student with date

(Md Sajid)

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

27/07/2026

Signature of thesis supervisor with date

(Prof. M. Tanveer)

Md Sajid has successfully given his Ph.D. Oral Examination held on **July 27, 2026**.

27/07/2026

Signature of thesis supervisor with date

(Prof. M. Tanveer)

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my heartfelt gratitude to all those who have contributed to making this journey both enriching and memorable.

First and foremost, I am deeply grateful to the Almighty Allah Rabbul Alamin, without whose blessings and mercy none of this would have been possible.

I extend my sincerest thanks to my supervisor, Prof. M. Tanveer, whose constant guidance, vision, and inspiration proved invaluable at every stage of this research. It was his remarkable research profile that first drew me to IIT Indore, and I can say with great conviction that it remains one of the best decisions of my life. He has been the cornerstone of my academic growth, not merely as a supervisor, but as a mentor who shaped the way I think, work, and contribute to the scientific community. Coming from a pure mathematics background, I had no exposure to machine learning when I first joined the OPTIMAL Research Lab. It was Prof. Tanveer's belief in his students and his thoughtful, structured approach to mentorship that helped me transition into the field and eventually conduct research that is both meaningful and impactful. I must also acknowledge that my understanding of a PhD underwent a profound transformation under his mentorship. I once believed that a doctorate was simply about fulfilling a publication requirement, but Prof. Tanveer showed me that it is so much more. Through the opportunities he provided, mentoring interns, B.Tech., M.Sc., and Ph.D. students, organizing conferences, workshops, and summer schools: I learned to manage multiple responsibilities simultaneously, collaborate across disciplines and institutes, and remain attuned to the evolving demands and expectations of the global research community. His encouragement to explore the biomedical AI domain opened new avenues for my research, particularly in Alzheimer's disease diagnosis. This exposure offered valuable insights into real-world healthcare challenges and practical solutions, including translational technologies and patent contributions. In retrospect, I received far more than I had ever expected from this Ph.D.; and that is, in no small measure, because of my supervisor. The growth I see in myself today, both as a researcher and as a professional, would not have been possible without being a part of his lab and his vision.

I am deeply grateful to Dr. Hem C. Jha and Dr. Mohd. Arshad, my research progress

committee members, for their insightful feedback and constructive comments, which significantly enhanced the quality of my work. Their critical evaluation and thoughtful suggestions at various stages helped shape the direction of this research. I am particularly indebted to Dr. Arshad, who mentored me not only through my coursework but throughout my PhD journey, always being available for technical discussions and guidance whenever I needed it.

My heartfelt gratitude goes to my Ph.D. seniors Dr. Ashwani Kumar Malik (Assistant Professor at SRM University) and Dr. M.A. Ganaie (Assistant Professor at IIT Ropar) for their invaluable guidance during the initial phase of my research, which laid a strong foundation for my work. I would also like to express my sincere appreciation to my colleagues and amazing lab mates: Mushir, Abdul, Abdur, Ritik, and Dr. Anuradha Kumari (currently a postdoctoral researcher at KU Leuven), for their continuous support, encouragement, and the collaborative environment they helped create. I am equally thankful to my junior lab members, Kritika and Chhavi, for their cooperation and positive presence in the lab. I would also like to express my sincere gratitude to Dr. Rahul Sharma (Assistant Professor, LNMIIT Jaipur) for his efforts in helping me understand the nuances of biomedical data, particularly MRI for Alzheimer's disease research, which greatly enriched my knowledge and research perspective.

I would also like to sincerely thank my collaborators, especially Prof. P. N. Suganthan and Prof. Javier (Javi) Del Ser, for their invaluable collaboration, insightful discussions, constructive feedback, and continuous support, which significantly enriched the quality and impact of this research.

I would like to express my heartfelt gratitude to my close friend, Dr. Pinki Khatun (Ph.D., IIT Indore; currently a postdoctoral researcher at the University of Florence), for being a constant source of strength and support throughout my doctoral journey. From the very beginning, her presence brought a sense of clarity and reassurance during both the most challenging and transformative moments of my Ph.D. I am especially thankful to her for introducing me to LaTeX, which fundamentally changed how I approached academic writing and documentation. She cultivated in me a disciplined and structured approach to life, from daily routines to study habits, which became instrumental in my personal and

professional growth. Our countless discussions, particularly in linear algebra, where her expertise in numerical linear algebra beautifully complemented my background in machine learning, were not only intellectually enriching but also deeply inspiring. These exchanges broadened my mathematical perspective and enhanced my research in ways I had not anticipated. Beyond academics, she played a pivotal role in shaping my experience at IIT Indore in a way that made the journey more meaningful, memorable, and vibrant. Her presence brought a sense of balance, positivity, and warmth that extended far beyond the confines of research. Whether it was offering encouragement during difficult times, celebrating small victories, or simply being there when it mattered most, her influence has been profound and enduring. In ways that words can scarcely capture, she has contributed immensely to who I have become, both as a researcher and as a person. Her unwavering belief in me, even when I doubted myself, gave me the courage to persevere. For all of this and more, I remain deeply and eternally grateful.

I would like to express my heartfelt gratitude to my close friends, Abhishek, Lalita and Riju, for their unwavering friendship, constant encouragement, and support throughout my Ph.D. journey.

I would like to further thank my friends of life: Shadman, Sarwar Bhai, Shahfahad, Dr Sajjad Bhai, Shatrudhan, Shubham, Aman, Ritesh, Arif bhai, Atif bhai, Zahir bhai, Nissar bhai, Shahid bhai, and Muntazir bhai, who have been my pillars of strength throughout this journey. Your laughter in moments of joy, your reassurance during times of doubt, and your unwavering belief in me when I struggled to believe in myself have made all the difference. The late-night conversations, the shared meals, the countless memories we created together, these became the emotional sustenance that kept me going. You celebrated my smallest victories as if they were your own and stood by me through every setback without judgment. This thesis bears the imprint of your friendship, encouragement, and the sense of belonging you gave me. I am forever grateful for each one of you.

I am also thankful to my juniors from M.Sc. and B.Tech. at IIT Indore: Rahul, Akshat, Moksha, Anushka, Asif, Soham, Khush, and Sathwik: whose energy, curiosity, and fresh perspectives brought new life to our research discussions and made the lab environment more vibrant and engaging.

I extend my gratitude to the Head and DPGC of the Department of Mathematics, and to professors from the departments, as well as to the administrative and support staff of the Department of Mathematics, particularly Jitendra Sir and Murphy Sir, for their invaluable assistance and cooperation. My thanks also go to the institute's administrative staff for their support throughout my Ph.D. journey.

Most importantly, I would like to express my deepest gratitude to the teachers and mentors who shaped my academic path and made this moment possible. I am profoundly grateful to Mr. Sanjay Kumar Shandilya, my 10th-grade teacher and lifelong mentor, whose guidance laid the foundation for all that followed. My sincere thanks to Dr. Manoj Kumar Singh, Assistant Professor at St. Xavier's College, Ranchi, who mentored me throughout my B.Sc., inspired my love for mathematics, and guided me to crack IIT JAM, opening the doors to IIT Delhi for my Master's. I am deeply indebted to Dr. Amit Priyadarshi, my Master's thesis supervisor at IIT Delhi, for introducing me to research and shaping my analytical thinking. Special thanks to Prof. Ritumoni Sarma at IIT Delhi, one of my most inspiring professors, whose support during my Master's and Ph.D. applications was invaluable. Each of these remarkable individuals played a pivotal role in bringing me to this defining moment of my life.

Above all, I would like to express my deepest respect, love, and gratitude to my parents: my father, Md Sajjad, and my mother, Ruhi Afja. I come from a very small village where I am the first person to pursue a Ph.D., and all the credit for this achievement goes to my papa and ammi. There were many friends in my village who were as academically brilliant as I was, but they did not get the opportunity to continue their education. It was my parents who ensured I never felt any difficulty, whether in terms of care, emotional support, or financial resources. They allowed me to pursue higher studies away from home without ever letting me realize the hardships they faced to make it possible. Only later did I come to understand the sacrifices they made silently and selflessly. My father, with his quiet strength and unwavering determination, worked tirelessly to provide for our family and ensure that education remained our top priority. He believed in the power of knowledge to transform lives, and he instilled that belief in me from a very young age. My mother, with her boundless love, prayers, and endless patience, became my emotional anchor. Their

encouragement during moments of self-doubt, their pride in every small achievement, and their ability to find joy in my success despite their own struggles have been my greatest source of motivation. Together, they created an environment where dreams were nurtured, not limited by circumstances. Their unwavering belief in education and in me has been the foundation of everything I have accomplished. Every page of this thesis is a testament to their sacrifices, their hopes, and their unconditional love.

My sisters, Najia (Appi) and Sabiha (Nagma), are the pillars of love and support in my life. I still cherish those beautiful days when we all walked to school together, holding hands and dreaming of the future. Even after their marriages, the bond we share remains as pure and strong as it was in our childhood. Appi used to teach me with such patience and care, and we spent countless hours playing together. Nagma, my beloved babu, no matter how much she grows, will always remain our little one. Whenever we meet, it is with the same warmth, laughter, and energy that defined our childhood. These memories and moments make me miss home deeply and also motivate me to visit whenever I can.

My nephew, Arhaan, and my niece, Kashaf (Appi's children) wait eagerly for my visits home. Their excitement (perhaps second only to my parents') fills the house with joy and reminds me of the purest forms of love and happiness. They have brought immense light and laughter into our lives, and I am truly blessed to have them. I am also deeply grateful to my brothers-in-law, Raju (Appi's husband) and Mahmood (Nagma's husband), for their understanding, kindness, and support. I am fortunate to have gained not just family members, but true well-wishers and friends through them.

I would like to express my profound gratitude to the Director of the Indian Institute of Technology Indore for providing the research facilities and resources necessary for my work. I also acknowledge the financial support provided by the Council of Scientific and Industrial Research (CSIR), New Delhi, under Grant 09/1022(13847)/2022-EMR-I, which enabled me to pursue this research.

Finally, I am thankful to everyone who, directly or indirectly, contributed to this work and to my journey. Each person mentioned here, and many unnamed, have left an indelible mark on my life and work.

Dedicated

to

Papa (Md Sajjad) & Ammi (Ruhi Afja)

ABSTRACT

Can the predictive power of deep learning be achieved without its computational burden and training instability? This thesis addresses this question through the systematic advancement of Randomized Neural Networks (RdNNs), a theoretically grounded learning framework in which hidden-layer parameters are randomly assigned and fixed, reducing the learning problem to a closed-form solution for the output weights alone. This design eliminates iterative gradient-based optimization entirely, yielding fast training, guaranteed global optimality, and freedom from the hardware demands characteristic of conventional deep learning, while retaining universal approximation capability.

This thesis advances shallow, broad, deep, and ensemble RdNN architectures along four fundamental and interconnected directions. First, to address the inherent opacity of randomized architectures, interpretability is systematically embedded into the Random Vector Functional Link (RVFL) network and its ensemble deep variants through neuro-fuzzy integrations that encode human-readable IF-THEN fuzzy rules within the learning process, enabling transparent and auditable decision-making without sacrificing predictive performance. Second, robustness is advanced through novel formulations that incorporate intuitionistic fuzzy sample weighting, graph-embedded geometric representations, and the wave loss function, collectively mitigating the adverse effects of noise, outliers, and class imbalance while preserving the intrinsic topological structure of data across diverse and challenging real-world environments. Third, scalability is addressed by overcoming the matrix inversion bottleneck inherent in standard RVFL models through the integration of granular ball computing, which operates on compact granular representations of the training data rather than individual samples, and through ADAM optimization formulations that bypass matrix inversion entirely, enabling efficient learning on large-scale datasets. Fourth, expressivity is enhanced along two complementary directions. At the architecture level, complex-valued randomized networks are developed that introduce principled mechanisms for transforming real-valued tabular datasets into the complex domain, exploiting the richer representational capacity that complex-valued weights, activation functions, and feature interactions afford beyond what real-valued parameterizations can capture. At the feature learning level, the

expressive capacity of ensemble deep RVFL is substantially augmented through the integration of a fuzzy inference system within each hidden layer, wherein unsupervised fuzzification generates diverse and complementary fuzzified feature representations while supervised defuzzification produces semantically meaningful, human-interpretable outputs. This dual enhancement of fuzzified and defuzzified features across successive layers enriches the hidden representations available to each base model, capitalizing on inherent ensemble diversity and enabling the network to capture nuanced patterns and complex relationships that standard random projection alone cannot express.

Extensive theoretical analysis and empirical evaluation across diverse benchmark datasets, including UCI, KEEL, and large-scale NDC datasets, as well as high-stakes medical applications such as Alzheimer's disease diagnosis and breast cancer classification, demonstrate that the proposed models consistently and significantly outperform state-of-the-art baselines. The frameworks developed in this thesis establish a rigorous and principled path toward randomized neural networks that are simultaneously interpretable, robust, scalable, and expressive, bridging the gap between computationally efficient learning theory and practical deployability, particularly in data-limited and resource-constrained settings where conventional deep learning remains impractical.

Keywords: Randomized neural networks, Random vector functional link network, Broad learning system, deep and ensemble deep Randomized neural networks, Intuitionistic fuzzy theory, Granular ball computing, Graph embedding, Complex-valued neural networks, Fuzzy inference system.

List of Publications

A. Outcomes from Ph.D. Thesis Work

A1. In Refereed Journals

1. **M. Sajid**, M. Tanveer and P. N. Suganthan (2025). "Ensemble Deep Random Vector Functional Link Neural Network Based on Fuzzy Inference System," in ***IEEE Transactions on Fuzzy Systems***, v. 33, no. 1, pp. 479-490, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3411614 **[SCI Indexed with Impact Factor.: 10.2, Q1]**

2. M. A. Ganaie*, **M. Sajid***, A. K. Malik and M. Tanveer (2024). "Graph Embedded Intuitionistic Fuzzy Random Vector Functional Link Neural Network for Class Imbalance Learning," in ***IEEE Transactions on Neural Networks and Learning Systems***, v. 35, no. 9, pp. 11671–11680, **DOI:** https://doi.org/10.1109/TNNLS.2024.3353531 **[SCI Indexed with Impact Factor.: 9.7, Q1]** * M. Sajid and M. A. Ganaie contributed equally to this work.

3. **M. Sajid**, A. K. Malik, M. Tanveer and P. N. Suganthan (2024). "Neuro-Fuzzy Random Vector Functional Link Neural Network for Classification and Regression Problems," in ***IEEE Transactions on Fuzzy Systems***, v. 32, no. 5, pp. 2738 - 2749, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3359652 **[SCI Indexed with Impact Factor.: 10.2, Q1]**

4. **M. Sajid**, A. K. Malik and M. Tanveer (2024). "Intuitionistic Fuzzy Broad Learning System: Enhancing Robustness Against Noise and Outliers," in ***IEEE Transactions on Fuzzy Systems***, v. 32, no. 8, pp. 4460 - 4469, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3400898 **[SCI Indexed with Impact Factor.: 10.2, Q1]**

5. **M. Sajid**, A. Quadir and M. Tanveer (2025). "GB-RVFL: Fusion of Randomized Neural Network and Granular Ball Computing," in ***Pattern Recognition, Elsevier***, v. 159, p. 111142, **DOI:** https://doi.org/10.1016/j.patcog.2024.111142 **[SCI Indexed with Impact Factor.: 9.1, Q1]**

A2. In Refereed Conferences

6. **M. Sajid**, A. Quadir, M. Akhtar and M. Tanveer (2025). "RVFL-X: A Novel Randomized Network Based on Complex Transformed Real-Valued Tabular Datasets," in ***International Joint Conference on Neural Networks (IJCNN)***, pp. 1-8, **DOI:** https://doi.org/10.1109/IJCNN64981.2025.11227691 **[Core Rank B Conference]**

7. **M. Sajid**, A. Quadir and M. Tanveer (2024). "Wave-RVFL: A Randomized Neural Network Based on Wave Loss Function," in ***The 31st International Conference on Neural Information Processing (ICONIP)***, v. 15287, pp. 242-257, **DOI:** https://doi.org/10.1007/978-981-96-6579-2_17 **[Core Rank B Conference]**

A3. Under review

8. **M. Sajid**, A. Quadir, A. Rahaman and M. Tanveer (2026). "Robust Ensemble Deep Randomized Neural Networks".

B. Other Outcomes Outside of the Ph.D. Thesis Work

B1. In Refereed Journals

1. **M. Sajid**, R. Sharma, I. Beheshti and M. Tanveer (2024). "Decoding Cognitive Health Using Machine Learning: A Comprehensive Evaluation for Diagnosis of Significant Memory Concern," in ***WIREs Data Mining and Knowledge Discovery***, v. 14, no. 5, p. e1546, **DOI:** https://doi.org/10.1002/widm.1546 **[SCI Indexed with Impact Factor.: 15.0, Q1]**

2. A. Quadir, **M. Sajid**, and M. Tanveer (2025). "Granular Ball Twin Support Vector Machine," in ***IEEE Transactions on Neural Networks and Learning Systems***, v. 36, no. 7, pp. 12444 - 12453, **DOI:** https://doi.org/10.1109/TNNLS.2024.3476391 **[SCI Indexed with Impact Factor.: 9.7, Q1]**

3. M. Tanveer, M. Pathak, **M. Sajid**, A. Quadir, and Priyamvada (2026). "Fuzzy-Driven Broad Learning System with Class Probability and Density Awareness for Multi-view Data", in ***Neural Networks, Elsevier***, **DOI:** https://doi.org/10.1016/j.neunet.2026.108914 **[SCI Indexed with Impact Factor.: 7.2, Q1]**

4. M. Tanveer, A. Mishra, **M. Sajid**, and A. Quadir (2026). "BLS-CIL: Class Imbalance Broad Learning System Via Dual Weighting and Layer Trimming", in ***Pattern Recognition, Elsevier***, v. 174, p. 112940, **DOI:** https://doi.org/10.1016/j.patcog.2025.112940 **[SCI Indexed with Impact Factor.: 9.1, Q1]**

5. M. Tanveer, A. Mishra, A. Quadir, and **M. Sajid** (2026). "EDA-OCBLS: An Error-Distribution Aware One-Class Broad Learning System for Anomaly Detection," ***Neural Networks, Elsevier***, v. 203, p. 109177, **DOI:** https://doi.org/10.1016/j.neunet.2026.109177 **[I.F.: 7.2]**

6. M. Akhtar, A. Kumari, **M. Sajid**, A. Quadir, M. Arshad, P. N. Suganthan and M. Tanveer (2025). "Towards Robust and Inversion-Free Randomized Neural Networks: The XG-RVFL Framework," in ***Pattern Recognition, Elsevier***, v. 172,

p. 112711, **DOI:** https://doi.org/10.1016/j.patcog.2025.112711 [**SCI Indexed with Impact Factor.:** 9.1, Q1]

7. M. Tanveer, **M. Sajid**, M. Akhtar, A. Quadir, T. Goel, A. Aimen, S. Mitra, Y-D Zhang, CT Lin and J. Del Ser (2024). "Fuzzy Deep Learning for the Diagnosis of Alzheimer's Disease: Approaches and Challenges," in ***IEEE Transactions on Fuzzy Systems***, v. 32, no. 10, pp. 5477 - 5492, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3409412 [**SCI Indexed with Impact Factor.:** 10.2, Q1]

8. M. Tanveer, R. K. Sharma, A. Quadir and **M. Sajid** (2025). "Enhancing Robustness and Efficiency of Least Square Twin SVM via Granular Computing, in ***Pattern Recognition, Elsevier***, v. 170, p. 112021, **DOI:** https://doi.org/10.1016/j.patcog.2025.112021 [**SCI Indexed with Impact Factor.:** 9.1, Q1]

9. M. Tanveer, R. K. Sharma, **M. Sajid**, and A. Quadir (2025). "GRVFL-MV: Graph Random Vector Functional Link Based on Multi- View Learning, in ***Information Sciences, Elsevier***, v. 704, p. 121947, **DOI:** https://doi.org/10.1016/j.ins.2025.121947 [**SCI Indexed with Impact Factor.:** 6.0, Q1]

10. N. N. Tun, F. C. P. Abucci, F. Essid, M. Z. El-Deen, K. Vyas, **M. Sajid**, E. Chowdhury, and P. Li (2026). "Empowering Leaders: The Impact of Student Mentorship on Professional Growth, in ***IEEE Pulse***, v. 16, no. 5, pp. 67-73, **DOI:** https://doi.org/10.1109/MPULS.2025.3618469 [**SCI Indexed with Impact Factor.:** 1.2, Q4]

B2. In Refereed Conferences

11. **M. Sajid**, S. Maheshwari, A. Mishra, R. Mishra, and M. Tanveer (2026). "I^2-Med: Interpretable Medical Inference through Visual-Guided Dynamic Logits Calibration" in ***MICCAI'26*** [**Core Rank A Conference**].

12. R. Mishra, **M. Sajid***, V. Gupta*, and M. Tanveer (2026). "Neuro-Fuzzy Concept Learning for Interpretable Large Multimodal Models" in ***International Conference***

on Machine Learning (ICML) [**Core Rank A* Conference**].

*Equal contribution

13. **M. Sajid**, D. Gupta, Y. Modi, S. Jain, H. J. S. Ganji, A. Rahaman, H. Choudhary, N. Saleem, A. Hussain and M. Tanveer (2025). "AUREXA-SE: Audio-Visual Unified Representation Exchange Architecture with Cross-Attention and Squeezeformer for Speech Enhancement," in *Interspeech-W 2025*, pp. 56-61, **DOI:** https://doi.org/10.21437/AVSEC.2025-13 [**Core Rank A Conference**]

14. **M. Sajid**, M. Akhtar, M. Tanveer and S. Mitra (2025). "Fuzzy Learning at 60: Future of Trustworthy AI in Healthcare and LLM," in *IEEE International Conference on Fuzzy Systems (IEEE-FUZZ)*, pp. 1-5, **URL:** https://www.researchgate.net/publication/394478518 [**Core Rank B Conference**]

15. A. Rahaman, A. Quadir, **M. Sajid**, M. Akhtar, and M. Tanveer (2026). "ECA-BLS: An Efficient Complex-Augmented Broad Learning System," in *IEEE World Congress on Computational Intelligence (WCCI)* [**Core Rank A Conference**]

16. A. Quadir, **M. Sajid**, M. Akhtar, and M. Tanveer (2025). "Twin Restricted Kernel Machines for Multiview Classification," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, **DOI:** https://doi.org/10.1109/IJCNN64981.2025.11228247 [**Core Rank B Conference**]

17. A. Jain, J. S. Sanjotra, H. Choudhary, K. Agrawal, R. Shah, R. Jha, **M. Sajid**, A. Hussain, and M. Tanveer (2024). "LSTMSE-Net: Long Short Term Speech Enhancement Network for Audio-visual Speech Enhancement," in *Interspeech-W 2024*, pp. 33-37, **DOI:** https://doi.org/10.21437/AVSEC.2024-8 [**Core Rank A Conference**]

18. A. Quadir, **M. Sajid**, and M. Tanveer (2024). "One Class Restricted Kernel Machines," in *The 31st International Conference on Neural Information Processing (ICONIP)*, v. 2284, pp. 244-257,
DOI: https://doi.org/10.1007/978-981-96-6954-7_17 [**Core Rank B Conference**]

19. Babita D, K. S. Akash, **M. Sajid**, D. R. Nayak, and M. Tanveer (2024). "CaDT-Net: A Cascaded Deformable Transformer Network for Multiclass Breast Cancer Histopathological Image Classification," in ***The 31st International Conference on Neural Information Processing (ICONIP)***, v. 15292, pp. 361–372, **DOI:** https://doi.org/10.1007/978-981-96-6594-5_26 **[Core Rank B Conference]**

B3. Under Review

20. **M. Sajid**, S. Pandit, M. A. Hussain, K. K. Singh, N. S. Sathwik, K.-C. Wong, G. Yang, J. D. Ser, and M. Tanveer (2026). "BRAIN-X: Graph-Attentive Neural Cross-Factorization with X-Loss for EEG-Based Alzheimer's Diagnosis" [Under Review in **Nature Mental Health**].

21. **M. Sajid**, A. Rajhans, L. Dalmia, T. Banerjee, K.-C. Wong, and M. Tanveer (2026). "From Convolutions to Transformers: A Large-Scale Comprehensive Evaluation and Benchmarking for Alzheimer's Disease Diagnosis Using MRI" [Under Review in **WIREs DMKD**].

22. **M. Sajid**, K. K. Singh, and M. Tanveer (2026). "GA-DBT: Geometry-Aware Dual Brain Transformer for fMRI-based Diagnosis" [Under Submission].

23. **M. Sajid**, and M. Tanveer (2026). "Uncovering Subjective Cognitive Decline from MRI: A Complex-Valued Ensemble Deep Network Approach" [Under Review].

24. **M. Sajid**, A. Rahaman, and M. Tanveer (2026). "Towards Early Alzheimer's Detection: A Complex-Valued MRI Transformation and Complex Broad Learning System Approach" [Under Review in **MICCAI-W**].

25. M. Tanveer, A. S. Rana*, S. Jain*, A. Kumar*, A. A. Tiwari*, A. Rahaman*, A. Quadir*, and **M. Sajid*** (2026). "GraM-Diff: A Unified Graph–Mamba Diffusion Framework for EEG-Based Alzheimer's Disease Data Generation and Diagnosis *Equal contribution" [Under Rebuttal in **MICCAI-W**].

26. A. Quadir, **M. Sajid**, and M. Tanveer (2024). "Multiview Random Vector Functional Link Network For Predicting DNA-binding Proteins, **DOI:** https://doi.org/10.48550/arXiv.2409.02588".

27. A. Quadir, **M. Sajid**, and M. Tanveer (2024). "Enhanced Feature Based Granular Ball Twin Support Vector Machine, **DOI:** https://doi.org/10.48550/arXiv.2410.05786".

28. R. Mishra, **M. Sajid**, K. K. Singh, and M. Tanveer (2026). "Beyond Additivity: Copula Interaction Score for Interpreting Large Multimodal Models." [Under Review in NeurIPS'26].

29. R. Mishra, **M. Sajid**, and M. Tanveer (2026). "SpectRA: Parameter-Efficient High-Rank Singular Vector Adaptation for Large Language Models" [Under Review].

B4. Patent (Filed/ Published/ Granted)

30. M. Tanveer, **M. Sajid**, S. Pandit, A. Hussain, N.S. Sathwik, and K.K. Singh (2025). "Cloud-based Platform and Method for EEG-based Diagnostic Analysis" (Publication No.: 48/2025, Publication Date: 28 November 2025, Application No.: 202521075675) [Published].

31. M. Tanveer, K. Agrawal, J. Rohan, R. Shah, and **M. Sajid** (2025). "EEG-based System and Method for Multiclass and Multilabel Emotion Recognition" (Publication No.: IN202521132671 A1, Journal No: 10/2026, Publication Date: 06 March 2026, Application No.: 202521132671, Application Date: 27 December 2025) [Published].

32. M. Tanveer, **M. Sajid**, A. Hussain, N.S. Sathwik, and K.K. Singh (2026). "MRI Analysis and Diagnostic Decision Support System for Alzheimer's Disease" [Under Filed].

Contents

List of Figures xi

List of Tables xix

List of Notations xx

List of Abbreviations xxi

1 Introduction 1

1.1 Background . . . 3

1.1.1 Randomized Neural Networks and the Random Vector Functional Link Framework . . . 4

1.1.2 Variants and Extensions of RVFL . . . 5

1.1.3 Robustness, Loss Functions, and Scalability . . . 6

1.1.4 Complex-Valued Neural Representations . . . 7

1.1.5 Broad Learning Systems and Deep Randomized Networks . . . 8

1.2 Motivation and Research Gaps . . . 9

1.2.1 The Interpretability Gap in Randomized Neural Networks . . . 10

1.2.2 Scalability and the Promise of Granular Computing . . . 10

1.2.3 Loss Function Sensitivity and Optimization Instability . . . 11

1.2.4 Noise, Outliers, and Class Imbalance in Standard RdNNs . . . 12

1.2.5 Unexploited Representational Power of Complex-Valued Architectures . . . 13

1.2.6 Expressiveness and Robustness Deficiencies in Deep and Ensemble Randomized Networks . . . 14

1.2.7 Noise Susceptibility in Broad RdNNs 15
1.3 Research Challenges in the Broader Neural Network Landscape 16
1.4 Why Randomized Neural Networks? Practical Advantages over Deep Neural Networks 18
1.5 Objectives 19
1.6 Chapters and Contributions of the Thesis 20

2 Literature Review **27**
2.1 Revisit: Notations 28
2.2 Critical Review of Existing Approaches for Addressing Neural Network Challenges 29
2.2.1 Interpretability in Neural Networks 29
2.2.2 Robust Learning under Noise and Outliers 31
2.2.3 Gradient-based Explanations 33
2.2.4 Learning from Imbalanced Data 35
2.2.5 Complex-valued Neural Networks 37
2.3 Shallow Randomized Neural Network: RVFL [1] 39
2.3.1 Mathematical Formulation: Shallow RVFL 40
2.3.2 RVFL without Direct Link (RVFLwoDL): Extreme Learning Machine (ELM) 42
2.4 Deep and Ensemble Deep RdNNs 42
2.4.1 Mathematical Formulation of dRVFL [2] 44
2.4.2 Mathematical Formulation of edRVFL [2] 45
2.5 Broad Randomized Neural Network 47
2.5.1 Mathematical Formulation of BLS [3] 50
2.6 Fuzzy Theory 52
2.6.1 Fuzzy Membership Scheme 52
2.6.2 Intuitionistic Fuzzy Membership (IFM) Scheme 53
2.6.3 Kernel Technique Used for Finding Distance and Radius in Assigning Intuitionistic Fuzzy Score 54

2.7 TSK Fuzzy System 57
2.8 Granular Ball Computation [4] 58
2.9 Graph Embedding (GE) [5] 61
2.10 Performance Metrics 62
2.11 Statistical Tests for Comparing Multiple Models 64
2.11.1 Ranking Scheme 64
2.11.2 Friedman Test 64
2.11.3 Nemenyi Post Hoc Test 65
2.11.4 Win–Tie–Loss Sign Test 65
2.11.5 Wilcoxon Signed-Rank Test 66

3 Interpretable Randomized Neural Networks 69

3.1 Background 70
3.2 Proposed Method 75
3.2.1 NF-RVFL 75
3.2.2 Time Complexity of the Proposed NF-RVFL Models 81
3.3 Dataset, Experimental Setup and Hyperparameter Setting 82
3.3.1 Binary, Multiclass, and Regression Dataset Description 82
3.3.2 Experimental Setup and Hyperparameter Settings 83
3.4 Experiments and Results 85
3.4.1 Compared Models 86
3.4.2 For Binary Classification: Experimental Results and Statistical Analysis on UCI Datasets 86
3.4.3 For Multiclass Classification: Experimental Results and Statistical Analysis on UCI Datasets 89
3.4.4 Comparision of RVFL and BLS-based Neuro-Fuzzy Models 90
3.4.5 For Regression Task: Experimental Results and Statistical Analysis on UCI Datasets 92
3.4.6 Interpretability of the Proposed NF-RVFL Models 95

3.4.6.1 Dependence Plot (PDP)-based feature importance interpretability . . . 95
3.4.6.2 Feature importance-based interpretability considering weights assigned to each feature during prediction . . . 97
3.5 Ablation Study and Further Analysis . . . 103
3.5.1 Ablation Study . . . 103
3.5.2 Number of Fuzzy Rules (clusters) V/S Performance of NF-RVFL Models . . . 105
3.5.3 Role of an Activation Function in the Proposed NF-RVFL Models . 106
3.6 Summary . . . 112

4 Robust and Scalable Randomized Neural Networks based on Granularity and Loss Function 115

4.1 Background . . . 116
4.2 Proposed Work: Granular Ball-based RVFLs . . . 120
4.2.1 Granular Ball Random Vector Functional Link (GB-RVFL) Network 120
4.2.2 Graph Embedded Granular Ball Random Vector Functional Link (GE-GB-RVFL) Network . . . 124
4.2.3 Time and Space Complexity Analysis of the Proposed Models . . . 129
4.2.4 Numerical Experiments and Results . . . 131
4.2.4.1 Datasets . . . 132
4.2.4.2 Compared Models . . . 132
4.2.4.3 Experimental Setup and Hyperparameter Setting . . . 132
4.2.4.4 Results and Discussions on UCI and KEEL Datasets . . . 133
4.2.4.5 Results and Discussions on UCI and KEEL Datasets in Noisy Environment . . . 136
4.2.4.6 Results and Discussions on NDC Dataset . . . 140
4.2.4.7 Application on Biomedical Domain . . . 141
4.2.5 Interpretability of the Proposed Models Through Graph Embedding and Granular Computing . . . 144

4.2.6 Discussion . . . 149
4.2.6.1 Balance Between Number of GBs Generated and the Resulting Accuracy of the Proposed GB-RVFL and GE-GB-RVFL Models with Different Purities . . . 149
4.2.6.2 Effect of Granular Ball Parameters "pur" and "num" on the Performance of the Proposed GB-RVFL and GE-GB-RVFL Models . . . 150
4.2.6.3 Effect of Parameter "pur" and "$Actfun$" on the Performance of the Proposed GB-RVFL Model . . . 152
4.2.6.4 Effect of Parameters "$\mathcal{C}$" and "$Actfun$" on the performance of the proposed GB-RVFL and GE-GB-RVFL models . . . 152
4.2.6.5 Effect of Parameters "α" on the performance of the proposed GE-GB-RVFL model . . . 153
4.3 Proposed Work: Wave-RVFL . . . 155
4.3.1 Proposed Method . . . 156
4.3.1.1 Wave loss Function . . . 156
4.3.1.2 Optimization Problem: Wave-RVFL . . . 158
4.3.1.3 Soultion: Wave-RVFL . . . 159
4.3.2 Experiments, Results and Discussion . . . 160
4.3.2.1 Compared Models, Datasets and Experimental Setup . . . 161
4.3.2.2 Experimental Results on UCI Dataset . . . 162
4.3.2.3 Robustness Evaluation of the Proposed Wave-RVFL Model on Datasets with Outliers . . . 165
4.3.2.4 Robustness Evaluation of the Proposed Wave-RVFL Model on Datasets with Gaussian Noise . . . 166
4.3.2.5 Sensitivity Analyses . . . 167
4.4 Comparison of Wave Loss, Granular-ball Computing, and Graph Embedding 169
4.5 Summary . . . 172

5 Geometry and Class Imbalance Aware Robust Randomized Neural Networks **173**

5.1 Background . . . 174

5.2 The Proposed Graph Embedded Intuitionistic Fuzzy RVFL for Class Imbalance Learning (GE-IFRVFL-CIL) . . . 177

5.2.1 Proposed Method . . . 177

5.2.2 Class Imbalance Weighting Scheme [6] . . . 182

5.2.3 LDA and LFDA under the GE Framework [5] . . . 182

5.3 Computational Complexity of the Proposed GE-IFRVFL-CIL Models . . . 184

5.4 Experiments and Results . . . 184

5.4.1 Experimental Procedure and Hyperparameter Settings . . . 185

5.4.2 Evaluation on KEEL Benchmark Datasets . . . 185

5.4.3 Evaluation on KEEL Datasets with Gaussian Noise . . . 189

5.4.4 Evaluation on ADNI Dataset . . . 192

5.5 Summary . . . 196

6 RVFL-X: A Novel Complex Randomized Network and Methods for Complex Transformation of Real-Valued Tabular Datasets **197**

6.1 Background . . . 198

6.2 Proposed Work . . . 201

6.2.1 Notations . . . 201

6.2.2 Conversion of Real-valued Data to Complex-valued Data . . . 201

6.2.2.1 **Natural Method** . . . 202

6.2.2.2 **Autoencoder-Based Method** . . . 202

6.2.3 RVFL-X . . . 204

6.3 Discussion on the Proposed Complex-valued Representation . . . 206

6.3.1 Motivation for Complex-valued Representation of Real-valued Tabular Data . . . 207

6.3.2 Difference Between Complex-valued Representation and Increasing Model Width . . . 208

6.3.3 Comparison with Existing Complex-valued Transformation Methods 209
6.3.4 Rationale for Employing an Autoencoder to Generate the Imaginary Component . . . 210
6.4 Experiments, Results and Discussion . . . 211
6.4.1 Compared Models, Datasets and Experimental Setup . . . 211
6.4.2 Evaluation on C-1 Category Datasets . . . 213
6.4.3 Evaluation on C-2 Category Datasets . . . 214
6.4.4 Evaluation on C-3 Category Datasets . . . 215
6.4.5 Evaluation on C-4 Category Datasets . . . 217
6.4.6 Sensitivity Analysis . . . 218
6.4.7 Ablation Study . . . 219
6.5 Summary . . . 219

7 Expressive and Interpretable Ensemble Deep Randomized Neural Networks 221

7.1 Background . . . 222
7.2 The Proposed Ensemble Deep RVFL Based on a Fuzzy Inference System (edRVFL-FIS) . . . 224
7.2.1 Proposed Method: Overall Framework of edRVFL-FIS . . . 225
7.2.1.1 Fuzzification Process for the Proposed Model . . . 227
7.2.1.2 Defuzzification Process for the edRVFL-FIS . . . 229
7.2.1.3 Output Layers Parameters for Each Base Model of the Proposed Model . . . 231
7.2.1.4 Final Output: Test Condition . . . 232
7.2.1.5 Further Takeaways from the Defuzzification Matrix of the Proposed Model . . . 233
7.2.2 Time and Space Complexity of the Proposed edRVFL-FIS Models . . . 235
7.3 Experiments and Results . . . 236
7.3.1 Experimental Setup . . . 236
7.3.2 Datasets . . . 238

7.3.3 Compared Models 240
7.3.4 Experimental Results, Discussions and Statistical Analysis on Category-1 UCI Dataset 242
7.3.5 Experimental Results, Discussions and Statistical Analysis on Category-2 UCI Dataset 243
7.3.6 Experimental Results, Discussions, Statistical Analysis on NDC Dataset 245
7.4 Discussion, Sensitivity Analysis and Erosion Experiment 246
7.4.1 Discussions Based on the Results on Category-1 Datasets 246
7.4.2 Discussions Based on the Results on Category-2 Datasets 247
7.4.2.1 Number of Fuzzy Rules (Clusters) V/S Performance of edRVFL-FIS Models 247
7.4.3 Number of Hidden Layers V/S Performance of edRVFL-FIS Models 250
7.5 Evaluation of the proposed models on UCI datasets with Gaussian Noise: Erosion experiments 252
7.6 Summary 253

8 Robust Deep, Ensemble Deep and Broad Randomized Neural Networks 255

8.1 Background 257
8.2 Proposed Methods: Intuitionistic Fuzzy-Based Deep and Ensemble Deep RVFLs 261
8.2.1 Proposed IF-dRVFL Network 261
8.2.2 Proposed IF-edRVFL Network 264
8.2.3 Experimental Results 265
8.2.3.1 Experimental Setup 266
8.2.3.2 Models in Comparison 267
8.2.3.3 Experimental Results and Statistical Analysis on Real-World Datasets Against SOTA Non-Fuzzy RdNNs 268

8.2.3.4 Experimental Results and Statistical Analysis on Real-World Datasets Against SOTA Fuzzy Baselines 269
8.2.3.5 Robustness Evaluation of Proposed Models on Datasets with Gaussian Noise 270
8.2.3.6 Sensitivity Analyses 271
8.3 Proposed Method Fuzzy and Intuitionistic Fuzzy BLS 273
8.3.1 Proposed F-BLS and IF-BLS 273
8.3.2 Complexity Analysis of the Proposed F-BLS and IF-BLS 277
8.3.3 Experiments and Results Analysis 277
8.3.3.1 Setup for Experiments 280
8.3.3.2 Results and Statistical Analyses on UCI Dataset 280
8.3.3.3 Evaluation on UCI Datasets with Gaussian Noise 283
8.3.3.4 Evaluation on ADNI Dataset 285
8.3.3.5 Sensitivity Analyses of the Hyperparameters of the Proposed Models 288
8.4 Summary 294

9 Comprehensive Evaluation of the Proposed Methods 297
9.1 Introduction 298
9.2 Experimental Evaluation 299
9.2.1 Discussion on Binary UCI Datasets 299
9.2.2 Discussion for Multiclass Datasets 301
9.2.3 Discussion on KEEL Datasets 304
9.2.4 Discussion on Large-Scale NDC Datasets 305
9.2.5 Performance Evaluation on KEEL Imbalanced Datasets based on Multiple Metrics 308
9.3 Guidelines for Selecting an Appropriate Randomized Neural Network Framework 309
9.3.1 Case I: Applications Requiring Explainability 309
9.3.2 Case II: Large-Scale Learning Problems 310

9.3.3 Case III: Imbalanced Classification Problems 310
9.3.4 Case IV: Noisy and Outlier-Contaminated Data 311
9.3.5 Case V: Complex Nonlinear Data Distributions 311
9.3.6 Case VI: Rich Feature Representation 311
9.3.7 Summary of Model Selection 312
9.4 Summary 312

10 Conclusions and Future Work **315**

10.1 Conclusion 315
10.2 Future Research Directions 317

List of Figures

2.1 The architecture of the RVFL model. 40

2.2 The architecture of the dRVFL model. 44

2.3 The architecture of the edRVFL model. 46

2.4 The architecture of the BLS model. 50

2.5 Granular ball generation process. 59

2.6 The visualization of generating granular balls by splitting the "fourclass" dataset. It assigns the label "$+1$" to the green granular balls, while the label "-1" is used for the magenta granular balls. 60

3.1 The architecture of the Neuro-fuzzy RVFL model. 77

3.2 Partial Dependence Plot (PDP) of each feature of the dataset pittsburg_bridges_T_OR_D *w.r.t.* the proposed NF-RVFL models, RVFL and Decision Tree. F-i in the X-axis denotes the ith feature of the dataset. 96

3.3 (a) and (b) showcase the accuracy of the proposed NF-RVFL models on binary and multiclass datasets, respectively, whereas (c) showcases the $RMSE$ of the proposed NF-RVFL models on regression datasets with respect to activation functions. 107

3.4 (a) and (b) showcase the accuracy of the proposed NF-RVFL models on binary and multiclass datasets, respectively, whereas (c) showcases the $RMSE$ of the proposed NF-RVFL models on regression datasets with respect to activation functions. 108

3.5 (a) and (b) showcase the accuracy of the proposed NF-RVFL models on binary and multiclass datasets, respectively, whereas (c) showcases the $RMSE$ of the proposed NF-RVFL models on regression datasets with respect to activation functions. 109

4.1 Effect of different labels of noise on the performance of the proposed GB-RVFL and GE-GB-RVFL models. 138

4.2 Effect of different labels of noise on the performance of the proposed GB-RVFL and GE-GB-RVFL model with the baseline IF-RVFL and NF-RVFL models. 138

4.3 Performance comparison of the proposed GB-RVFL and GE-GB-RVFL models against baseline models on ADNI data using sensitivity, specificity, and precision. 141

4.4 Visualization of the feature distance matrices $\mathcal{DE}_i$ through Heat map on "acute_inflammation" dataset. 147

4.5 Visualization of the feature distance matrices $\mathcal{DE}_i$ through Heat map on "conn_bench_sonar_mines_rocks" dataset. 148

4.6 Effect of granular ball parameters pur and num on the performance of the proposed GB-RVFL and GE-GB-RVFL models. 151

4.7 Effect of parameters purity (pur) and activation function $(Act\ fun)$ on the performance of the proposed GB-RVFL model. 153

4.8 Effect of parameters $\mathcal{C}$ and activation function $(Act\ fun)$ on the performance of the proposed GB-RVFL and GE-GB-RVFL models. 154

4.9 Effect of parameters α on the performance of the proposed GE-GB-RVFL model. 155

4.10 A graphical representation of the Wave loss function is provided, with η set to 1, and varying values of γ. 157

4.11 Effect of hyperparameters on the proposed Wave-RVFL model. 167

5.1 Performance comparison of classification models on ADNI data using F-Measure analysis. 193

5.2 Performance comparison of classification models on ADNI data using G-Mean analysis. 194

6.1 Architecture of the proposed RVFL-X network. 204

6.2 Sensitivity analyses of the proposed RVFL-X-N and RVFL-X-Auto models w.r.t. α. 216

7.1 The architecture of the edRVFL-FIS model. 225

7.2 (a), (b), and (c) showcase the accuracy of the proposed edRVFL-FIS models on NDC-10K, NDC-100K, and NDC-1M datasets, respectively, *w.r.t.* the number of fuzzy rules. 248

7.3 (a), (b), and (c) showcase the accuracy of the proposed edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C, respectively *w.r.t.* hidden layers. . . . 249

8.1 Flowchart of the proposed models. 265

8.2 Robustness: Effect of different noise levels on the performance of dRVFL, edRVFL, IF-dRVFL (proposed) and IF-edRVFL (proposed) models. 270

8.3 Effect of parameter σ on the performance of the proposed IF-dRVFL and IF-edRVFL models. 271

8.4 Effect of parameter $\mathbf{L}$ on the performance of the proposed IF-dRVFL and IF-edRVFL models. 272

8.5 The flowchart of finding the output weight matrix W_T of the proposed F-BLS and IF-BLS models. 277

8.6 Accuracy along with the standard deviation (shown in thin lines on the top) with respect to the regularization parameter $\mathcal{C}$ on the ADNI dataset. 286

8.7 Evaluation of the proposed IF-BLS model's testing accuracy on the CN_VS_AD case for Alzheimer's disease diagnosis, considering different numbers of feature groups (m), while varying the Kernel parameter (μ) and regularization parameter ($\mathcal{C}$). 289

8.8 Evaluation of the proposed IF-BLS model's testing accuracy on the CN_VS_MCI case for Alzheimer's disease diagnosis, considering different numbers of feature groups (m), while varying the Kernel parameter (μ) and regularization parameter ($\mathcal{C}$). 290
8.9 Evaluation of the proposed IF-BLS model's testing accuracy on the MCI_VS_AD case for Alzheimer's disease diagnosis, considering different numbers of feature groups (m), while varying the Kernel parameter (μ) and regularization parameter ($\mathcal{C}$). 291
8.10 Evaluation of the proposed F-BLS and IF-BLS models' testing accuracy while varying the number of feature nodes (p). 292

List of Tables

3.1 Details of the datasets used in experiments for binary and multiclass classification sourced from the UCI [7] repository. 84

3.2 Classification accuracies of the proposed NF-RVFL-R, NF-RVFL-K and NF-RVFL-C models along with RVFL, ELM, IFRVFL, GEELM-LDA, GEELM-LFDA, Total-Var-RVFL, and MCVELM on binary UCI datasets. . 87

3.3 Statistical analysis of the proposed models (listed in the first column) with all the models (listed in the first row) for the binary UCI datasets based on the Nemenyi post hoc test. 87

3.4 Classification accuracies of the proposed NF-RVFL-R, NF-RVFL-K and NF-RVFL-C models along with the compared baseline models, *i.e.,* RVFL, ELM, IFRVFL, Total-Var-RVFL, MCVELM on multiclass UCI datasets. . . 88

3.5 Statistical analysis of the proposed (listed in the first column) with all the models (listed in the first row) for multiclass UCI datasets based on the Nemenyi post hoc test. 89

3.6 Comparision of Neuro-Fuzzy based RVFL (proposed) and BLS models in terms of accuracy, standard deviation, rank, number of fuzzy rules for binary and multiclass classification. 91

3.7 The *RMSE* values and the corresponding best hyperparameters of the proposed NF-RVFL-R, NF-RVFL-K and NF-RVFL-C models along with the compared baseline models, *i.e.,* RVFL, ELM on regression UCI datasets. . . 94

3.8 Comparison of win-tie-loss results on UCI regression datasets. 95

3.9 Description of features in the Breast Cancer dataset. 98

3.10 Fuzzy vector generation through fuzzification. 99

3.11 The norm of the weights (from β) linking Breast Cancer features to the output layer for the final prediction in the proposed NF-RVFL models. The left column indicates the features, while the corresponding row entries for each feature denote the weights assigned by each model. 102

3.12 Abbreviation used for the NF-RVFL models with and without key components for the ablation analysis. ✓and ✗ denote the proposed model (row) with and without key component (column), respectively. 104

3.13 The details of the datasets used for the ablation study. 104

3.14 Performance (in terms of accuracy) of the proposed NF-RVFL models with and without key components for ablation analysis. The top three models are boldfaced *w.r.t.* each dataset. 104

3.15 Count of activation functions for proposed NF-RVFL models. 111

4.1 Index of activation functions. 133

4.2 Accuracy and rank of the proposed GB-RVFL and GE-GB-RVFL models against the baseline models on real-world datasets, i.e., KEEL and UCI. . . 134

4.3 Comparison of the proposed GE-GB-RVFL and other compared models using Nemenyi post hoc test. 135

4.4 Classification accuracies of the proposed GB-RVFL and GE-GB-RVFL models against the baseline models on UCI and KEEL datasets with contaminated label noise. 137

4.5 Accuracies of the proposed models versus the baseline models on NDC datasets. 140

4.6 Accuracies of the proposed GB-RVFL and GE-GB-RVFL models against the baseline models on Biomedical datasets. 142

4.7 Feature interpretability results for the datasets conn_bench_sonar_mines_rocks and acute_inflammation, based on the Frobenius norm distance of the feature distance matrices. 145

4.8 The number of GBs and the corresponding Accuracies by the proposed GB-RVFL model under different purities. 150

4.9 The number of granular balls and the corresponding Accuracies by the proposed GE-GB-RVFL model under different purities. 150
4.10 Testing accuracy, standard deviation and average rank of all the baseline models along with the proposed Wave-RVFL model. 163
4.11 Win Tie Loss test for the results tabulated in Table 4.10. 163
4.12 Performance of the proposed Wave-RVFL along with the RVFL with varying levels of outliers and noise. 165
4.13 Comparison between Wave Loss, Granular-ball Computing, and Graph Embedding. 170

5.1 Statistical description of the KEEL datasets. 186
5.2 The classification accuracies of the proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 models along with the existing models, *i.e.,* IFTWSVM, IFKRR, ELM, RVFL, IFRVFL and GE-IFWRVFL on KEEL [8] imbalanced benchmark datasets. 187
5.3 Performance of classification models on different levels of noise. 188
5.4 Performance of proposed as well as existing models on ADNI Data. 189
5.5 Statistical description of the KEEL datasets. 190
5.6 The description of the datasets used for Gaussian noise experiments. 190
5.7 Classification performance of the proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 along with the baseline models w.r.t. different levels of Gaussian Noise. 191

6.1 Accuracy and rank across 21 C-1 category datasets. 212
6.2 Accuracy, rank and standard deviation (Std. Dev.) across 14 C-2 category datasets. 212
6.3 Average accuracy and rank across 23 C-3 category datasets. 215
6.4 Average accuracy, rank and standard deviation (Std. Dev.) across 22 C-4 category datasets. 216
6.5 Ablation Study of the RVFL-X-N Model 218
6.6 Ablation Study of the RVFL-X-Auto Model 218

7.1 Details of the Category-1 datasets used in experiments sourced from the UCI [7] repository. 239

7.2 Details of the Category-2 datasets used in experiments sourced from the UCI [7] repository. 239

7.3 Classification accuracies of the proposed edRVFL-FIS models and the baseline models on Category-1 UCI datasets. 241

7.4 Wilcoxon-signed-rank test of the proposed edRVFL-FIS models *w.r.t.* baseline models on Category-1 UCI datasets. † represents the proposed models. 241

7.5 Classification accuracies of the proposed edRVFL-FIS and the baseline models on Category-2 UCI datasets. 241

7.6 Wilcoxon-signed-rank test of the proposed edRVFL-FIS models *w.r.t.* baseline models on Category-2 UCI datasets. † represents the proposed models. 242

7.7 Classification accuracies of the proposed edRVFL-FIS models and the baseline models on NDC datasets. 245

7.8 Performance of the proposed edRVFL-FIS models over NDC datasets with varying Gaussian noise 251

8.1 Average accuracy and standard deviation with SOTA non-fuzzy RdNNs on 13 KEEL datasets. 265

8.2 Average accuracy, standard deviation and rank comparison with SOTA fuzzy baselines on 12 UCI datasets. 269

8.3 Hyperparameters' description and range for the baseline and proposed models. 278

8.4 The testing accuracy of the proposed F-BLS and IF-BLS models along with the baseline models on UCI datasets. 279

8.5 Wilcoxon signed-rank test of the proposed F-BLS and IF-BLS models *w.r.t.* baseline models on UCI datasets. 279

8.6 Pairwise win-tie-loss of proposed and existing models on UCI datasets. . . . 279

8.7 The classification accuracies of the proposed F-BLS and IF-BLS models along with the existing models, *i.e.,* BLS, ELM, NeuroFBLS, IF-TSVM, and H-ELM on UCI dataset with varying levels of 5%, 10%, 15%, and 20% Gaussian noise. 283

8.8 Description of UCI datasets chosen for experiments with added Gaussian noise. 284

8.9 The testing accuracy of the proposed F-BLS and IF-BLS models and the baseline models on ADNI datasets. 287

9.1 Results on 27 binary UCI datasets. 300

9.2 Results on 30 multiclass UCI datasets. 302

9.3 Results on 20 KEEL datasets. 303

9.4 Accuracies of the proposed GB-RVFL models on 10 large NDC datasets. . . 305

9.5 AUC, Sensitivity, Specificity, Precision, F-Measure, and G-Mean for the GE-IFRVFL-CIL models on 56 KEEL datasets. 307

9.6 Guidelines for selecting the proposed randomized neural network frameworks according to application characteristics. 312

List of Notations

$\mathcal{C}$	Regularization parameter
$u_r \in \mathbb{R}^{1\times L}$	r^{th} input sample
$t_r \in \mathbb{R}^{1\times C}$	Target vector corresponding to the input sample u_r
$U \in \mathbb{R}^{M\times L}$	Input matrix containing all training samples
$T \in \mathbb{R}^{M\times C}$	Target matrix containing all output vectors
L	Number of input features
C	Number of output classes
M	Total number of training samples
$\psi(\cdot)$	Activation function
GB_j	j^{th} granular ball
$o_j \in \mathbb{R}^{1\times L}$	Center of the granular ball GB_j
$\otimes$	Kronecker product operator
$\oplus$	Concatenation operator
$\mathcal{S} \in \mathbb{R}^{M\times M}$	Fuzzy or intuitionistic fuzzy score matrix
$U_{uni} \in \mathbb{R}^{M\times M}$	Graph embedding matrix
$\mathbf{L}$	Total number of hidden layers in (e)dRVFL
l	l^{th} hidden layer index of the (e)dRVFL, where $l = 1, 2, \ldots, \mathbf{L}$
$(\cdot)^t, (\cdot)^T$	Transpose operator
$\varphi(\cdot)$	Feature mapping function used in Kernel methods
$K(\cdot, \cdot)$	Kernel function

List of Abbreviations

IF	Intuitionistic Fuzzy
GE	Graph Embedding (Embedded)
ERM	Empirical Risk Minimization
ANNs	Artificial Neural Networks
RdNNs	Randomized Neural Networks
RVFL	Random Vector Functional Link
ELM	Extreme Learning Machine
RVFLwoDL	Random Vector Functional Link without Direct Link
IFRVFL	Intuitionistic Fuzzy RVFL
BLS	Broad Learning System
CI	Class Imbalance
CIL	Class Imbalance Learning
GE-IFRVFL-CIL	Graph-Embedded Intuitionistic Fuzzy RVFL for CIL
NF-RVFL	Neuro-Fuzzy RVFL
F-BLS	Fuzzy Broad Learning System
IF-BLS	Intuitionistic Fuzzy Broad Learning System
TSK	Takagi–Sugeno–Kang
GD	Gradient Descent
DNNs	Deep Neural Networks
GB-RVFL	Granular Ball RVFL
GE-GB-RVFL	Graph-Embedded Granular Ball RVFL
Wave-RVFL	RVFL based on Wave Loss Function
RVFL-X	Complex-Transformed RVFL
edRVFL-FIS	edRVFL based on Fuzzy Inference System
dRVFL	Deep RVFL
edRVFL	Ensemble dRVFL
IF-dRVFL	Intuitionistic Fuzzy dRVFL
IF-edRVFL	Intuitionistic Fuzzy edRVFL

Chapter 1

Introduction

The pursuit of intelligent systems has long been driven by a fundamental question: can we design models that learn efficiently while remaining transparent, reliable, and scalable? Contemporary deep learning models have demonstrated extraordinary predictive capabilities, yet they often sacrifice interpretability and robustness for performance. This trade-off poses significant challenges in domains where trust and reliability are essential.

To formally study these challenges, we adopt the classical framework of statistical learning theory, which provides a principled foundation for understanding learning from data. Let $\mathcal{U} \subseteq \mathbb{R}^L$ denote the input space and $\mathcal{T}$ denote the target space. Assume that samples (u, t) are drawn independently from an unknown but fixed probability distribution $P(U, T)$ defined over $\mathcal{U} \times \mathcal{T}$ for $U \in \mathcal{U}$ and $T \in \mathcal{U}$. Given a training dataset $\mathcal{D} = (u_i, t_i)_{i=1}^M$, where each pair (u_i, t_i) is independently sampled from $P(U, T)$, the objective of supervised learning is to construct a function $f : \mathcal{U} \to \mathcal{T}$ that can accurately predict the target corresponding to unseen inputs.

In statistical learning theory, the performance of a prediction function f is measured through the expected risk and defined as:

$$R(f) = \mathbb{E}_{(u,t)\sim P}[L(t, f(u))],$$

where $L(\cdot, \cdot)$ denotes a loss function that quantifies the discrepancy between the predicted value and the true target. Since the underlying distribution $P(U, T)$ is unknown in practice, the expected risk cannot be computed directly. Instead, learning algorithms rely on the empirical risk defined over the training dataset and defined as:

$$R_{\text{emp}}(f) = \frac{1}{M} \sum_{i=1}^{M} L(t_i, f(u_i)).$$

The principle of empirical risk minimization (ERM) seeks to determine a function f from a hypothesis space $\mathcal{H}$ that minimizes this empirical risk.

The generalization ability of a learning algorithm depends on the relationship between the empirical risk and the expected risk. Classical results from statistical learning theory establish that the difference between these two quantities can be bounded in terms of complexity measures of the hypothesis space, such as the Vapnik–Chervonenkis (VC) dimension or Rademacher complexity. Consequently, designing learning models that balance expressive power with controlled model complexity is essential for achieving reliable generalization to unseen data.

In practice, highly expressive hypothesis spaces are often required to capture complex nonlinear relationships present in real-world data. Deep learning architectures represent one prominent class of such models and have demonstrated remarkable success in many learning tasks. However, training deep models typically involves significant computational resources, extensive hyperparameter tuning, and iterative optimization procedures that may suffer from slow convergence and sensitivity to initialization.

This observation motivates the search for alternative learning paradigms that can retain strong predictive capability while alleviating the computational burden and training instability associated with deep architectures. In this context, an important research question emerges: *can we achieve the predictive power of deep learning models without incurring*

their substantial computational cost and training instability? This thesis investigates learning frameworks that aim to address this challenge by combining expressive modeling mechanisms with efficient training procedures. In particular, the proposed approaches explore the use of randomized representations, structured feature transformations, and uncertainty-aware mechanisms to construct predictive models that are interpretable, efficient, scalable, robust, expressive and capable of providing reliable generalization performance.

1.1 Background

The structure and operations of biological neurons in the brain serve as the foundation for artificial neural networks (ANNs), a powerful class of machine learning models composed of interconnected nodes that process and transfer information through mathematical operations [9]. By discovering patterns and correlations in data, ANNs have demonstrated remarkable success across a wide range of disciplines, including rainfall forecasting [10], clinical medicine [11], stock market prediction [12], solving differential equations [13], and brain age estimation [14]. Deep learning, one of the most prominent branches of ANNs, employs multi-layer architectures to automatically extract high-level features from raw data, eliminating the need for manual feature engineering and enabling state-of-the-art performance in tasks such as speech recognition [15], natural language processing [16], and feature interpretation [17].

Despite their impressive capabilities, deep learning architectures suffer from several fundamental challenges. Their training relies on the gradient descent (GD) method, an iterative optimization procedure that backpropagates the difference between the predicted and actual outputs to update model weights and biases. This iterative strategy is time-consuming, prone to convergence at local rather than global optima [18], and highly sensitive to the choice of learning rate and initialization of parameters. Furthermore, deep learning models

typically consist of millions of parameters distributed across numerous layers, demanding substantial computational resources, such as GPUs or TPUs, and large volumes of labeled training data. These constraints severely limit the applicability of deep learning models beyond resource-intensive tasks such as speech and image recognition, thereby creating space for alternative neural architectures.

1.1.1 Randomized Neural Networks and the Random Vector Functional Link Framework

Randomized neural networks (RdNNs) [19] were developed precisely to circumvent the pitfalls of GD-based learning. In RdNNs, a subset of network parameters, specifically the hidden-layer weights and biases, are randomly generated and kept fixed throughout the training phase [20]. Only the output layer parameters are determined analytically, using the least-squares method or pseudo-inverse, which yields a closed-form solution and eliminates the need for iterative optimization entirely. This design philosophy results in fast training speed, universal approximation capability [21], and significantly reduced computational complexity compared to conventionally trained deep networks.

Among the most prominent RdNNs, the Random Vector Functional Link (RVFL) network [1] and the Extreme Learning Machine (ELM, also known as RVFL without direct links, or RVFLwoDL) [22] have attracted considerable research attention. The defining characteristic that distinguishes RVFL from ELM is the establishment of direct connections between the input layer and the output layer. This direct link serve as an intrinsic regularization mechanism, substantially improving generalization performance [23] while simultaneously reducing model complexity in alignment with Occam's parsimony principle and probably approximately correct (PAC) learning theory [2, 24]. As a result, RVFL achieves competitive classification and regression performance with a fraction of the computational

overhead of deep learning models.

1.1.2 Variants and Extensions of RVFL

The success of the conventional RVFL has motivated a rich body of research aimed at improving its generalization, stability, and applicability [25, 26]. A key instability in the standard RVFL arises from its transformation of original features into randomized features; to address this, Zhang et al. [27] proposed the sparse autoencoder RVFL (SP-RVFL), which employs ℓ_1-norm regularization to learn more stable network parameters. To exploit privileged information during training, Zhang and Yang [28] introduced RVFL+ and its kernel-based extension KRVFL+, the latter additionally handling nonlinear interactions in high-dimensional spaces. The challenge of selecting an optimal number of hidden nodes, a critical factor in RVFL performance [29], was addressed by Dai et al. [30] through the Incremental RVFL+ (IRVFL+), which constructively enlarges the hidden layer, and by Chakravorti and Satyanarayana [31] through the kernel-based exponentially expanded RVFL (KERVFL), which incorporates a kernel function to bypass hidden node selection entirely.

Recognizing the standard RVFL's susceptibility to noise and outliers, a consequence of its uniform sample weighting scheme, Malik et al. [32] proposed the Intuitionistic Fuzzy RVFL (IFRVFL), which assigns an intuitionistic fuzzy (IF) score to each sample using membership and non-membership functions [33, 34]. IFRVFL demonstrated improved generalization over standard RVFL on noisy datasets. Building on this, Malik et al. [35] proposed the Graph Embedded Intuitionistic Fuzzy Weighted RVFL (GE-IFWRVFL), which additionally preserves the geometric structure of data through a graph embedding (GE) framework [36]. However, both IFRVFL and GE-IFWRVFL fail to address the class imbalance problem inherent in many real-world datasets. Class imbalance, wherein the number of

samples in one class substantially exceeds those in others, causes classifiers that assign uniform importance to all samples to become biased toward the majority class. Efforts such as the Improved Fuzziness-based RVFL (IF-RVFL) [37] and Class-Specific Weighted RVFL (CSWRVFL) [38] have targeted this issue in isolation, but no RVFL model had simultaneously and holistically addressed noise, outliers, geometric structure preservation, and class imbalance within a single unified framework, until the proposal of GE-IFRVFL-CIL [39]. Furthermore, efficient angle-based twin RVFL (ATRVFL) [40] has been proposed for binary classification. Several recent extensions of RVFL and their developments are discussed in [25].

1.1.3 Robustness, Loss Functions, and Scalability

A less-explored but equally significant limitation of the standard RVFL concerns its reliance on the square loss function, which implicitly assumes that prediction errors follow a Gaussian distribution [41]. In practice, this assumption is routinely violated in the presence of noise and outliers [42]. Samples with large deviations exert disproportionate influence on the learned parameters, over-penalizing large errors, inducing steep gradients, and compromising numerical stability.

Scalability represents another fundamental challenge. The closed-form solution of RVFL requires the inversion of the entire training data matrix, an operation whose computational cost grows cubically with dataset size and becomes intractable for large-scale problems. Models such as IFRVFL and GE-IFRVFL-CIL further amplify this cost by requiring membership and non-membership computations in kernel spaces [32, 39]. Inspired by the cognitive principle that human perception processes information coarse-to-fine [43], granular ball (GB) computing has emerged as a promising paradigm that represents datasets as collections of granular regions rather than individual points, enabling compact and noise-

resistant learning [44, 45]. The Granular Ball SVM (GBSVM) [46] demonstrated the effectiveness of this approach in improving both robustness and scalability. The integration of granular ball representations into RVFL-based architectures therefore offers a principled route to simultaneously addressing noise vulnerability and scalability constraints.

1.1.4 Complex-Valued Neural Representations

Beyond robustness and scalability, improving the expressive capacity of RdNNs is another important research frontier. Complex-valued neural networks have been shown to offer substantially richer representational capacity than their real-valued counterparts [47, 48]. By encoding both magnitude and phase, complex numbers enable efficient associative memory retrieval and exhibit enhanced noise-resistant properties. From a computational perspective, Danihelka et al. [47] demonstrated the efficiency and numerical stability of holographic reduced representations [49], which exploit complex numbers for memory encoding. Biologically inspired architectures have similarly leveraged complex-valued neurons to model neural firing rates and temporal synchrony [50]. From a signal processing standpoint, the phase component of complex numbers carries critical information: phase information in speech signals influences intelligibility [51], and Oppenheim and Lim [52] demonstrated that the phase of an image can reconstruct the majority of its structural content, including shapes, edges, and orientations. Despite these well-established advantages, complex-valued representations remain largely unexplored in RVFL-based architectures, primarily due to the absence of any principled mechanism for transforming real-valued tabular data, the dominant format in practical applications, into complex-valued representations.

1.1.5 Broad Learning Systems and Deep Randomized Networks

Inspired by the RVFL architecture, the Broad Learning System (BLS) [3] was proposed as a flat neural network that employs a layer-wise approach to extract informative features through three primary components: a feature learning segment, an enhancement segment, and an output segment. Random projection mappings transform input data into a high-dimensional feature space, and only the output layer parameters are computed via the least-squares method, avoiding backpropagation entirely and substantially reducing training time. BLS possesses universal approximation capability [53], supports efficient incremental learning [3], and demonstrates superior generalization over deep learning models when training data is scarce [54]. Its architectural flexibility has inspired numerous variants, including the double-kernelized weighted BLS (DKWBLS) [55] for imbalanced data, a hybrid fuzzy BLS [56] that integrates fuzzy rules in classification and regression problems, and the broad residual learning system (BRLS) [57], which integrates norm-convergent residual operators to address performance degradation during incremental expansion. Despite these advances, BLS remains susceptible to noise and outliers due to two structural deficiencies: the indiscriminate propagation of noisy features through the feature and enhancement groups, and the uniform weighting of all training samples regardless of their reliability [58].

Conventional RVFL models are inherently limited by their single hidden layer, which restricts their ability to capture intricate and hierarchical relationships in data [59]. To overcome this, Shi et al. [2] proposed deep RVFL (dRVFL) and ensemble deep RVFL (edRVFL), which introduce multiple hidden layers and implicit ensemble learning, respectively, treating each hidden layer as an individual RVFL classifier and aggregating their outputs. The edRVFL framework has since been extended in numerous directions, including online learning variants [60], EEG-based brain-computer interface classification [61], Alzheimer's disease diagnosis [62, 63], ship order prediction [64], underwater acoustic signal processing

[65], interpretability enhancement via fuzzy inference systems [66], and time-series forecasting [67]. However, both dRVFL and edRVFL inherit the robustness deficiencies of shallow RVFL: noisy features introduced at early hidden layers propagate and accumulate through successive layers, corrupting higher-level representations, while the uniform treatment of all training samples prevents the model from distinguishing between reliable and corrupted observations. The integration of uncertainty modeling frameworks, in particular, intuitionistic fuzzy theory, which simultaneously quantifies membership and non-membership degrees, offers a principled mechanism for addressing these robustness limitations in both broad and deep randomized network architectures [39].

The foregoing review establishes the theoretical and empirical foundations upon which this thesis builds. It reveals a constellation of unresolved challenges, interpretability, robustness to noise and outliers, class imbalance, scalability, geometric structure preservation, complex-valued representation, and robustness in deep and ensemble architectures, that span the full breadth of the randomized neural network family. These challenges are elaborated as specific research gaps in Section 1.2.

1.2 Motivation and Research Gaps

RdNNs have emerged as a compelling class of models owing to their fast training speed, strong generalization capacity, and architectural simplicity. By randomly fixing the hidden-layer parameters and learning only the output layer weights, RVFL networks drastically reduce computational complexity while retaining universal approximation capability. However, despite these appealing properties, the RVFL family of models suffers from a cluster of interconnected and largely unresolved challenges pertaining to interpretability, robustness, scalability, geometric awareness, expressive representation, and uncertainty quantification.

The following discussion systematically identifies these research gaps and articulates the motivations that drive the original contributions of this thesis.

1.2.1 The Interpretability Gap in Randomized Neural Networks

ANNs are inherently opaque models, and the random assignment of hidden-layer parameters in RdNNs further compounds this opacity, rendering the internal reasoning of such models inaccessible to human scrutiny. In high-stakes application domains, including medical diagnosis, financial risk assessment, and legal decision support, interpretability is no longer a desirable property but a fundamental prerequisite for trustworthy and regulatory-compliant AI deployment [68]. Despite the critical importance of this requirement, interpretable learning mechanisms within the RdNN family remained entirely absent from the literature at the commencement of this research. This gap constitutes the first and foremost motivation of this thesis. To address it, we develop the Neuro-Fuzzy RVFL (NF-RVFL) [17] and the Ensemble Deep RVFL with Fuzzy Inference System (edRVFL-FIS) [66], both of which ground their decision-making in human-interpretable IF-THEN fuzzy rules. The supervised defuzzification mechanism, applied per layer in the ensemble structure, simultaneously enhances feature learning expressiveness while preserving full decision transparency.

1.2.2 Scalability and the Promise of Granular Computing

The closed-form solution of the RVFL network requires the inversion of the entire training data matrix, an operation whose computational cost scales cubically with the number of training samples and becomes intractable for large-scale datasets. RVFL extensions that incorporate fuzzy or graph-based mechanisms, such as IFRVFL [32] further amplify this cost by additionally requiring the computation of membership and non-membership values in kernel spaces. Consequently, despite their improved robustness to noise and geomet-

ric distortion, these models remain fundamentally constrained to small and medium-scale learning problems, a significant practical limitation. Human cognition offers an instructive alternative paradigm: the visual system prioritizes global topological features and processes information coarse-to-fine, following a "large-scope-first" principle [43]. GB computing operationalizes this principle by representing datasets as collections of granular regions rather than individual data points, thereby enabling learning algorithms to operate on compact, noise-resistant abstractions [46]. The effectiveness of this paradigm has been demonstrated in granular ball support vector machines (GBSVM) [46] and granular ball twin SVM (GB-TSVM) [69], which exhibit superior noise resilience and scalability compared to point-based counterparts. Motivated by these properties, this thesis introduces GB-RVFL and its graph-embedded extension GE-GB-RVFL [70], which inverts the matrix of GB centers rather than the full training matrix. To the best of our knowledge, this represents the first integration of GB inputs into an RVFL-based architecture, simultaneously advancing scalability, robustness, and geometric structure preservation.

1.2.3 Loss Function Sensitivity and Optimization Instability

The robustness and generalization of RVFL models are fundamentally shaped by their training objective. The standard square loss function implicitly assumes that training errors follow a Gaussian distribution [41], an assumption that is systematically violated in the presence of noise and outliers. Samples with large deviations from the expected output exert disproportionate influence during training, skewing the estimated parameters and inducing steep gradients that lead to numerical instability. Equally penalizing all errors, regardless of their origin or magnitude, further compromises the learning process by failing to distinguish between meaningful signal and spurious deviation. Collectively, these properties render the square loss ill-suited for real-world noisy datasets [42]. The recently pro-

posed wave loss function [71] has demonstrated favorable properties of smoothness, noise insensitivity, and outlier resilience in support vector machine formulations. This raises a natural and important question: can the wave loss function be systematically integrated into the RVFL optimization framework while preserving its architectural simplicity and closed-form tractability? Motivated by this question, this thesis proposes Wave-RVFL [72], an RVFL model trained under the wave loss function and optimized via the Adam algorithm [73], thereby advancing robustness in RVFL without sacrificing its computational appeal.

1.2.4 Noise, Outliers, and Class Imbalance in Standard RdNNs

Real-world datasets are routinely contaminated by noise, corrupted by outliers, and characterized by severely imbalanced class distributions. Outliers are samples that exhibit feature characteristics typical of one class while belonging to another, whereas noise refers to random distortions inherent in the data acquisition process. The class imbalance (CI) problem arises when the cardinality of one class substantially exceeds that of the remaining classes, causing standard classifiers to exhibit systematic bias towards the majority class. The conventional RVFL assigns uniform importance to every training sample during output parameter estimation, making it inherently blind to these three challenges despite its otherwise strong performance on clean and balanced datasets. Although fuzzy learning frameworks have demonstrated effectiveness in mitigating the impact of noise and outliers in machine learning models [34], their integration into RdNNs remains limited. Furthermore, and critically, the standard RVFL discards the intrinsic geometric structure of the data when computing output parameters due to the randomization of hidden-layer weights [25], a shortcoming that is particularly damaging when the topology of the feature space carries discriminative information. To the best of our knowledge, no model within the RVFL family holistically and simultaneously addresses noise, outliers, geometric structure

preservation, and class imbalance within a single unified framework. This multi-faceted gap motivates the proposal of the Graph-Embedded Intuitionistic Fuzzy RVFL for Class Imbalance Learning (GE-IFRVFL-CIL) [39], which integrates graph embedding, intuitionistic fuzzy (IF) membership functions, and CI-aware sample weighting into a coherent and theoretically grounded learning architecture.

1.2.5 Unexploited Representational Power of Complex-Valued Architectures

Recent theoretical and empirical studies have established that complex-valued neural representations possess substantially richer expressive capacity than their real-valued counterparts [47, 48]. By encoding both magnitude and phase, complex-valued networks enable efficient associative memory retrieval, exhibit enhanced noise resistance, and support biologically plausible representations of neural firing dynamics [49, 50]. Despite these well-documented advantages, the application of complex-valued representations in RdNNs has remained almost entirely unexplored. A fundamental barrier to their adoption has been the absence of any principled mechanism for transforming real-valued tabular data, the predominant format in practical machine learning, into complex-valued representations. This gap motivates the development of complex-transformed RVFL (RVFL-X) [74], a novel complex-valued extension of RVFL that introduces a systematic transformation mechanism for real-valued tabular datasets into the complex domain. To the best of our knowledge, RVFL-X constitutes the first work to systematically integrate complex-valued transformations into a real-valued RVFL architecture, unlocking a previously inaccessible dimension of representational richness in randomized networks.

1.2.6 Expressiveness and Robustness Deficiencies in Deep and Ensemble Randomized Networks

Shallow randomized neural networks (RdNNs) suffer from limited expressiveness and representation capability. To address this limitation, deep RVFL (dRVFL) and ensemble deep RVFL (edRVFL) architectures significantly enhance the representational power of shallow RVFL networks through hierarchical feature learning and ensemble aggregation. However, despite these improvements, such deeper architectures remain susceptible tacross successive hidden layers, progressively contaminating higher-level representations and distorting the decision boundaries ional capability, the theory of fuzzy inference systems [75] and the development of NF-RVFL [17] motivate the exploration of edRVFL integrated with fuzzy inference systems (edRVFL-FIS) [66]. This framework constructs unsupervised fuzzified features and supervised defuzzified features at each layer of the network, thereby enriching feature representations.

Nevertheless, noisy or corrupted features introduced in early layers tend to propagate and accumulate across successive hidden layers, progressively contaminating higher-level representations and distorting the decision boundaries at the output layer. Furthermore, the use of uniform sample weights during training prevents these models from effectively distinguishing between clean, noisy, and outlier samples, leading to degraded generalization performance in real-world scenarios [39]. Intuitionistic fuzzy (IF) theory offers a principled solution to these limitations. By jointly modeling the degrees of membership and non-membership of each sample with respect to its class, IF theory enables the assignment of data-dependent reliability weights. This mechanism effectively suppresses the influence of noisy and outlier samples during hierarchical feature learning. Motivated by this insight, we develop intuitionistic fuzzy-based deep and ensemble RVFL models, namely IF-dRVFL and IF-edRVFL [76]. These models incorporate IF-based sample weighting into deep and

ensemble randomized architectures, resulting in frameworks that are simultaneously expressive, interpretable, and robust.

1.2.7 Noise Susceptibility in Broad RdNNs

The Broad Learning System (BLS) presents a computationally efficient alternative to deep learning, offering fast incremental training and competitive generalization performance [77]. However, BLS exhibits a fundamental vulnerability to noise and outliers arising from two structural properties. First, noisy features within a training sample propagate indiscriminately through both the feature and enhancement node groups, where they become entangled with clean features, corrupting higher-level representations and distorting the learning signal. Second, BLS applies a uniform weighting scheme across all training samples, treating clean, noisy, and anomalous samples identically during output parameter estimation, an approach that is fundamentally inconsistent with the reliability differences among samples [58]. These two structural deficiencies collectively degrade the generalization performance of BLS in real-world scenarios characterized by data imperfection. Motivated by the well-established capability of intuitionistic fuzzy theory to model uncertainty and quantify sample reliability, this thesis proposes fuzzy BLS (F-BLS) and intuitionistic fuzzy BLS (IF-BLS) [78], which embed membership-based sample weighting into the broad learning paradigm, enabling principled distinction between reliable and corrupted samples and yielding substantially improved robustness.

The seven research gaps identified above, spanning interpretability, multi-faceted robustness, scalability, geometric structure preservation, loss function sensitivity, complex-valued expressiveness, and uncertainty quantification, collectively define the research agenda of this thesis. Each gap is directly addressed by a specific technical contribution, and together they advance the state of the art in randomized neural network research along four orthog-

onal but complementary axes: *interpretability*, *robustness*, *expressivity* and *scalability*. The specific research objectives formulated to address these gaps are presented in Section 1.5.

1.3 Research Challenges in the Broader Neural Network Landscape

Neural networks have become one of the most influential paradigms in modern artificial intelligence owing to their remarkable success across computer vision, natural language processing, speech recognition, healthcare, finance, and scientific computing [79, 80]. In particular, deep neural networks (DNNs) have demonstrated outstanding representation learning capabilities by automatically extracting hierarchical features from raw data, thereby significantly reducing the need for handcrafted feature engineering. Despite these successes, several fundamental challenges remain unresolved and continue to motivate active research across both conventional and randomized neural network paradigms.

One of the major challenges is the *lack of interpretability*. Modern neural networks often operate as black-box models whose decision-making processes are difficult to explain, limiting their adoption in high-stakes applications such as healthcare, autonomous systems, and finance where transparent and trustworthy predictions are essential [81, 82]. Consequently, explainable neural network architectures have become an important research direction.

Another long-standing challenge is the *robustness of neural networks*. Real-world datasets frequently contain noisy observations, outliers, missing values, and corrupted labels, all of which may substantially degrade predictive performance. Numerous robust learning algorithms, regularization techniques, and robust loss functions have therefore been proposed to improve generalization under imperfect data conditions [83, 84].

The increasing availability of highly imbalanced datasets further introduces significant

challenges for neural network learning. Conventional optimization procedures tend to bias learning towards majority classes, often resulting in poor minority-class recognition. This problem is particularly important in medical diagnosis, fraud detection, cybersecurity, and fault diagnosis, where correctly identifying rare events is often more important than maximizing overall classification accuracy.

Scalability also remains a critical issue. Contemporary deep neural networks typically require millions or even billions of trainable parameters together with iterative gradient-based optimization, making training computationally expensive and energy intensive. As datasets continue to grow in size and complexity, improving computational efficiency while maintaining predictive performance has become an increasingly important research objective.

Furthermore, although deep architectures possess remarkable representational power, effectively learning highly nonlinear relationships, preserving intrinsic geometric structures, and improving feature expressiveness remain active research topics. Consequently, recent studies have explored geometry-aware learning, complex-valued neural networks, fuzzy neural systems, and hybrid learning frameworks to improve representation learning while maintaining robustness and interpretability.

It is important to emphasize that these challenges are not unique to randomized neural networks but represent fundamental open problems across the broader neural network literature. Therefore, the research contributions presented in this thesis address several challenges that are shared by both conventional and randomized neural network models, while exploiting the unique computational advantages offered by randomized learning.

1.4 Why Randomized Neural Networks? Practical Advantages over Deep Neural Networks

Although deep neural networks have achieved state-of-the-art performance across numerous machine learning tasks, they are often associated with substantial computational and optimization costs. Training deep networks generally requires iterative backpropagation, careful hyperparameter tuning, large annotated datasets, specialized GPU hardware, and considerable training time [79, 80]. Moreover, gradient-based optimization may suffer from issues such as slow convergence, sensitivity to initialization, vanishing or exploding gradients, and convergence to poor local optima, particularly when training data are limited.

Randomized neural networks (RdNNs), including Random Vector Functional Link (RVFL) networks and Extreme Learning Machines (ELMs), provide an attractive alternative by randomly assigning hidden-layer parameters while analytically determining only the output weights [85, 86, 87]. Consequently, the training process is transformed into a linear optimization problem that can often be solved in closed form, eliminating the need for computationally expensive iterative backpropagation. This significantly reduces training time while preserving strong approximation and generalization capabilities.

The advantages of randomized neural networks become particularly evident in applications involving tabular data, small- and medium-sized datasets, real-time learning, edge computing, embedded systems, online learning, and rapid model deployment, where computational efficiency is often as important as predictive accuracy [88]. Furthermore, randomized neural networks naturally possess simple network architectures, fewer trainable parameters, deterministic optimization of the output layer, and excellent scalability, making them attractive for resource-constrained environments.

Despite these practical advantages, conventional randomized neural networks still in-

herit several limitations that are also encountered in traditional neural network models. They generally lack interpretability, remain sensitive to noisy observations and outliers, exhibit limited robustness under class imbalance, and often struggle to capture complex nonlinear representations due to their shallow randomized architectures. Moreover, preserving intrinsic geometric structures, improving feature expressiveness, and extending randomized learning to large-scale problems remain active research challenges [84, 88].

The primary motivation of this thesis is therefore not to replace deep neural networks, but rather to advance randomized neural network architectures by systematically addressing these practical limitations while preserving their inherent computational efficiency. The proposed frameworks integrate fuzzy systems, intuitionistic fuzzy learning, graph embedding, granular-ball computing, complex-valued representations, deep randomized architectures, and broad learning systems to produce a family of randomized neural networks that simultaneously improve interpretability, robustness, scalability, expressive capability, and practical applicability across a wide range of machine learning problems.

1.5 Objectives

The primary objectives of this thesis are as follows:

[1] To design and develop interpretable RdNNs that emulate human-like decision-making by integrating explainability frameworks, enabling transparent feature attribution and rule extraction in RVFL-based architectures.

[2] To propose robust and scalable RdNNs by incorporating granular computing principles and advanced loss functions, enhancing model resilience to noise, outliers, and large-scale data complexities.

[3] To develop geometry-aware and class imbalance-robust RdNNs by embedding graph-theoretic learning and manifold structures, ensuring discriminative performance on imbalanced and topologically complex datasets.

[4] To introduce a novel complex-valued randomized neural network (RVFL-X) capable of operating in the complex domain, alongside systematic methods for transforming real-valued tabular datasets into complex representations for enriched and expressive feature learning.

[5] To develop expressive, interpretable, and robust deep and ensemble RdNNs grounded in fuzzy inference systems and intuitionistic fuzzy logic, unifying depth, ensemble diversity, and fuzzy uncertainty quantification within a coherent learning framework.

[6] To construct an intuitionistic fuzzy-based robust broad randomized neural network (IF-BLS) by unifying intuitionistic fuzzy theory with the broad learning system, strengthening uncertainty handling and generalization across diverse classification tasks.

1.6 Chapters and Contributions of the Thesis

The contributions of this thesis are presented through a comprehensive literature review chapter, six research-oriented chapters, and a final chapter devoted to conclusions and future research directions. Each research chapter addresses a specific yet interconnected challenge in the development of RdNNs. Collectively, these contributions aim to advance the current state of the art by improving the interpretability, robustness, and scalability of randomized neural network models. The proposed methodologies explore various learning paradigms, including fuzzy reasoning, geometric learning, granular computing, robust optimization, complex-valued representations, and broad, deep and ensemble randomized architectures.

Together, these developments provide frameworks for designing more reliable and efficient randomized neural network models. The key contributions of the individual chapters are summarized below.

[1] **Chapter-2: Literature Review.** This chapter presents a comprehensive review of the theoretical concepts and methodologies relevant to this thesis. It introduces the foundations of RdNNs, including RVFL, BLS, and their deep and ensemble variants. The chapter also reviews important supporting frameworks such as fuzzy inference systems, intuitionistic fuzzy theory, granular ball computing, and graph embedding techniques. In addition, the mathematical notations used throughout the thesis are introduced for consistency and clarity. Finally, commonly used performance evaluation metrics and statistical testing methods for comparing machine learning models are discussed. These concepts provide the theoretical foundation for the models developed in the subsequent chapters.

[2] **Chapter-3: Interpretable Randomized Neural Networks.** A Neuro-Fuzzy RVFL (NF-RVFL) [17] model is developed by integrating the RVFL network with a fuzzy inference system, directly addressing the black-box nature of conventional RVFL architectures. The model introduces a fuzzy layer that fuzzifies input features and a supervised defuzzified layer that produces human-interpretable IF-THEN rule-based representations, enabling decisions that are transparent and aligned with human reasoning. Three variants, namely NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C, are obtained by employing randomly initialized centers, K-means, and Fuzzy C-means clustering, respectively, for fuzzy layer construction, each yielding a distinct set of fuzzified and defuzzified features. Comprehensive experiments across 29 binary, 28 multiclass, and 15 regression UCI benchmark datasets confirm the superior predic-

tive performance of NF-RVFL over baseline models. Interpretability is rigorously demonstrated through partial dependence plots (PDP) and feature importance analysis, establishing NF-RVFL as the first interpretable model within the RVFL family.

[3] **Chapter-4: Robust and Scalable Randomized Neural Networks Based on Granularity and Loss Function.** Two families of models are proposed in this chapter to simultaneously advance robustness and scalability in RVFL-based learning from complementary directions. In the first direction, the Granular Ball RVFL (GB-RVFL) and its graph-embedded extension (GE-GB-RVFL) [70] are proposed. GB-RVFL departs from the conventional point-based input paradigm by using granular ball (GB) representations of the training data as inputs, requiring only the inversion of the compact GB centers matrix rather than the full training data matrix, substantially reducing computational cost while inherently suppressing noise and outliers through the coarse granularity of GBs. GE-GB-RVFL further incorporates subspace learning criteria within a graph embedding framework to preserve the intrinsic geometric structure of the data in the projected space. To the best of our knowledge, this is the first work to integrate granular ball inputs into an RVFL-based architecture. In the second direction, a Wave-RVFL [72] model is proposed by replacing the conventional square loss function with the wave loss function, whose properties of smoothness, noise insensitivity, and outlier resilience prevent the excessive penalization of large deviations during training. The Wave-RVFL optimization problem is solved using the Adam algorithm, marking its first application in an RVFL optimization context, and is formulated to bypass matrix inversion entirely, thereby further enhancing scalability. Extensive evaluations on KEEL, UCI, NDC, and biomedical datasets validate the superiority and practical applicability of the proposed models.

[4] **Chapter-5: Geometry and Class Imbalance Aware Robust Randomized Neural Network.** A Graph-Embedded Intuitionistic Fuzzy RVFL for Class Imbalance Learning (GE-IFRVFL-CIL) [39] is proposed as a unified framework that simultaneously addresses noise, outliers, geometric structure preservation, and class imbalance within a single RVFL-based model. The framework incorporates intuitionistic fuzzy membership and non-membership functions to assign data-dependent reliability weights to training samples, suppressing the influence of noisy and outlier instances. A class imbalance weighting scheme assigns unit weight to minority class samples while proportionally reducing the weights of majority class samples. Subspace learning criteria within the graph embedding framework, employing both linear discriminant analysis (LDA) and local Fisher discriminant analysis (LFDA), are integrated into the output weight computation to preserve the intrinsic geometric topology of the data, yielding two model variants: GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2. The proposed generic framework is designed to adapt to different criteria used to handle imbalance scenarios. The efficacy of the proposed models is validated on KEEL imbalanced benchmark datasets with and without Gaussian noise across diverse domains, and on the ADNI dataset for Alzheimer's disease diagnosis, demonstrating consistent superiority over numerous state-of-the-art models.

[5] **Chapter-6: RVFL-X: A Novel Complex Randomized Network and Methods for Complex Transformation of Real-Valued Tabular Datasets.** To enhance the expressiveness of RdNNs, a novel complex-valued extension of RVFL, termed RVFL-X [74], is proposed to unlock the richer representational capacity of complex arithmetic within the randomized network family. To the best of our knowledge, this constitutes the first systematic attempt to integrate complex-valued transformations into real-valued RVFL architectures. Two principled transformation mechanisms are in-

troduced to convert real-valued tabular datasets into complex-valued representations: an intuitive natural transformation approach and an autoencoder-based transformation method. RVFL-X employs complex weights and complex activation functions throughout the network while producing real-valued outputs, preserving compatibility with standard evaluation pipelines. Comprehensive comparisons against ten state-of-the-art models across binary and multiclass classification tasks on both small-scale and large-scale datasets demonstrate the expressive advantages conferred by the complex-valued formulation.

[6] **Chapter-7: Expressive and Interpretable Ensemble Deep Randomized Neural Networks.** In this chapter, the edRVFL-FIS [66] model is proposed, and it enhances the expressive and interpretable capacity of edRVFL by augmenting each hidden layer's feature space through the concatenation of original features, unsupervised fuzzy layer features, and supervised defuzzified features derived from a fuzzy inference system. Three variants, namely edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C, are obtained using random, K-means, and Fuzzy C-means clustering for fuzzy layer construction, each generating diverse fuzzified and defuzzified representations that capitalize on inherent ensemble diversity for richer output prediction. Evaluations on UCI benchmark and large-scale NDC datasets across multiple domains validate the superiority of both model families over established baselines.

[7] **Chapter-8: Robust Deep, Ensemble Deep and Broad Randomized Neural Networks.** In this chapter, IF-dRVFL and IF-edRVFL [76] are developed by embedding intuitionistic fuzzy-based adaptive sample weighting into deep and ensemble RVFL architectures. Each training sample is assigned an intuitionistic fuzzy score derived from its membership value, based on distance to the class centroid, and its

non-membership value, capturing neighborhood heterogeneity, enabling precise discrimination among clean, noisy, and outlier instances. In IF-dRVFL, each hidden layer is individually trained using intuitionistic fuzzy-weighted samples, preventing the propagation and accumulation of corrupted features across layers. IF-edRVFL achieves implicit ensemble learning within a single deep network, avoiding the computational overhead of training multiple independent models while simultaneously benefiting from deep and ensemble learning principles.

Further, in the broad networks, Fuzzy BLS (F-BLS) and Intuitionistic Fuzzy BLS (IF-BLS) [78] are proposed to address the susceptibility of the standard Broad Learning System to noise and outliers. F-BLS assigns adaptive sample weights based on each sample's proximity to its corresponding class center, enabling differential treatment of clean and corrupted samples during output parameter estimation. IF-BLS extends this by incorporating intuitionistic fuzzy membership, non-membership, and score functions that simultaneously capture intra-class homogeneity and inter-neighborhood heterogeneity in the kernel space, yielding a more precise and principled reliability weighting mechanism. Experimental evaluations on UCI benchmark datasets with and without Gaussian noise demonstrate consistent robustness advantages over baseline BLS models across diverse domains. Application to the clinically significant task of Alzheimer's disease diagnosis further validates the practical utility of the proposed models.

[8] **Chapter-9: Comprehensive Evaluation of the Proposed Methods.** This chapter presents a comprehensive empirical evaluation of all the randomized neural network frameworks proposed throughout this thesis by consolidating their performance across diverse benchmark datasets, including binary classification, multiclass classification, imbalanced learning, and large-scale classification problems. The unified

evaluation facilitates a systematic comparison of the proposed methods under different learning scenarios using multiple performance metrics. Furthermore, the chapter provides practical guidelines for selecting the most appropriate randomized neural network framework based on specific application requirements, such as interpretability, robustness to noise and outliers, class imbalance handling, scalability, expressive representation learning, and computational efficiency, thereby establishing a unified perspective on the applicability of the proposed methodologies to real-world machine learning problems.

[9] **Chapter-10: Conclusion and Future Directions.** The final chapter summarizes the main findings and contributions of the thesis. It provides a consolidated discussion of the proposed models and highlights how the developed methodologies address the key challenges of interpretability, robustness, and scalability in RdNNs. The chapter also reflects on the broader implications of the proposed approaches and their potential applications in real-world machine learning problems. Furthermore, several promising research directions are outlined, including the extension of RdNNs to emerging learning paradigms, improvements in model interpretability, and the development of more scalable and adaptive architectures. These recommendations aim to guide future research toward further advancing the capabilities of RdNNs.

Chapter 2

Literature Review

This chapter presents a comprehensive review of the literature relevant to the methodologies and concepts employed in this thesis. The objective of this chapter is to establish the theoretical foundation of RdNNs, fuzzy learning frameworks, and related techniques that are utilized throughout the proposed models.

Section 2.1 revisits the mathematical notations and symbols used throughout the thesis to ensure clarity and consistency in subsequent formulations. Section 2.2 deals with the comprehensive discussion of existing studies that address the research gaps identified in the Introduction chapter within the broader context of neural networks. Section 2.3 discusses the shallow randomized neural network model, namely the RVFL network, along with its mathematical formulation and its closely related variant. Section 2.5 reviews the BLS, a flat randomized neural network architecture inspired by RVFL, and discusses its structure and mathematical formulation. Subsequently, Section 2.4 presents the deep and ensemble extensions of RVFL, including the dRVFL and edRVFL models, which enhance the representation learning capability of the conventional RVFL framework. Section 2.7 introduces the Takagi–Sugeno–Kang (TSK) fuzzy inference system, which plays a significant role in modeling nonlinear systems under uncertainty. Section 2.8 reviews the concept of granu-

lar ball computing, which provides an efficient data granulation mechanism for reducing data complexity while preserving essential structural information. Section 2.9 discusses the graph embedding framework, which aims to preserve the intrinsic topological structure of data during feature transformation. Section 2.6 presents the theoretical background of fuzzy and intuitionistic fuzzy membership schemes that are commonly used to handle uncertainty, noise, and outliers in machine learning models. Furthermore, Section 2.10 introduces the performance evaluation metrics used for assessing classification models. Finally, Section 2.11 discusses the statistical testing methodologies used for comparing multiple learning algorithms across different datasets.

2.1 Revisit: Notations

Let the training set be $\{(u_r, t_r) | \, u_r \in \mathbb{R}^{1\times L}, \, t_r \in \mathbb{R}^{1\times C}; r \in \{1, \ldots, M\}\}$, where L and C represent the number of features of the input sample and the number of classes, respectively with M training samples. For the classification task, the variable t_r represents the one-hot version of the label corresponding to the input sample u_r, and for the regression task, t_r represents a real value. Let $U = \left(u_1^t, u_2^t, \ldots, u_M^t\right)^t \in \mathbb{R}^{M\times L}$ and $T = \left(t_1^t, t_2^t, \ldots, t_M^t\right)^t \in \mathbb{R}^{M\times C}$ be the collection of all input and target vectors, respectively, where $(\cdot)^t$ (or, $(\cdot)^T$) is the transpose operator. $\psi(\cdot)$ is the activation function. Let k number of GBs generated for the training set be denoted as: $\{GB_1, GB_2, \ldots, GB_k\}$ and o_j be the centre of the granular ball GB_j for $j = 1, 2, \ldots, k$. $\otimes$ and $\oplus$ denote the Kronecker product and the concatenation operator, respectively and are defined below. For $C \in \mathbb{R}^{r\times s}$, $D \in \mathbb{R}^{t\times u}$ and $E \in \mathbb{R}^{t\times v}$, then

$$C \otimes D = \begin{bmatrix} c_{11}D & \ldots & c_{1s}D \\ \vdots & \vdots & \vdots \\ c_{r1}D & \ldots & c_{rs}D \end{bmatrix} \in \mathbb{R}^{rt\times su} \text{ and } D \oplus E = [D \; E] \in \mathbb{R}^{t\times(u+v)}.$$ $\mathcal{S}$ denotes the

fuzzy or intuitionistic fuzzy score matrix, while U_{uni} represents the graph embedding matrix. The parameter $\mathcal{C}$ is the regularization coefficient. Furthermore, $\mathbf{L}$ denotes the total number of hidden layers in the (e)dRVFL model, and $l \in 1, 2, \ldots, \mathbf{L}$ indexes the lth hidden layer.

2.2 Critical Review of Existing Approaches for Addressing Neural Network Challenges

The preceding sections introduced the theoretical foundations of randomized neural networks and the methodologies employed throughout this thesis. However, the research challenges identified in Chapter 1 are not unique to randomized neural networks but have been extensively investigated in the broader neural network literature. This section critically reviews representative approaches developed for addressing these challenges and discusses the remaining limitations that motivate the research contributions presented in this thesis.

2.2.1 Interpretability in Neural Networks

Interpretability has become one of the central research themes in modern artificial intelligence due to the widespread deployment of neural networks in high-stakes application domains such as healthcare, autonomous driving, finance, and scientific decision-making. Although deep neural networks have demonstrated remarkable predictive performance by learning highly expressive hierarchical representations from data, their complex nonlinear architectures often behave as black-box models whose internal decision-making processes are difficult to understand or verify [79, 80, 89]. The lack of transparency limits the trustworthiness and practical adoption of neural networks in safety-critical applications, where users and domain experts require not only accurate predictions but also understandable explanations. Consequently, improving the interpretability of neural networks has become

an active research direction, leading to the emergence of Explainable Artificial Intelligence (XAI) [81, 82].

Existing research on neural network interpretability can broadly be divided into two categories: post-hoc explanation methods and intrinsically interpretable learning models. Post-hoc explanation techniques seek to explain the predictions of an already trained black-box model without modifying its internal architecture. Representative approaches include Local Interpretable Model-Agnostic Explanations (LIME) [90], SHapley Additive exPlanations (SHAP) [91], Integrated Gradients [92], Grad-CAM [93], saliency maps, partial dependence plots, and feature attribution methods. These techniques estimate either local or global feature importance and have been successfully applied to explain deep neural networks across image analysis, natural language processing, and biomedical applications. However, because these methods generate explanations only after the learning process has been completed, the explanations are not directly incorporated into model construction and may not always faithfully represent the true decision-making mechanism of the underlying neural network [81, 82].

To overcome the limitations of post-hoc explanations, increasing attention has been devoted to developing inherently interpretable neural network architectures. Representative approaches include prototype-based neural networks, attention-based learning, sparse neural models, symbolic neural networks, and neuro-fuzzy systems that explicitly encode knowledge through human-readable IF–THEN rules [94, 95]. Among these approaches, neuro-fuzzy systems are particularly attractive because they simultaneously preserve the nonlinear approximation capability of neural networks while providing transparent rule-based reasoning. Despite these advances, the majority of research has focused on conventional gradient-based neural networks, whereas comparatively little attention has been devoted to interpretability within randomized neural networks. Existing Random Vector Func-

tional Link (RVFL) networks and Extreme Learning Machines (ELMs) primarily emphasize computational efficiency through randomized hidden-layer representations and closed-form output weight computation [22, 23, 88]. Consequently, developing inherently interpretable randomized neural network architectures that maintain both computational efficiency and transparent decision-making remains an important open research problem, which motivates the methodologies proposed in this thesis.

2.2.2 Robust Learning under Noise and Outliers

Robustness has become one of the fundamental research topics in modern neural networks because real-world datasets are rarely clean and often contain noisy observations, corrupted labels, missing values, and outliers. Such imperfections may arise from sensor failures, human annotation errors, transmission noise, environmental disturbances, or data acquisition artifacts. Since conventional neural networks optimize empirical risk over all training samples, they generally assume that every observation is equally reliable. Consequently, noisy and anomalous samples may dominate the optimization process, resulting in degraded generalization performance, unstable decision boundaries, and increased susceptibility to overfitting [79, 80, 96]. These challenges become even more severe in safety-critical applications such as medical diagnosis, autonomous driving, cybersecurity, and financial forecasting, where robustness against imperfect data is essential for reliable deployment.

To address these challenges, numerous robust learning strategies have been proposed within the broader neural network literature. One major research direction focuses on the development of robust loss functions that reduce the influence of noisy observations during optimization. Representative examples include the Huber loss [97], correntropy-based loss functions [98], generalized cross-entropy loss [99], symmetric cross-entropy loss [100], and bi-tempered logistic loss [101]. Unlike the conventional squared loss or cross-entropy loss,

these robust objectives prevent large prediction errors from disproportionately influencing parameter estimation, thereby improving robustness under label noise and outlier contamination. More recently, robust optimization methods based on uncertainty estimation, adversarial learning, and Bayesian inference have also demonstrated significant improvements in model reliability under uncertain environments [102, 103].

Another important direction involves adaptive sample selection and sample reweighting techniques that assign different importance to individual training samples according to their estimated reliability. Curriculum learning [104] and self-paced learning [105] gradually introduce difficult samples during training to improve optimization stability. Similarly, Co-teaching [106] and DivideMix [107] exploit the memorization behavior of deep neural networks to identify and suppress noisy labels during training. Sample weighting has also been extensively investigated through fuzzy systems and uncertainty-aware learning frameworks. Fuzzy membership functions [94] and intuitionistic fuzzy theory [108] provide principled mechanisms for quantifying the reliability of individual observations by simultaneously modeling their degrees of membership and uncertainty. Consequently, fuzzy neural networks have demonstrated improved robustness against noise and outliers across numerous classification and regression problems [95, 109].

Although substantial progress has been achieved in conventional neural networks, comparatively fewer studies have investigated robustness within randomized neural networks. Existing Random Vector Functional Link (RVFL) networks and Extreme Learning Machines (ELMs) primarily emphasize computational efficiency by randomly generating hidden-layer parameters and analytically determining output weights [22, 23, 88]. However, most randomized neural networks continue to assign identical importance to all training samples, making them vulnerable to noisy observations and outliers. Recent efforts have attempted to incorporate fuzzy learning into RVFL architectures to improve robust-

ness [32, 37], yet these approaches generally address only individual aspects of robustness without simultaneously considering uncertainty modeling, geometric structure preservation, deep randomized learning, and broad learning architectures. Consequently, developing a unified family of robust randomized neural networks capable of effectively handling noise, outliers, and uncertainty while preserving the computational efficiency of randomized learning remains an important open research problem. This observation directly motivates the intuitionistic fuzzy-based randomized neural network frameworks developed in the subsequent chapters of this thesis.

2.2.3 Gradient-based Explanations

Gradient-based explanation methods constitute one of the most widely adopted categories of post-hoc interpretability techniques for deep neural networks. Unlike model-agnostic explanation methods, these approaches exploit the gradients of the prediction function with respect to the input features or intermediate feature maps to estimate the contribution of individual features towards a particular prediction. Since modern deep neural networks are differentiable end-to-end, gradients naturally provide information regarding the sensitivity of the network output to small perturbations in the input, thereby offering insight into the internal decision-making process [80, 110]. Owing to their computational efficiency and compatibility with a broad range of neural network architectures, gradient-based explanations have become standard tools for interpreting convolutional neural networks (CNNs), recurrent neural networks (RNNs), graph neural networks (GNNs), and transformer-based models across computer vision, natural language processing, and medical image analysis [79, 93].

One of the earliest gradient-based explanation methods is the saliency map proposed by Simonyan *et al.* [110], which computes the gradient of the class score with respect to

the input image to identify pixels that most strongly influence the prediction. Although simple and computationally efficient, saliency maps often suffer from noisy visualizations and gradient saturation, limiting their interpretability. To overcome these shortcomings, Sundararajan *et al.* [92] introduced Integrated Gradients (IG), which computes feature attributions by integrating gradients along a continuous path from a baseline input to the actual input. Integrated Gradients satisfy desirable theoretical properties such as implementation invariance and sensitivity, making them one of the most theoretically grounded attribution methods. Other notable variants include DeepLIFT [111], which compares neuron activations against reference activations, Guided Backpropagation [112], and SmoothGrad [113], which averages gradients over multiple noisy perturbations of the input to reduce visual noise and improve attribution stability.

For convolutional neural networks, Gradient-weighted Class Activation Mapping (Grad-CAM) has become one of the most influential visualization techniques [93]. Grad-CAM utilizes the gradients flowing into the final convolutional layer to generate coarse localization maps that highlight the image regions most responsible for a particular prediction. Numerous extensions have subsequently been proposed, including Grad-CAM++ [114], Score-CAM [115], XGrad-CAM [116], and Eigen-CAM [117], each aiming to improve localization accuracy, visual quality, and robustness. These techniques have been widely adopted in medical image analysis, remote sensing, autonomous driving, and industrial inspection due to their intuitive visual explanations.

Despite their widespread success, gradient-based explanation methods possess several inherent limitations. Since they rely on local gradients, the generated explanations may become unstable in regions where gradients vanish or explode, particularly in very deep architectures [80]. Furthermore, gradient-based explanations are inherently post-hoc and therefore do not modify the learning mechanism itself. Several studies have also shown that

saliency maps can be sensitive to small perturbations, adversarial examples, and model reparameterization, potentially reducing their reliability for safety-critical applications [81, 118]. Consequently, increasing attention has shifted towards inherently interpretable neural networks that integrate explainability directly into the learning architecture rather than relying solely on post-training visualizations. In the context of randomized neural networks, gradient-based explanations are considerably less applicable because models such as RVFL and ELM determine their output weights analytically instead of learning hierarchical representations through iterative gradient optimization. This limitation provides additional motivation for developing intrinsically interpretable randomized neural network architectures, such as the neuro-fuzzy frameworks proposed in this thesis.

2.2.4 Learning from Imbalanced Data

Class imbalance is one of the most prevalent challenges encountered in supervised learning, particularly in biomedical diagnosis, fraud detection, fault detection, remote sensing, and cybersecurity applications, where the number of minority class samples is substantially smaller than that of the majority class. Conventional neural networks generally optimize empirical risk by assuming balanced class distributions and equal misclassification costs for all classes. Consequently, the learning process becomes biased towards the majority class, often resulting in poor sensitivity and reduced recognition performance for minority classes despite achieving high overall classification accuracy [79, 80, 119]. This problem becomes increasingly severe in medical diagnosis, where minority classes often correspond to clinically important disease categories, making accurate detection significantly more important than maximizing overall accuracy.

Existing approaches for handling class imbalance can generally be categorized into three groups: data-level methods, algorithm-level methods, and hybrid approaches. Data-

level methods primarily modify the training distribution through oversampling or undersampling strategies. Representative techniques include the Synthetic Minority Over-sampling Technique (SMOTE) [120], Adaptive Synthetic Sampling (ADASYN) [121], Borderline-SMOTE [122], and numerous generative adversarial network (GAN)-based oversampling approaches. Although these techniques effectively increase minority class representation, they may introduce redundant synthetic samples, increase computational complexity, and occasionally amplify noise present in the minority class. Algorithm-level methods instead modify the optimization objective by incorporating cost-sensitive learning, weighted loss functions, focal loss [123], class-balanced loss [124], and margin-aware optimization strategies to improve discrimination of minority classes without altering the underlying data distribution.

More recently, deep learning research has explored imbalance-aware representation learning through metric learning, contrastive learning, graph neural networks, and self-supervised learning to improve feature discrimination under skewed data distributions [125**?**]. Ensemble learning has also demonstrated considerable success by combining multiple complementary classifiers to improve minority class recognition and reduce prediction variance [126]. Furthermore, fuzzy systems have been widely integrated with neural networks to address uncertainty and class overlap by assigning adaptive membership values to training samples according to their confidence or importance [94, 109]. Such adaptive weighting mechanisms naturally reduce the influence of ambiguous observations while emphasizing informative minority samples during learning.

Despite these advances, relatively limited attention has been devoted to imbalance learning in randomized neural networks. Existing Random Vector Functional Link (RVFL) networks and Extreme Learning Machines (ELMs) primarily optimize overall prediction accuracy while assigning equal importance to all training samples during analytical output

weight estimation [22, 23, 88]. Consequently, these models remain susceptible to biased learning under highly imbalanced datasets. Although several studies have incorporated cost-sensitive learning or ensemble strategies into randomized neural networks, the simultaneous treatment of class imbalance, uncertainty modeling, geometric structure preservation, and computational efficiency remains largely unexplored. These limitations motivate the granular intuitionistic fuzzy and ensemble randomized neural network frameworks proposed in this thesis, which seek to improve minority class discrimination while preserving the computational advantages of randomized learning.

2.2.5 Complex-valued Neural Networks

Complex-valued neural networks (CVNNs) have emerged as an important extension of conventional real-valued neural networks by representing network parameters, activations, and feature representations in the complex domain. Unlike real-valued models that process only magnitude information, complex-valued neural networks simultaneously encode both magnitude and phase information, thereby providing richer feature representations for learning complex nonlinear relationships [127, 128]. This additional representational capability has made CVNNs particularly attractive for applications involving naturally complex-valued signals, including radar signal processing, wireless communications, electromagnetic imaging, quantum computing, speech analysis, medical imaging, and electroencephalography (EEG) analysis [129, 130].

The development of deep complex-valued neural networks has significantly expanded the applicability of CVNNs beyond traditional signal processing. Recent studies have introduced complex-valued convolutional neural networks, recurrent neural networks, transformers, and graph neural networks capable of directly processing complex-valued data while preserving phase relationships throughout the learning process [129, 131]. Several

investigations have demonstrated that complex-valued representations often achieve superior convergence characteristics, improved feature discrimination, and enhanced robustness when compared with equivalent real-valued architectures, particularly for oscillatory and frequency-domain signals [127]. Moreover, advances in complex-valued optimization algorithms, activation functions, and backpropagation techniques have substantially improved the stability and practical applicability of deep complex-valued learning [128].

Despite these advantages, several challenges continue to limit the widespread adoption of complex-valued neural networks. The mathematical formulation of complex-valued optimization is considerably more involved than that of real-valued learning because differentiation must satisfy the Cauchy–Riemann conditions or employ Wirtinger calculus for gradient computation [132]. Furthermore, the design of suitable activation functions, weight initialization strategies, normalization techniques, and regularization methods remains considerably more challenging than in conventional neural networks [129]. Consequently, most existing complex-valued neural networks rely on computationally intensive gradient-based optimization, resulting in higher training complexity and longer convergence times.

Comparatively fewer studies have investigated complex-valued representations within randomized neural networks. Existing complex-valued RVFL and Extreme Learning Machine architectures primarily exploit random complex-valued feature mappings while preserving the computational efficiency associated with analytical output weight estimation [22, 23, 88]. However, existing approaches generally neglect uncertainty modeling, graph-based feature relationships, fuzzy reasoning, and deep randomized architectures. Therefore, integrating complex-valued feature learning with randomized neural networks represents a promising research direction for simultaneously improving representational capability, computational efficiency, and robustness. These observations provide the primary motivation for the complex-valued randomized neural network models developed in this thesis.

2.3 Shallow Randomized Neural Network: RVFL [1]

The ability of ANNs to approximate nonlinear mappings is the primary reason for their huge success in many disciplines among numerous machine learning approaches [9]. ANNs have demonstrated success in various fields such as rainfall forecasting [10], clinical medicine [11], stock market predictions [12], solving differential equations [13], brain age prediction [14] and so on.

The gradient descent (GD)-based method, an iterative process, is one of the most often used techniques for optimizing the cost function to train ANNs. In the GD-based technique, the difference between the real output and the anticipated output of the model backpropagates in an effort to optimize the weights and biases of the model. This iterative strategy has a number of underlying issues, such as being time-consuming having a tendency to converge to local rather than global optima [18], being extremely sensitive to the choice of the learning rate and the point of initialization of the iteration.

RdNNs [19] were proposed to avoid the pitfalls of GD-based neural networks. In RdNN, some of the network parameters are fixed during the training period, and only the parameters of the output layer are calculated via the closed-form solution [20]. RVFL network [1], and RVFL without direct link (RVFLwoDL) (also known as extreme learning machine (ELM)) [22] are among the prominent RdNNs. The distinctive feature setting RVFL apart from other RdNNs lies in its establishment of direct linkages between the input and output layers. The weights and biases within the hidden layer of the RVFL are randomly generated and kept fixed throughout the training phase. The output parameters, namely direct link weights and the weights connecting the hidden layer to the output layer, are analytically calculated using the pseudo-inverse or the least-square method. The incorporation of direct links in the RVFL has been observed to significantly improve the learning performance by functioning as a

regularization for the randomization [23]. Furthermore, the thinner topology of the RVFL, compared to the RVFLwoDL, aids in reducing its complexity and enhancing attractiveness in alignment with Occam's principle and probably approximately correct (PAC) learning theory [2, 24]. The RVFL offers fast training speed as well as universal approximation ability [21].

2.3.1 Mathematical Formulation: Shallow RVFL

The RVFL neural network is structured with three layers: input, hidden, and output. Weights connecting the input and hidden layers and biases at the hidden layer are randomly generated and fixed throughout training. The hidden layer features and the original input features (through direct links) are connected to the output layer without any associated bias. The output weights are calculated using analytical methods such as the Moore-Penrose inverse or the least square method. Figure 6.1 depicts the architecture of RVFL. Let H_1 be the

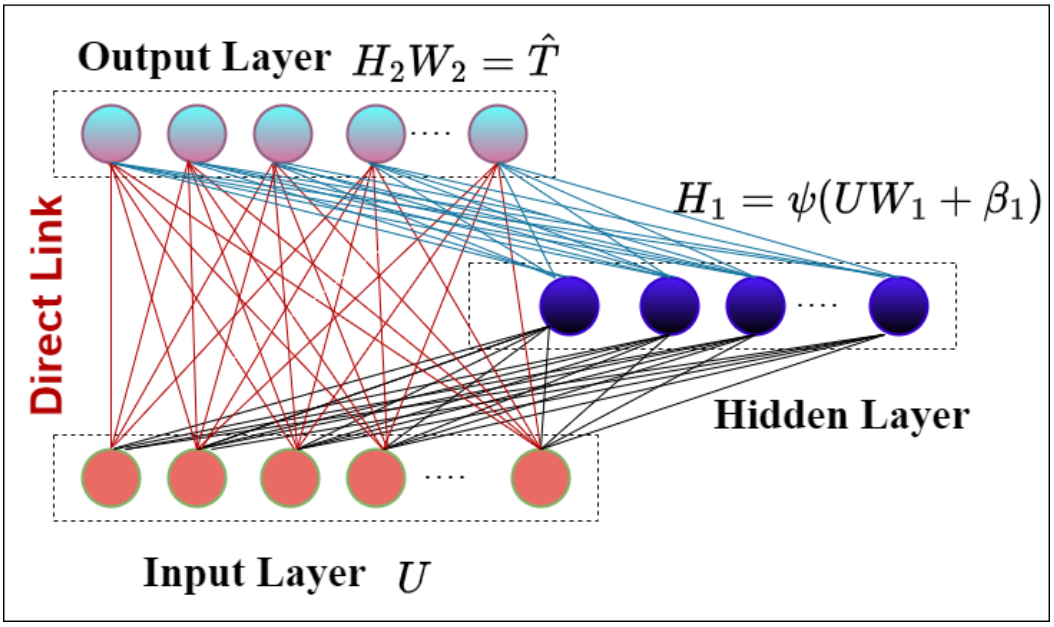


Figure 2.1: The architecture of the RVFL model.

hidden layer matrix obtained by the projection of the input matrix U using randomly initial-

ized weights and biases followed by the non-linear activation function ψ, which is defined as:

$$H_1 = \psi(UW_1 + \beta_1) \in \mathbb{R}^{M \times h}, \tag{2.1}$$

where $W_1 \in \mathbb{R}^{L \times h}$ represents the randomly initialized weights obtained from a uniform distribution within the range of $[-1, 1]$ (following [23]), $\beta_1 \in \mathbb{R}^{M \times h}$ is the bias matrix and h is the number of hidden nodes. Therefore, H_1 is given as:

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

where $u_i \in \mathbb{R}^{1 \times L}$ is the ith row (sample) of the inputs matrix U, $w_j \in \mathbb{R}^{L \times 1}$ is the jth column vector of the weights matrix W_1 which connects all the input features to the jth node of the hidden layer and ζ_j is the bias term of the jth hidden node. The input and hidden features are concatenated as:

$$H_2 = \begin{bmatrix} U & H_1 \end{bmatrix} \in \mathbb{R}^{(M \times (L+h))}. \tag{2.2}$$

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

$$H_2 W_2 = \hat{T}, \tag{2.3}$$

where $W_2 \in \mathbb{R}^{(L+h) \times C}$ is the weights matrix connecting the input layer and hidden layer to the output layer. The resultant optimization problem from Eq. (2.3) is formulated as:

$$(W_2)_{\min} = \underset{W_2}{\operatorname{argmin}} \frac{\mathcal{C}}{2} \|H_2 W_2 - T\|_2^2 + \frac{1}{2} \|W_2\|_2^2, \tag{2.4}$$

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

The solution of (2.4) is given as:

$$(W_2)_{\min} = \begin{cases} \left({H_2}^t H_2 + \frac{1}{\mathcal{C}} I\right)^{-1} {H_2}^t T, & (L+h) \leq M, \\ {H_2}^t \left(H_2 {H_2}^t + \frac{1}{\mathcal{C}} I\right)^{-1} T, & M < (L+h), \end{cases} \tag{2.5}$$

where I is an identity matrix of conformal dimension.

2.3.2 RVFL without Direct Link (RVFLwoDL): Extreme Learning Machine (ELM)

ELM (RVFLwoDL) [22] is one of the most popular RdNNs. The distinguishing factor between the topology of RVFLwoDL and RVFL is the presence of direct connections from the input to the output layer in RVFL, whereas such connections are absent in RVFLwoDL. The optimization problem of the regularized RVFLwoDL is given by:

$$\beta_{\min} = \underset{\beta}{\operatorname{argmin}} \frac{C}{2} \|H_1 \beta - T\|_2^2 + \frac{1}{2} \|\beta\|_2^2. \tag{2.6}$$

The solution of (2.6) is given as follows:

$$\beta_{\min} = \begin{cases} \left({H_1}^T H_1 + \frac{1}{C} I\right)^{-1} {H_1}^T T, & h \leq M, \\ {H_1}^T \left(H_1 {H_1}^T + \frac{1}{C} I\right)^{-1} T, & M < h. \end{cases} \tag{2.7}$$

2.4 Deep and Ensemble Deep RdNNs

The shallow RVFL model is associated with certain limitations, notably an unstable classifier [59], and its reliance on a single hidden layer hinders its ability to effectively capture the intricate hidden relationships present in the data. To overcome those limitations, within the standard RVFL model, notable enhancements have been implemented using two

distinct research frameworks: ensemble learning [133] and deep learning [134]. Shi et al. [2] proposed deep RVFL (dRVFL) and ensemble dRVFL (edRVFL) networks. The dRVFL introduces a deep architecture that can accommodate multiple hidden layers and captures complex relationships within the dataset. The edRVFL network is based on implicit ensemble learning and leverages each hidden layer as a classifier to form an ensemble. The incorporation of multiple RVFL models (base models) within the edRVFL framework enhances its stability and effectiveness compared to the shallow RVFL model.

Over the years, several variants based on the edRVFL architecture have been developed. The original edRVFL lacks provisions for online learning. Recognizing this gap, Gao et al. [60] introduced the dynamic edRVFL, a variant that introduces three crucial online components: online decomposition, online training, and online dynamic ensemble. In the field of cognitive research, Li et al. [61] proposed the spectral-edRVFL (SedRVFL) network, which specializes in feature learning from electroencephalogram (EEG) data to enhance passive brain-computer interface (pBCI) classification tasks. In [62], authors focused on medical imaging, where they combined magnetic resonance imaging (MRI) and positron emission tomography (PET) scans using a convolutional neural network (CNN) and an ensemble of non-iterative RVFL models for diagnosing Alzheimer's disease. Cheng et al. [64] contributed an edRVFL variant tailored for predictive analytics, specifically in the context of ship order book dynamics. He et al. [65] presented the model-embedded self-supervised edRVFL (ME-SedRVFL) network for underwater signal processing, particularly in estimating the direction-of-arrival (DOA) of acoustic signals. These advancements underscore the versatility of the edRVFL model across various domains, ranging from cognitive neuroscience to signal processing. Below, we discuss the mathematical formulations of the standard dRVFL and edRVFL.

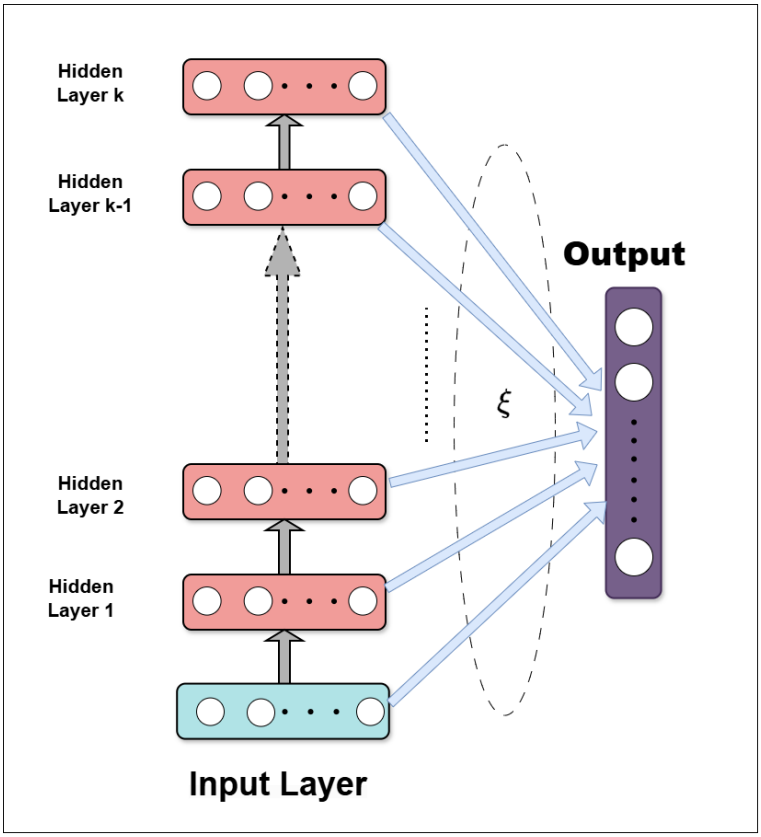


Figure 2.2: The architecture of the dRVFL model.

2.4.1 Mathematical Formulation of dRVFL [2]

Within the dRVFL architecture, multiple hidden layers are vertically stacked. These hidden layers' weights are initially generated randomly and remain unchanged throughout the training phase. Afterward, the features from all hidden layers, combined with the original features, are fused together, resulting in an augmented feature set. The output layer's weights are then computed using an analytical method. Let there be k number of hidden layers, then the first hidden layer's output of dRVFL is defined as:

$$E^{(1)} = \psi(U\omega^{(1)}), \tag{2.8}$$

and the higher hidden layers' ($g > 1$) output is defined as:

$$E^{(g)} = \psi(E^{(g-1)}\omega^{(g)}), \;\; g = 1, 2, \ldots, k. \tag{2.9}$$

Here, h is the number of hidden nodes in the hidden layer. $\omega^{(1)}$ and $\omega^{(g)}$ are the randomly initialized weights for the first and g^{th} hidden layers, respectively. The enhanced feature matrix is defined as:

$$E^* = [E^{(k)}, E^{(k-1)}, \ldots, E^{(1)}, U]. \tag{2.10}$$

The dRVFL's predicted output is given as:

$$\hat{T} = E^*\xi, \tag{2.11}$$

where $\xi \in \mathbb{R}^{(kh+m)\times c}$ denotes weights of the output layer and is calculated as:

$$\xi = E^{*\dagger}T, \tag{2.12}$$

where $E^{*\dagger}$ denotes the pseudo-inverse of E^*. We can employ the least-square method as well to calculate ξ. The architecture of the dRVFL is shown in Fig. 2.2.

2.4.2 Mathematical Formulation of edRVFL [2]

The Ensemble deep RVFL (edRVFL) model was conceived by merging implicit ensemble learning principles into the dRVFL architecture, effectively combining the strengths of both ensemble and deep learning frameworks. Within edRVFL, every base model (hidden layer) receives inputs comprising both the original features and the randomized features computed by the preceding layer. This integration enriches the learning process. The output

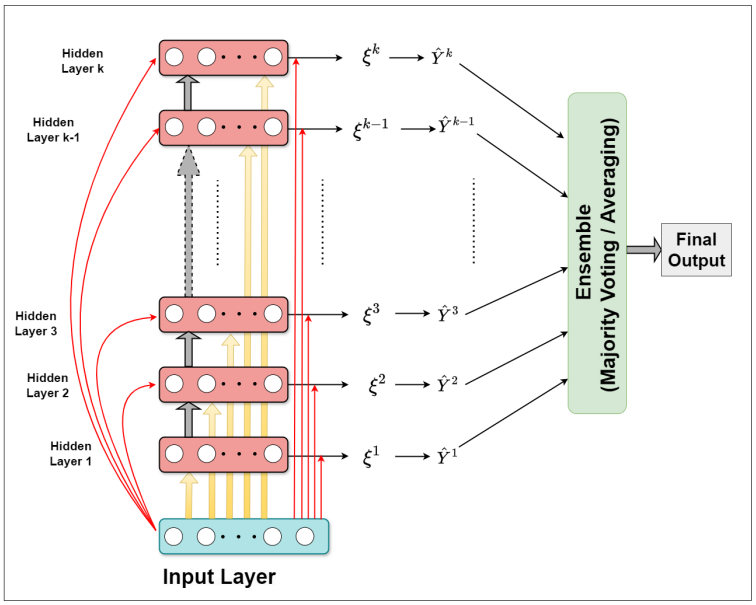


Figure 2.3: The architecture of the edRVFL model.

from the first hidden layer in edRVFL is given as follows:

$$E^{(1)} = \psi(U\omega^{(1)}), \tag{2.13}$$

and the higher hidden layers' ($g > 1$) output is defined as:

$$E^{(g)} = \psi([E^{(g-1)}, U]\omega^{(g)}), \quad g = 1, 2, \ldots, k, \tag{2.14}$$

where $\omega^{(1)}$ and $\omega^{(g)}$ ($g > 1$) represent the randomly generated weights for the first and g^{th} ($g > 1$) hidden layers, respectively. Each hidden layer in the edRVFL comprises a base model (RVFL). The output of the 1st base model (generated from the first hidden layer) is defined as:

$$\hat{T^1} = [E^{(1)}, U]\xi^1, \tag{2.15}$$

where ξ^1 are the output layer weights for the first base model. The gth base model's output is defined as follows:

$$\hat{T^g} = [E^{(g)}, E^{(g-1)}, U]\xi^g. \tag{2.16}$$

Here, ξ^g are the output layer weights for the gth base model and calculated using (2.12). Finally, the decision is taken by consolidating the outcomes from all base models (hidden layers) through averaging or majority voting methods. The architecture of the edRVFL is shown in Fig. 2.3.

2.5 Broad Randomized Neural Network

The structure and operations of biological neurons in the brain serve as the basis for the class of machine learning techniques known as ANNs. ANNs are composed of interconnected nodes (neurons) that use mathematical operations to process and transfer information. ANNs are designed to discover patterns and correlations in data and then utilize that knowledge to make predictions. Deep learning is one of the most prominent branches of machine learning that employs ANNs with multiple layers to perform complex tasks such as speech recognition [15], natural language processing [16], feature interpretation [17] and so on.

One of the prime advantages of deep learning is its capacity to automatically extract high-level features from raw data, eliminating the need for manual feature engineering. Deep learning also has the upper hand of being able to generalize effectively to new data, which allows it to make correct predictions even on data that has never been shown to it before. Apart from many advantages, there are some challanges of deep learning architectures: (a) Deep learning models consist of numerous layers and millions of parameters, making

them extremely complicated. (b) The iterative learning process in deep learning models can be computationally intensive and time-consuming. (c) Training deep learning models often requires powerful hardware, such as GPUs or TPUs. A shortage of such hardware may hinder the training process. These drawbacks confine the applicability of deep learning models primarily to tasks such as speech and image recognition. Consequently, it gives space to other models to spread their wings.

RdNNs [135, 136] are neural network that incorporates randomness in the topology and learning process of the model. This randomness allows RdNNs to learn with fewer tunable parameters in less computational time and without the requirement of savvy hardware. ELM [22, 137] and RVFL [1, 25] are two popular RdNNs. Both ELM and RVFL use the least-square method (provides a closed-form solution) to determine the output parameters. The direct links from the input layer to the output layer in RVFL significantly improve the generalization performance by functioning as an intrinsic regularization tool [23]. In addition, RVFL provides quick training speed and universal approximation capability [21].

Recently, the BLS [3] (inspired by RVFL), a class of flat neural networks, was proposed. BLS employs a layer-by-layer approach to extract informative features from the input data. BLS has three primary segments: a feature learning segment, an enhancement segment, and an output segment. In the feature learning and enhancement segment, the input data is transformed into a high-dimensional feature space using a set of random projection mappings. As a result of the feature learning and enhancement segment, a collection of high-dimensional features is produced that effectively extract important information from the input data. To train the BLS model, only the output layer parameters need to be computed by the least-squares method. In comparison to traditional deep neural networks (DNNs), BLS has several advantages. For example, it does not employ backpropagation, which might simplify the training process and lessen the likelihood of overfitting; its ability

to learn from a small number of training samples; and faster training speed due to closed form solution. The universal approximation capability of BLS [53] makes it more promising among researchers. The adaptable topology of BLS makes it possible to train and update the model effectively in an incremental way [3], and when training data is scarce, BLS may have superior generalization performance than deep learning models [54].

The BLS has been applied in various applications. In [138], recurrent-BLS (R-BLS) and gated-BLS (G-BLS) models were developed and applied for text classification. In [139], a limited penetrable visibility graph (LPVG)-based BLS method was used for the visual evoked potential (VEP)-based Brain–computer interface (BCI) system. The graph convolutional broad network (GCB-net) and BLS were employed collaboratively in recognizing emotions through EEG signals in [140]. Using a knowledge module and a broad selective sampling module, Wang et al. [141] suggested a knowledge-augmented BLS (KABLS) that offers a multichannel selective sampling network to decouple the mixed-type pattern defects of wafer maps. In order to increase communication efficiency and lower communication costs in a communication distributed architecture, Liang et al. [142] introduced the decentralized learning communication-efficient algorithm for elastic-net BLS.

There has been a range of distinct variants of the BLS architecture proposed in the literature, each with its unique features, advantages, and limitations. The double-kernelized weighted BLS (DKWBLS) [55] was developed to manage the imbalanced data. Wang et al. [143] proposed a BLS-based multi-modal material identification paradigm. This technique lowers the joint feature dimensionality by optimizing the correlation between two features of the input samples. In [144], the authors proposed a time-varying iterative learning algorithm utilizing GD for optimizing the parameters in the enhancement layer, feature layer, and output layer. In [56], a hybrid neuro-fuzzy BLS (NeuroFBLS) was proposed by combining a human-like reasoning approach based on a set of IF-THEN fuzzy rules with the learning

and linking structure of the BLS. NeuroFBLS employs the k-means clustering technique to create fuzzy subsystems, thereby increasing the computational burden of the model.

Below, we give the detailed mathematical formulation of the BLS.

2.5.1 Mathematical Formulation of BLS [3]

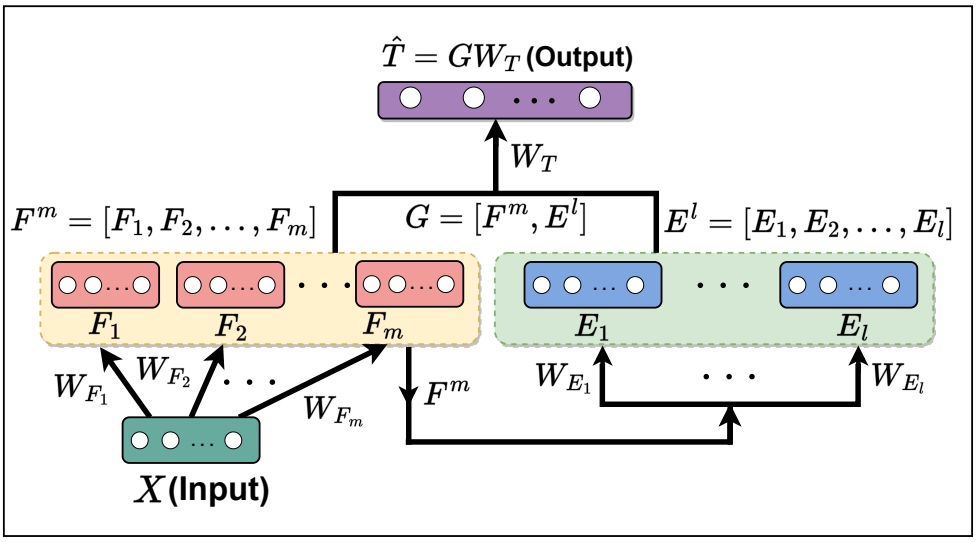


Figure 2.4: The architecture of the BLS model.

The architecture of the BLS is shown in Figure 2.4. We briefly give the mathematical formulations of the BLS model.

Segment-1: Let there be m groups and each feature group having p nodes. Then the ith feature group is given as follows:

$$F_i = \mathcal{F}_i \left(UW_{F_i} + \beta_{F_i} \right) \in \mathbb{R}^{M \times p}, \quad i = 1, 2, \ldots, m, \tag{2.17}$$

where $\mathcal{F}i$, $W_{F_i} \in \mathbb{R}^{L \times p}$ and $\beta_{F_i} \in \mathbb{R}^{M \times p}$ are feature map, randomly generated weight matrix, and bias matrix for the ith feature group, respectively. The augmented output of the

m number of feature groups is:

$$F^m = [F_1, F_2, \ldots, F_m] \in \mathbb{R}^{M \times mp}. \tag{2.18}$$

Segment-2: Augmented feature matrix (F^m) is projected to enhancement spaces via random transformations followed by the activation functions. Let l be the number of enhancement groups, and each enhancement group has q nodes. Then

$$E_j = \psi\left(F^m W_{E_j} + \beta_{E_j}\right) \in \mathbb{R}^{M \times q}, \;\; j = 1, 2, \ldots, l. \tag{2.19}$$

Where ψ is the activation function used to generate nonlinear features. $W_{E_j} \in \mathbb{R}^{mp \times q}$ is a randomly generated weight matrix connecting the augmented feature matrix (F^m) to j^{th} enhancement group, and $\beta_{E_j} \in \mathbb{R}^{M \times q}$ is the bias matrix corresponding to the jth enhancement group. The resultant output of the l enhancement groups is

$$E^l = [E_1, E_2, \ldots, E_l] \in \mathbb{R}^{M \times lq}. \tag{2.20}$$

Segment-3: Finally, the resultant enhancement matrix (E^l) along with the augmented feature matrix (F^m) is sent to the output layer and the final outcome is calculated as follows:

$$\hat{T} = [F^m, E^l] W_T = G W_T, \tag{2.21}$$

where

$$G = [F^m, E^l] \in \mathbb{R}^{M \times (mp+lq)} \tag{2.22}$$

is the concatenated matrix, and $W_T \in \mathbb{R}^{(mp+lq) \times C}$ is the unknown weights matrix connecting the augmented feature layer and resultant enhancement layer to the output layer. Finally, W_T

is calculated using the pseudo-inverse as: $W_T = G^{\dagger}\hat{T}$, where $G^{\dagger}$ denotes the pseudo-inverse of G.

2.6 Fuzzy Theory

Fuzzy theory is successfully employed in machine learning models [39, 145, 146] to ameliorate the adverse effects of noise or outliers on model performance. The fuzzy membership function uses sample-to-class center distance to generate a degree of membership to deal with outliers and noise. In the IF twin support vector machine (IF-TSVM) [33], the IF score was used as an advanced concept to mitigate noise and outliers in the TSVM model. The IF theory uses membership and nonmembership functions to assign an IF score to each sample, effectively handling noise and outliers. Membership and non-membership values of a sample measure the degree of belongingness and non-belongingness of that sample to a particular class, respectively. Below, we give mathematical definitions of fuzzy and intuitionistic fuzzy-based membership functions.

2.6.1 Fuzzy Membership Scheme

Let U be the crisp set and $\vartheta : U \rightarrow [0, 1]$ be the membership function which assigns fuzzy membership values to each sample $u_r \in U$ [147]. Define $\mathcal{A} = \left\{(u_r, t_r, \vartheta(u_r))|\, u_r \in \mathbb{R}^{1\times L},\, t_r \in \mathbb{R}^{1\times C},\, r = 1, 2, \ldots, M\right\}$ be the fuzzy set. The membership function for training sample u_r is defined as:

$$\vartheta\left(u_r\right) = \begin{cases} 1 - \frac{\|u_r - C_{pos}\|}{R_{pos}+\delta}, & t_r \in +1 \text{ class}, \\ 1 - \frac{\|u_r - C_{neg}\|}{R_{neg}+\delta}, & t_r \in -1 \text{ class}, \end{cases} \tag{2.23}$$

where $C_{pos}(C_{neg})$ and $R_{pos}(R_{neg})$ are the class center and the the radius of the positive (negative) class, respectively, and δ is a very small positive parameter. $t_r = +1\ (-1)$ denotes the

positive (negative) class. The membership function takes into account the distance between each sample and its associated class center. The center of each class is defined as:

$$C_{pos} = \frac{1}{N_{pos}} \sum_{t_r=+1} u_r \text{ and } C_{neg} = \frac{1}{N_{neg}} \sum_{t_r=-1} u_r, \tag{2.24}$$

where N_{pos} (N_{neg}) is the total number of samples in the positive (negative) class. The radii are defined as:

$$R_{pos} = \max_{t_r=+1} \|u_r - C_{pos}\| \text{ and } R_{neg} = \max_{t_r=-1} \|u_r - C_{neg}\|. \tag{2.25}$$

Finally fuzzy score matrix $\mathcal{S}$ for the dataset U is defined as:

$$\mathcal{S} = diag\left(\vartheta(u_1), \vartheta(u_2), \ldots, \vartheta(u_M)\right). \tag{2.26}$$

2.6.2 Intuitionistic Fuzzy Membership (IFM) Scheme

The intuitionistic fuzzy set (IFS) [33, 148] was defined to cope with uncertainty difficulties, and it allows for a precise simulation of the situation using current information and observations [33]. Here, the IFM scheme for the crisp set U is discussed by projecting the dataset U onto a kernel space. An IFS is defined as: $\widetilde{\mathcal{A}} = \left\{(u_r, t_r, \theta(u_r), \widetilde{\theta}(u_r)| \; r = 1, 2, \ldots, M\right\}$, where $\theta(u_r)$ and $\widetilde{\theta}(u_r)$ are the membership and non-membership values of u_r.

- **The membership mapping:** The membership function $\theta : X \to [0, 1]$ is defined as:

$$\theta\left(u_r\right) = \begin{cases} 1 - \frac{\|\varphi(u_r) - C_{pos}\|}{R_{pos} + \delta}, & t_r \in +1 \text{ class}, \\ 1 - \frac{\|\varphi(u_r) - C_{neg}\|}{R_{neg} + \delta}, & t_r \in -1 \text{ class}, \end{cases} \tag{2.27}$$

  where φ represents the kernel mapping. C_{pos}, C_{neg}, R_{pos} and R_{neg} are defined in the same way as mentioned in Subsection 2.6.1, with replacing u_r by $\varphi(u_r)$.

- **The non-membership mapping:** In the IFM scheme, each training sample u_r is additionally allocated a non-membership value that represents the ratio of heterogeneous data points to all data points in its vicinity. The non-membership function $\widetilde{\theta} : U \to [0, 1]$ is defined as: $\widetilde{\theta}(u_r) = \big(1-\theta(u_r)\big)\Theta(u_r)$, where $0 \leq \theta(u_r)+\widetilde{\theta}(u_r) \leq 1$ and Θ is calculated as follows:

$$\Theta(u_r) = \frac{|\{u_l : \|\varphi(u_r) - \varphi(u_l)\| \leq \epsilon,\ t_r \neq t_l\}|}{|\{u_l : \|\varphi(u_r) - \varphi(u_l)\| \leq \epsilon\}|}, \tag{2.28}$$

  where ϵ is the adjustable parameter used to create a neighborhood and $|\cdot|$ represents the cardinality of a set.

- **The score mapping:** After calculating the membership and non-membership values of each sample, an IF score (IFS) function is defined as follows:

$$\Theta^{\star}(u_r) = \begin{cases} \theta(u_r), & \widetilde{\theta}(u_r) = 0, \\ 0, & \theta(u_r) \leq \widetilde{\theta}(u_r), \\ \frac{1-\widetilde{\theta}(u_r)}{2-\theta(u_r)-\widetilde{\theta}(u_r)}, & \text{otherwise.} \end{cases} \tag{2.29}$$

- **The score matrix:** Finally, the score matrix $\mathcal{S}$ for the dataset U is defined as:

$$\mathcal{S} = diag\left(\Theta^{\star}(u_1), \Theta^{\star}(u_2), \ldots, \Theta^{\star}(u_M)\right). \tag{2.30}$$

The Kernel technique to calculate the radius is explored below.

2.6.3 Kernel Technique Used for Finding Distance and Radius in Assigning Intuitionistic Fuzzy Score

Theorem 1. *Let the kernel function be $K(u_i, u_j)$. Then, the inner product distance is given by:*

$\|\phi(u_i) - \phi(u_j)\| = \sqrt{K(u_i, u_i) - 2K(u_i, u_j) + K(u_j, u_j)}$.

Proof. We begin by considering the Euclidean distance between the mapped points $\phi(u_i)$ and $\phi(u_j)$ in the feature space. By definition, the squared distance can be expressed using the inner product as

$$\|\phi(u_i) - \phi(u_j)\| = \sqrt{(\phi(u_i) - \phi(u_j))^\top (\phi(u_i) - \phi(u_j))}.$$

Expanding the inner product, we obtain

$$\|\phi(u_i) - \phi(u_j)\| = \sqrt{\phi(u_i)^\top \phi(u_i) - 2\phi(u_i)^\top \phi(u_j) + \phi(u_j)^\top \phi(u_j)}.$$

Now, using the definition of the kernel function $K(u_i, u_j) = \phi(u_i)^\top \phi(u_j)$, the above expression can be rewritten entirely in terms of kernel evaluations as

$$\|\phi(u_i) - \phi(u_j)\| = \sqrt{K(u_i, u_i) - 2K(u_i, u_j) + K(u_j, u_j)}.$$

This completes the proof. □

Theorem 2. *The Euclidean distance between the samples and the corresponding class center is represented by:*

$\|\phi(u_i) - C^+\| = \sqrt{K(u_i, u_i) - \frac{2}{m_1} \sum_{t_j=+1} K(u_i, u_j) + \frac{1}{m_1^2} \sum_{t_u=+1} \sum_{t_j=+1} K(u_i, u_j)}$,

$\|\phi(u_i) - C^-\| = \sqrt{K(u_i, u_i) - \frac{2}{m_2} \sum_{t_j=-1} K(u_i, u_j) + \frac{1}{m_2^2} \sum_{t_u=-1} \sum_{t_j=-1} K(u_i, u_j)}$,

where m_1 and m_2 denote the number of samples in the $+1$ and -1 classes, respectively, while C^+ and C^- correspond to the class centers of the positive and negative classes.

Proof. We compute the distance between the mapped sample $\phi(u_i)$ and the class centroid

C^+ in the feature space. By definition, the centroid of the positive class is given by

$$C^+ = \frac{1}{m_1} \sum_{t_j=+1} \phi(u_j).$$

The Euclidean distance can be written as

$$\|\phi(u_i) - C^+\| = \sqrt{(\phi(u_i) - C^+)^\top (\phi(u_i) - C^+)}.$$

Expanding the inner product, we obtain

$$\|\phi(u_i) - C^+\| = \sqrt{\phi(u_i)^\top \phi(u_i) + (C^+)^\top C^+ - 2\,\phi(u_i)^\top C^+}.$$

Using the kernel function definition $K(u_i, u_j) = \phi(u_i)^\top \phi(u_j)$, we now express each term in kernel form. The first term becomes

$$\phi(u_i)^\top \phi(u_i) = K(u_i, u_i).$$

The second term can be expanded as

$$\begin{aligned}(C^+)^\top C^+ &= \left(\frac{1}{m_1} \sum_{t_p=+1} \phi(u_p) \right)^\top \left(\frac{1}{m_1} \sum_{t_q=+1} \phi(u_q) \right) \\ &= \frac{1}{m_1^2} \sum_{t_p=+1} \sum_{t_q=+1} K(u_p, u_q).\end{aligned}$$

Similarly, the third term becomes

$$\phi(u_i)^\top C^+ = \frac{1}{m_1} \sum_{t_j=+1} K(u_i, u_j).$$

Substituting all terms back, we obtain

$$\|\phi(u_i) - C^+\| = \sqrt{K(u_i, u_i) + \frac{1}{m_1^2} \sum_{t_p=+1} \sum_{t_q=+1} K(u_p, u_q) - \frac{2}{m_1} \sum_{t_j=+1} K(u_i, u_j)}.$$

Similarly, the distance $\|\phi(u_i) - C^-\|$ can be derived for the negative class centroid. □

2.7 TSK Fuzzy System

The TSK fuzzy system [75] is an effective method for modeling complicated systems under uncertainty. TSK may approximate non-linear connections with transparency and interpretability by integrating fuzzy sets, fuzzy rules, and local linear models. Its versatility and usefulness make it a popular option in a wide range of real-world applications [149, 150, 151], making it a vital tool in fuzzy logic and artificial intelligence. The TSK is based on a set of IF-THEN fuzzy rules and is typically defined as:

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

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

$$\mathcal{T}_{rj} = \prod_{s=1}^{L} \Theta_{js}(u_{rs}), \tag{2.31}$$

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

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

$$\Gamma = \frac{\sum_{j=1}^{J} \mathcal{T}_{rj} y_{rj}}{\sum_{j=1}^{J} \mathcal{T}_{rj}} = \frac{\sum_{j=1}^{J} \prod_{s=1}^{L} \Theta_{js}(u_{rs}) \xi_j(u_{r1}, \ldots, u_{rL})}{\sum_{j=1}^{J} \prod_{s=1}^{L} \Theta_{js}(u_{rs})}. \tag{2.32}$$

2.8 Granular Ball Computation [4]

In 1997, Lin and Zadeh [4] introduced the concept of "granular computing" with the goal of minimizing the quantity of training data points. This approach aims to capture the essence of data simplification while preserving representativeness during the learning process. Consider a granular ball (GB) encompassing q data points, i.e., $\{u_1, u_2, \ldots, u_q\}$, where each u_j belongs to the vector space $\mathbb{R}^{1\times L}$. The center o of a GB is defined as the center of gravity calculated from all sample points within the ball. Mathematically, they can be calculated as: $o = \frac{1}{q}\sum_{j=1}^{q} u_j$.

The class/label assigned to a GB is based on the labels of the data points with the highest occurrence of data samples within the ball. To quantitatively evaluate the degree of division within a GB, the notion of "threshold purity" is introduced [43]. This threshold purity signifies the proportion of the predominant samples sharing the same label within the GB. The optimization problem governing the generation of GBs on set T is expressed as follows:

$$\begin{aligned} &\min \ \gamma_1 \times \frac{M}{\sum_{j=1}^{k} |GB_j|} + \gamma_2 \times k, \\ &\text{s.t. } \ purity(GB_j) \geq \rho, \ \ j = 1, 2, \ldots, k, \end{aligned} \tag{2.33}$$

where γ_1 and γ_2 are weight coefficients. ρ is the threshold purity and $|\cdot|$ represents the cardinality of a GB. The entire dataset is treated as a unified GB at the outset. When

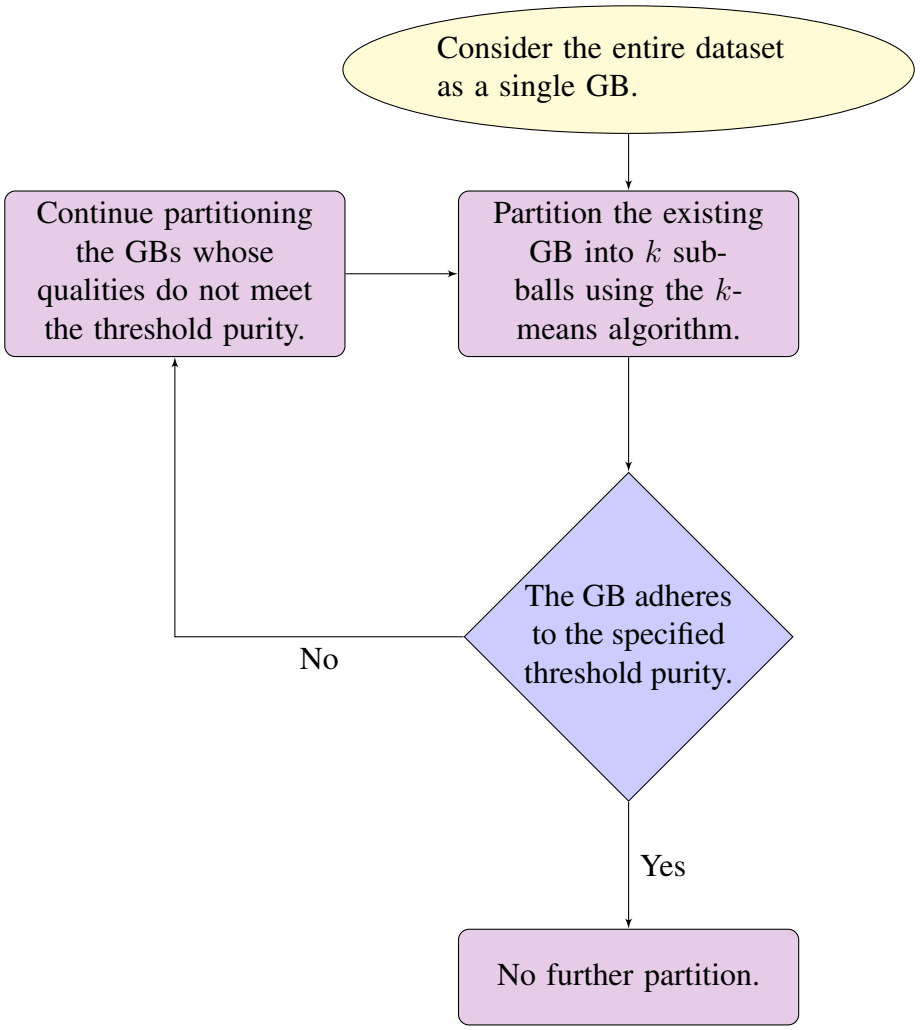
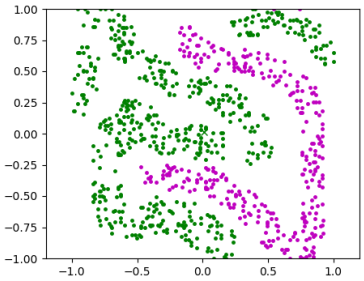


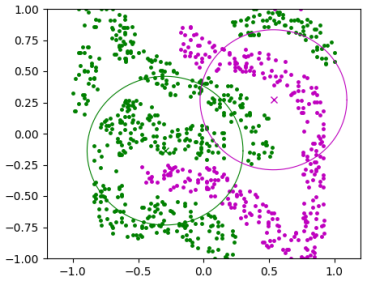
Figure 2.5: Granular ball generation process.

the purity of the GB falls below the given threshold, it must be divided several times until all sub-GBs reach or exceed the threshold purity. As the GBs' purity improves, so does their alignment with the original dataset's data distribution. Figure 2.5 demonstrates the technique for generating GBs.

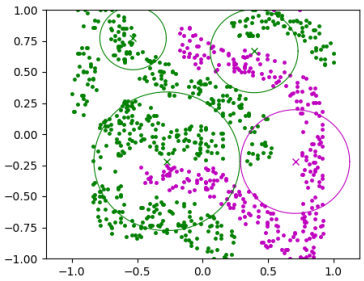
In Fig. 2.6, the GB generation and splitting process using the "fourclass" dataset is demonstrated. Initially, in Fig. 2.6a, the dataset consists of two non-linearly separable classes, with green points labelled as "$+1$" and magenta points labelled as "-1". In the first iteration, shown in Fig. 2.6b, large GBs are formed, loosely covering the green and magenta data points, providing a rough approximation of the data structure. As the process



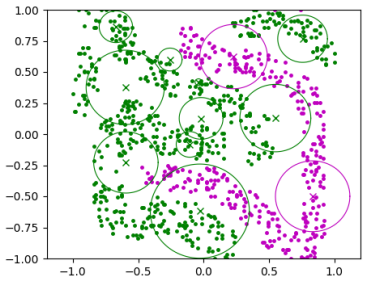
(a) The original dataset.



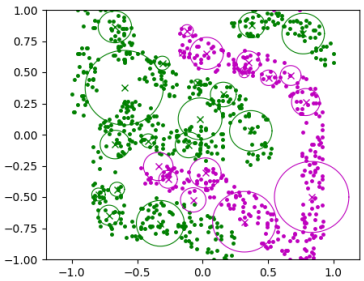
(b) Generated granular balls in the first iteration



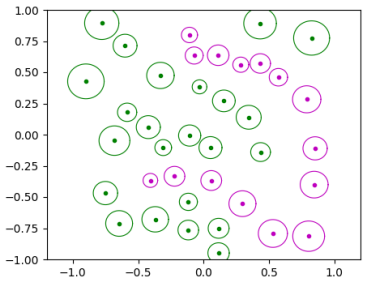
(c) Generated granular balls in the second iteration



(d) Generated granular balls in the middle iteration



(e) Results after stop splitting



(f) Extracted granular balls

Figure 2.6: The visualization of generating granular balls by splitting the "fourclass" dataset. It assigns the label "$+1$" to the green granular balls, while the label "-1" is used for the magenta granular balls.

continues in Fig. 2.6c and Fig. 2.6d, the GBs are progressively refined through iterative splitting, becoming smaller and more localized, capturing finer details of the dataset. By the final stages, shown in Fig. 2.6e and Fig. 2.6f, the GBs effectively encapsulate the underlying structure of the two classes, creating well-defined boundaries that can be used for classification purposes.

2.9 Graph Embedding (GE) [5]

The GE process [5] is designed to retain crucial graph structural details within the projection space. In the GE framework, considering an input dataset denoted as U, two components are defined: the intrinsic graph $\mathcal{U}^{int} = \{U, \Theta^{int}\}$ and the penalty graph $\mathcal{U}^{pen} = \{U, \Theta^{pen}\}$. The similarity weight matrix $\Theta^{int} \in \mathbb{R}^{M \times M}$ incorporates the graph weights corresponding to the intrinsic connections among vertices in U. Additionally, each element of the $\Theta^{pen} \in \mathbb{R}^{M \times M}$ represents the penalty matrix of U, which accounts for specific relationships among the graph's vertices. The optimization problem for GE is formulated as follows:

$$\begin{aligned} \hat{y} &= \underset{Tr({y_0}^t U^t \mathcal{S} U y_0)=a}{\operatorname{argmin}} \sum_{k \neq l} \left\| {y_0}^t u_k - {y_0}^t u_l \right\|_2^2 \Theta_{kl}^{int}, \\ &= \underset{Tr({y_0}^t U^t \mathcal{S} U y_0)=a}{\operatorname{argmin}} Tr({y_0}^t U^t \mathcal{L} U y_0). \end{aligned} \tag{2.34}$$

Here, y_0 represents the projection matrix, and the trace operator is denoted as $Tr(\cdot)$.

For $\mathcal{U}^{int}$, the Laplacian matrix is represented by $\mathcal{L} = \mathcal{F} - \Theta^{int} \in \mathbb{R}^{M \times M}$, where $\mathcal{F}$ is the diagonal matrix, whose diagonal elements are defined as $\mathcal{F}_{kk} = \sum_l \Theta_{kl}^{int}$. $\mathcal{S} = \mathcal{L}^{pen} = \mathcal{F}^{pen} - \Theta^{pen}$ is the Laplacian matrix of $\mathcal{U}^{pen}$, and a is a scalar term. The Eq. (2.34) reduces

to a generalized eigenvalue problem [152] as follows:

$$U_{int}s = \lambda U_{pen}s, \tag{2.35}$$

here $U_{int} = U^T \mathcal{L} U$ and $U_{pen} = U^T \mathcal{S} U$. This indicates that the eigenvectors of the matrix $U_{uni} = U_{pen}^{-1} U_{int}$ will be used to construct the transformation matrix. The unified matrix U_{uni} captures the inherent relationships among data samples via intrinsic and penalty graph connections.

2.10 Performance Metrics

To comprehensively evaluate the effectiveness of the proposed models in comparison with the baseline methods, several widely used classification performance metrics are considered. These metrics include Accuracy, Sensitivity, Specificity, and Precision. Each metric captures a different aspect of the classifier's predictive capability and provides a more complete assessment of model performance.

Accuracy measures the overall proportion of correctly classified samples among all predictions. Sensitivity, also known as recall or true positive rate, evaluates the ability of the model to correctly identify positive samples. Specificity measures the model's capability to correctly identify negative samples, while Precision quantifies the proportion of correctly predicted positive samples among all predicted positives.

The mathematical expressions used to compute these metrics are given as follows:

$$\text{Accuracy} = \frac{True_+ + True_-}{True_+ + False_+ + True_- + False_-}, \tag{2.36}$$

$$\text{Sensitivity} = \frac{True_+}{True_+ + False_-}, \tag{2.37}$$

$$\text{Specificity} = \frac{True_-}{True_- + False_+}, \tag{2.38}$$

$$\text{Precision} = \frac{True_+}{True_+ + False_+}. \tag{2.39}$$

In the above expressions, $True_+$ and $True_-$ denote the number of correctly predicted positive and negative samples, respectively. Similarly, $False_+$ represents the number of negative samples incorrectly classified as positive (false positives), while $False_-$ denotes the number of positive samples incorrectly classified as negative (false negatives). These metrics collectively provide a comprehensive evaluation of the classification performance of the proposed models.

Further, the AUC represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different decision thresholds. In discrete space, the AUC can be computed using the trapezoidal rule as

$$\text{AUC} = \sum_{i=1}^{n-1} \frac{TPR_{i+1} + TPR_i}{2} (FPR_{i+1} - FPR_i), \tag{2.40}$$

where TPR_i and FPR_i denote the true positive rate and false positive rate at the i-th threshold, respectively. The quantities TPR and FPR are defined as

$$TPR = \frac{True_+}{True_+ + False_-}, \tag{2.41}$$

$$FPR = \frac{False_+}{False_+ + True_-}. \tag{2.42}$$

2.11 Statistical Tests for Comparing Multiple Models

In machine learning research, evaluating models solely based on average accuracy may be misleading, as it can mask variations in performance across different datasets. To address this issue, Demšar [153] recommend the use of non-parametric statistical tests for comparing multiple learning algorithms across multiple datasets. These tests provide a principled way to determine whether observed differences in performance are statistically significant rather than occurring due to random variations. Commonly used statistical procedures include the ranking scheme, the Friedman test, the Nemenyi post hoc test, and the win–tie–loss sign test.

2.11.1 Ranking Scheme

In the ranking approach, models are ranked based on their performance on each dataset. For a given dataset, the best-performing model receives rank 1, the second-best receives rank 2, and so on, while poorer-performing models obtain higher rank values. Suppose $\mathscr{M}$ models are evaluated across $\mathscr{D}$ datasets. Let $\rho(m, d)$ denote the rank of the mth model on the dth dataset. The average rank of the mth model across all datasets is computed as

$$\rho(m, *) = \frac{1}{\mathscr{D}} \sum_{d=1}^{\mathscr{D}} \rho(m, d). \tag{2.43}$$

The model with the lowest average rank is considered the best-performing model overall.

2.11.2 Friedman Test

The Friedman test [154] is a non-parametric statistical test used to determine whether significant differences exist among multiple models based on their rankings across datasets. The test statistic follows a chi-square distribution with $\mathscr{M} - 1$ degrees of freedom and is

defined as

$$\chi_F^2 = \frac{12\mathscr{D}}{\mathscr{M}(\mathscr{M}+1)} \left(\sum_{m=1}^{\mathscr{M}} (\rho(m,*))^2 - \frac{\mathscr{M}(\mathscr{M}+1)^2}{4} \right). \tag{2.44}$$

However, it has been shown by Iman and Davenport [155] that the Friedman statistic may be conservative. To address this issue, an improved statistic, known as the F_F statistic, is used:

$$F_F = \chi_F^2 \left(\frac{\mathscr{D}-1}{\mathscr{D}(\mathscr{M}-1) - \chi_F^2} \right). \tag{2.45}$$

The F_F statistic follows an F-distribution with $(\mathscr{M}-1)$ and $(\mathscr{D}-1)(\mathscr{M}-1)$ degrees of freedom. If the null hypothesis (that all models perform equivalently) is rejected, further post hoc analysis is conducted.

2.11.3 Nemenyi Post Hoc Test

When the Friedman test indicates significant differences among models, the Nemenyi post hoc test [153] is applied to perform pairwise comparisons between models. According to this test, the performance of two models is considered significantly different if the difference between their average ranks exceeds the critical difference (C.D.), defined as

$$C.D. = q_\alpha \sqrt{\frac{\mathscr{M}(\mathscr{M}+1)}{6\mathscr{D}}}, \tag{2.46}$$

where q_α is the critical value obtained from the Studentized range distribution corresponding to the chosen significance level α. If the difference in average ranks between two models exceeds the critical difference, their performance is considered statistically different.

2.11.4 Win–Tie–Loss Sign Test

The win–tie–loss sign test [153] is a widely used non-parametric statistical procedure for comparing the performance of two classification models across multiple datasets. This

test evaluates pairwise differences between models by counting the number of datasets on which one model outperforms the other.

For a given pair of models evaluated on $\mathscr{D}$ datasets, three possible outcomes are considered for each dataset: a *win* when the first model performs better than the second model, a *loss* when it performs worse, and a *tie* when both models achieve identical performance. Under the null hypothesis, both models are assumed to perform equally well, implying that each model is expected to win approximately $\mathscr{D}/2$ datasets.

To determine whether the difference in performance between two models is statistically significant, the number of wins must exceed a threshold derived from the normal approximation of the binomial distribution. Specifically, two models are considered significantly different if the number of wins exceeds

$$\frac{\mathscr{D}}{2} + \frac{1.96\sqrt{\mathscr{D}}}{2}. \tag{2.47}$$

When ties occur, they are distributed equally between the two models if the number of ties is even. In cases where the number of ties is odd, one tie is discarded and the remaining ties are evenly divided between the models. The win–tie–loss sign test thus provides a simple and intuitive mechanism for assessing pairwise statistical differences between competing machine learning models across multiple datasets.

2.11.5 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test [153] is a non-parametric statistical test commonly used for pairwise comparison of two classifiers across multiple datasets. Unlike parametric tests, it does not assume that the performance differences follow a normal distribution, making it suitable for evaluating machine learning models where such assumptions may not hold.

Consider two models evaluated on $\mathscr{D}$ datasets. For each dataset d, the difference in

performance between the two models is computed. The absolute values of these differences are then ranked in ascending order while ignoring zero differences. Let R_d denote the rank assigned to the absolute difference corresponding to the dth dataset. Each rank is assigned a sign depending on whether the difference is positive or negative.

Let R^+ denote the sum of ranks corresponding to positive differences and R^- denote the sum of ranks corresponding to negative differences. The Wilcoxon test statistic is defined as:

$$W = \min(R^+, R^-). \tag{2.48}$$

Under the null hypothesis, both models are assumed to perform equivalently, implying that the positive and negative rank sums should be approximately equal. A significantly small value of the test statistic indicates that one model consistently outperforms the other across datasets. For larger sample sizes, the Wilcoxon statistic can be approximated by a normal distribution with mean and variance given by

$$\mu_W = \frac{\mathscr{D}(\mathscr{D}+1)}{4}, \text{ and } \sigma_W^2 = \frac{\mathscr{D}(\mathscr{D}+1)(2\mathscr{D}+1)}{24}, \text{ respectively.} \tag{2.49}$$

Based on this approximation, a p-value can be computed to determine whether the difference between the two models is statistically significant at a chosen significance level.

Chapter 3

Interpretable Randomized Neural Networks

Outcome of this Chapter

Publication: **M. Sajid**, A. K. Malik, M. Tanveer and P. N. Suganthan (2024). "Neuro-Fuzzy Random Vector Functional Link Neural Network for Classification and Regression Problems," in ***IEEE Transactions on Fuzzy Systems***, v. 32, no. 5, pp. 2738 - 2749, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3359652 **[SCI Indexed with Impact Factor.:** 10.2, Q1]

Source Code: https://github.com/mtanveer1/NeuroFuzzy-RVFL

RVFL neural networks provide efficient learning with fast training and good generalization. However, their decision-making process is difficult to interpret, which limits their applicability in scenarios where transparency is important. To address this limitation, this chapter proposes an interpretable Neuro-Fuzzy Random Vector Functional Link (NF-RVFL) network that integrates RVFL with a neuro-fuzzy system. The proposed model introduces a fuzzy layer that transforms input features into interpretable fuzzy representations through an IF–THEN rule mechanism, enabling human-understandable reasoning. The fuzzified features are further processed through random projections in the hidden layer and subsequently defuzzified to generate crisp representations. The final prediction is obtained by combining the original input features, hidden layer outputs, and defuzzified values. To construct the fuzzy layer, three clustering strategies are employed to determine fuzzy centers: randomly initialized centers (R-means), K-means clustering, and fuzzy C-means clustering, resulting in three model variants: NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C. The interpretability of the proposed models is further analyzed using partial dependence plots (PDP) and feature-based importance measures.

3.1 Background

ANNs, despite their numerous advantages, also exhibit several limitations. (i) The gradient descent (GD) algorithm, commonly used to optimize network parameters, iteratively updates weights and biases by minimizing the difference between predicted and expected outputs. However, this process can be slow and may converge to local optima instead of the global optimum. Moreover, the learning rate and initialization of parameters significantly influence the convergence and overall training performance. (ii) Training conventional neural networks on large-scale datasets is often computationally expensive and time-consuming, requiring substantial computational resources. (iii) Another major limitation is

their "*black-box*" nature, which makes it difficult to interpret and understand the reasoning behind their predictions. This lack of transparency raises concerns in critical domains such as healthcare and finance, where interpretability and trust are essential.

Despite the above-mentioned limitations, ongoing research is continually improving the efficiency, interpretability, and robustness of ANNs. Randomized neural networks (RdNNs) [1, 19] were introduced as an alternative to GD-based neural networks to overcome their drawbacks. In general, a touch of randomness is inherent in the RdNNs model's structure or learning process. This is achieved by holding certain parameters fixed during training and calculating the output layer's parameters using iterative process or closed-form solution [156, 157]. The randomness grants RdNNs the ability to learn with fewer adjustable settings, in less time, and without the need for high-end hardware. Random vector functional link (RVFL) neural network [1, 25] is one of the well-established RdNNs. In RVFL, the hidden layer's parameters (weights and biases) are initially set randomly, chosen from a suitable continuous probability distribution. Throughout the training phase, these parameters remain constant. RVFL stands out from other RdNNs due to the direct links between the input and output layers. The direct links characteristic has been observed to significantly enhance the learning performance of RVFL by acting as an implicit regularization for the randomization process [23, 158]. RVFL employs the Pseudo-inverse or the least-squares approach to find the optimal output parameters (weights connecting the input layer to the output layer and hidden layer to the output layer), which offers a closed-form solution. Furthermore, RVFL has fast training speed and universal approximation capability [21, 159].

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

The fusion of human-like reasoning, represented by IF-THEN fuzzy rules in a fuzzy system with an ANN's learning capabilities and connectivity, gives rise to a hybrid system known as a neuro-fuzzy model [160]. Neuro-fuzzy models use fuzzy logic to represent linguistic terms and rules, which are easily understandable to humans [160]. This linguistic interpretability allows experts and stakeholders to comprehend the decision-making process, making it easier to trust and validate the model's outputs. Neuro-fuzzy models have a transparent and explicit fuzzy rule base, which clearly shows the connections between the input and output variables. These rules can be interpreted and reviewed to gain insights into how the model arrives at its conclusions. Neuro-fuzzy models have garnered attention for their exceptional traits and wide-ranging applications in classification [161], regression [56], nonlinear system identification, control [162] and so on. Many advanced versions of neuro-fuzzy based models have been developed based on feature selection and rule extraction [163, 164]. Neuro-fuzzy models have notably contributed to the critical field of time series prediction [165], benefiting various domains such as finance [166], fault diagnosis in self-organizing cellular network [167], weather forecasting [168] and so on.

Although Neuro-fuzzy models have many advantages, however, designing neuro-fuzzy models presents several challenges. One such challenge is determining the appropriate architecture, including the number of fuzzy rules and neural network layers. Careful tuning is necessary to optimize both the neural network parameters and fuzzy logic rules, as improper choices may lead to suboptimal performance or difficulties in effective model training [169]. While neural networks can assist in extracting suitable fuzzy rules without human intervention, it remains essential to reduce the number of fuzzy rules without compromising accuracy, especially when dealing with high-dimensional inputs. Additionally, the training process for neuro-fuzzy models using traditional iterative methods like GD can be computationally expensive and time-consuming, especially for large datasets. The challenge further

intensifies as the training process involves optimizing non-convex loss functions, making it difficult to find the global minimum and potentially resulting in suboptimal solutions. To mitigate these issues, careful selection of optimization algorithms becomes essential for achieving better results in neuro-fuzzy model.

In this chapter, we propose the Neuro-fuzzy RVFL (NF-RVFL) model to deal with the aforementioned pitfalls of the RVFL and neuro-fuzzy models. The integration of RVFL with neuro-fuzzy systems presents a win-win situation for both RVFL and neuro-fuzzy networks. A key advantage of neuro-fuzzy models lies in their universal approximation capability and ability to adapt the interpretable IF-THEN rules to the training data [170]. The RVFL has already demonstrated its potential to overcome the limitations of traditional ANNs, and by combining it with neuro-fuzzy systems, the integrated NF-RVFL model can address the lack of interpretability inherent in RVFL. Moreover, RVFL possesses lesser number of unknown parameters in contrast to conventional neuro-fuzzy models. Consequently, the NF-RVFL model, being developed upon the RVFL framework, also inherits the characteristic of less unknown parameters. As a result, the proposed NF-RVFL model is expected to mitigate the memory-intensive characteristics often associated with traditional neuro-fuzzy models and make the proposed NF-RVFL model capable of handling large datasets with better generalization performance. Additionally, the convex quadratic optimization problem of RVFL overcomes the dilemma of non-convex optimization problem faced by traditional neuro-fuzzy models and produces more optimal solutions. The NF-RVFL model promises to be a versatile and effective solution, with good generalization performance and better interpretable results than the standard RVFL model.

Within the framework of NF-RVFL, input samples traverse the fuzzy layer employing the fuzzification technique. Subsequently, the fuzzified samples traverse a hidden layer via random projection, yielding defuzzified values via the defuzzification technique. These

defuzzified values, along with hidden layer outputs and the original inputs, collectively contribute to the output prediction process.

The following are the key highlights of this chapter:

[1] We propose the NF-RVFL model through the amalgamation of the RVFL neural network and the neuro-fuzzy systems. The proposed NF-RVFL model tries to address the lack of interpretability inherent in the original RVFL model.

[2] In the proposed NF-RVFL model, we use three distinct clustering methods to establish centers within the fuzzy layer: randomly initialized centers (referred to as R-means), K-means clustering centers, and Fuzzy C-means clustering centers. As a result, we obtain three different model variations, namely NF-RVFL-R, NF-RVFL-K and NF-RVFL-C, each producing a diverse set of fuzzified and defuzzified samples.

[3] The conducted experiments encompass a range of 29 binary, 28 multiclass, and 15 regression benchmark UCI datasets sourced from diverse domains with varying sizes. Statistical tests and analysis from these experiments illustrate that the proposed NF-RVFL models exhibit superior performance compared to baseline models.

[4] Numerous ablation analyses have been conducted, focusing on key elements integral to the NF-RVFL model. These include examinations of the direct link, Fuzzy layer, defuzzified layer, the number of fuzzy rules, and activation functions.

[5] A detailed justification for interpretability is achieved through the exploration of Partial Dependence Plots (PDP) [171] and feature importance-based interpretability in the proposed NF-RVFL model. The findings demonstrate the superior interpretability of the proposed NF-RVFL model compared to the standard RVFL model.

3.2 Proposed Method

In this section, we provide a platform for the neuro-fuzzy system (in our case, it is TSK) and RVFL to interact with each other, and we propose a hybrid NF-RVFL model that can make a logical decision based on the IF-THEN rule. In the proposed NF-RVFL's topological structure, input samples undergo the fuzzy layer using the technique of fuzzification. Afterward, the fuzzified samples traverse a hidden layer via random projection as well as produce defuzzified values using the defuzzification technique. These defuzzified values, the hidden layer outputs, and the initial inputs all contribute to the overall output prediction process. We start with defining and fixing notations for this chapter. Figure 3.1 depicts the structure of the proposed NF-RVFL model.

3.2.1 NF-RVFL

Following the discussion on the Takagi–Sugeno–Kang based neuro-fuzzy system in Section 2.7, let J denote the total number of fuzzy rules in the fuzzy layer. Let $\mathcal{F}_{js}$ be the fuzzy set with the membership function Θ_{js} of sth component of jth fuzzy rule of the fuzzy layer, where $s = 1, 2, \ldots, L$, $j = 1, 2, \ldots, J$.

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

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

$$y_{rj} = \xi_j(u_{r1}, u_{r2}, \ldots, u_{rL}) = \sum_{s=1}^{L} \alpha_{js} u_{rs}. \tag{3.1}$$

Here, α_{js} are the randomly generated coefficients, where $s = 1, 2, \ldots, L$ and $j = 1, 2, \ldots, J$. The fire strength of the jth fuzzy rule in the fuzzy layer is calculated using

Eq. (2.31). For each fuzzy rule, we take the weighted values of the fire strength as:

$$\omega_{rj} = \frac{\mathcal{T}_{rj}}{\sum_{j=1}^{J} \mathcal{T}_{rj}}. \tag{3.2}$$

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

$$\Theta_{js}\left(u_{rs}\right) = e^{-\left(\frac{u_{rs}-c_{js}}{\sigma_{js}}\right)^2}, \quad \text{for} \;\; s = 1, 2, \ldots, L. \tag{3.3}$$

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

[1] Randomly initialized centers (say R-means).

[2] Using K-means clustering.

[3] Using Fuzzy C-means clustering.

Noteworthy Remarks:

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

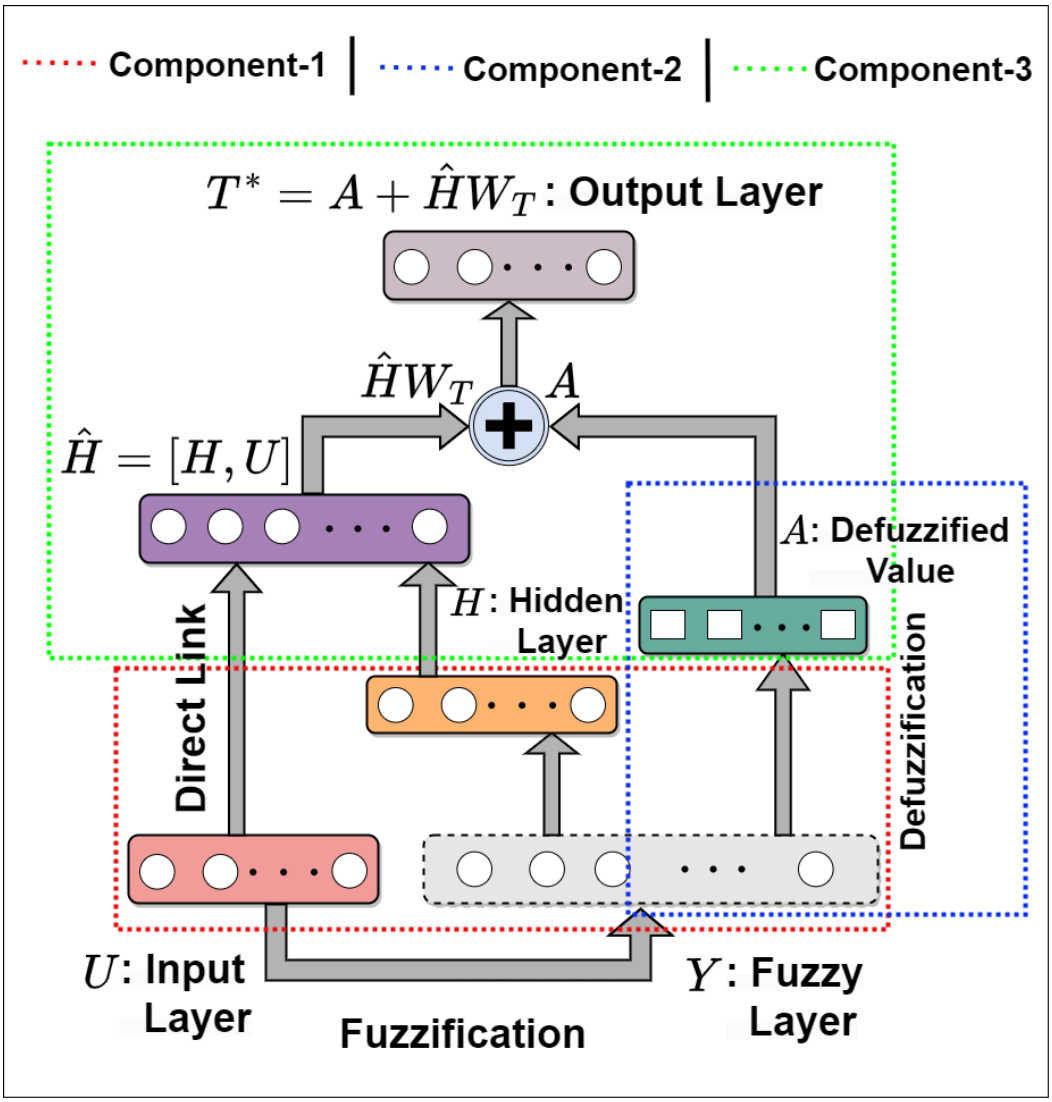


Figure 3.1: The architecture of the Neuro-fuzzy RVFL model.

(centers)) needed in the fuzzy layer.

The topology of the proposed NF-RVFL model is divided into three components. In the first component, fuzzification of the input vectors takes place by projecting the input layer into the fuzzy layer. Then, the fuzzy layer traverses the hidden layer of the RVFL through randomization. In the second component, the calculation of the output of the fuzzy layer takes place through the process of defuzzification. The output layer generation occurs in the third component, incorporating the available features, namely the original features, hid-

den layer features (generated through fuzzification), and the defuzzified features (generated through defuzzification). The idea behind the output layer generation involves projecting the original and hidden layer features using output layer weights, followed by translation using the defuzzified features.

Component-1: Fuzzification followed by non-linear transformation: The fuzzified vector of the rth training sample u_r is given as:

$$Y_r = (\omega_{r1}y_{r1}, \omega_{r2}y_{r2}, \ldots, \omega_{rJ}y_{rJ}), \ r = 1, 2, \ldots, M. \tag{3.4}$$

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

$$Y = \left(Y_1^t, Y_2^t, \ldots, Y_M^t\right)^t \in \mathbb{R}^{M \times J}. \tag{3.5}$$

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

$$H = \psi\left(YW_H + \beta_H\right) \in \mathbb{R}^{M \times h}, \tag{3.6}$$

where $\psi(\cdot)$ is the activation function.

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

fuzzy layer is defined as follows:

$$\begin{aligned}\mathcal{A}_r &= \left(\sum_{j=1}^{J}\omega_{rj}\lambda_{j1}y_{rj},\ldots,\sum_{j=1}^{J}\omega_{rj}\lambda_{jC}y_{rj}\right)\\ &= \left(\sum_{j=1}^{J}\omega_{rj}\lambda_{j1}\sum_{s=1}^{L}\alpha_{js}u_{rs},\ldots,\sum_{j=1}^{J}\omega_{rj}\lambda_{jC}\sum_{s=1}^{L}\alpha_{js}u_{rs}\right)\\ &= \Bigg(\omega_{r1}\sum_{s=1}^{L}\alpha_{1s}u_{rs}\lambda_{11}+\cdots+\omega_{rJ}\sum_{s=1}^{L}\alpha_{Js}u_{rs}\lambda_{J1},\end{aligned} \tag{3.7}$$

$$\begin{aligned}&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\ &\omega_{r1}\sum_{s=1}^{L}\alpha_{1s}u_{rs}\lambda_{1C}+\cdots+\omega_{rJ}\sum_{s=1}^{L}\alpha_{Js}u_{rs}\lambda_{JC}\Bigg)\\ &= \left(\omega_{r1}\sum_{s=1}^{L}\alpha_{1s}u_{rs},\ldots,\omega_{rJ}\sum_{s=1}^{L}\alpha_{Js}u_{rs}\right)\Lambda,\end{aligned} \tag{3.8}$$

where, $\Lambda = \begin{bmatrix}\lambda_{11} & \cdots & \lambda_{1C}\\ \vdots & \vdots & \vdots\\ \lambda_{J1} & \cdots & \lambda_{JC}\end{bmatrix}$, and λ_{jc} is the parameter used in THEN part of the Neuro-fuzzy system to defuzzify it ($j = 1, 2, \ldots, J$ and $c = 1, 2, \ldots, C$). The computation of the λ_{jc}'s is done in the component-3 using the closed form solution. Therefore, for the input matrix U, the defuzzified output matrix of the fuzzy layer is given as:

$$\mathcal{A} = \left(\mathcal{A}_1^t, \mathcal{A}_2^t, \ldots, \mathcal{A}_M^t\right)^t \in \mathbb{R}^{M\times C}, \tag{3.9}$$

$$\mathcal{A} = \begin{bmatrix}\left(\omega_{11}\sum_{s=1}^{L}\alpha_{1s}u_{1s},\cdots,\omega_{1J}\sum_{s=1}^{L}\alpha_{Js}u_{1s}\right)\Lambda\\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots\\ \left(\omega_{M1}\sum_{s=1}^{L}\alpha_{1s}u_{Ms},\cdots,\omega_{MJ}\sum_{s=1}^{L}\alpha_{Js}u_{Ms}\right)\Lambda\end{bmatrix}$$

$$\Rightarrow \mathcal{A} = \left(\Omega \odot \begin{bmatrix} \sum_{s=1}^{L} \alpha_{1s} u_{1s} & \cdots & \sum_{s=1}^{L} \alpha_{Js} u_{1s} \\ \vdots & \vdots & \vdots \\ \sum_{s=1}^{L} \alpha_{1s} u_{Ms} & \cdots & \sum_{s=1}^{L} \alpha_{Js} u_{Ms} \end{bmatrix} \right) \Lambda$$

$$\Rightarrow \mathcal{A} = \left(\Omega \odot \left(U\alpha^t \right) \right) \Lambda. \tag{3.10}$$

Where $\odot$ represents the componentwise matrix multiplication,

$$\alpha^t = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{J1} \\ \vdots & \vdots & \vdots \\ \alpha_{1L} & \cdots & \alpha_{JL} \end{bmatrix} \text{ and } \Omega = \begin{bmatrix} \omega_{11} & \cdots & \omega_{1J} \\ \vdots & \vdots & \vdots \\ \omega_{M1} & \cdots & \omega_{MJ} \end{bmatrix}. \tag{3.11}$$

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

$$\text{Let } \hat{H} = [H, U] \tag{3.12}$$

$$\text{then } T^* = \mathcal{A} + \hat{H} W_T \tag{3.13}$$

represents the anticipated output matrix of the proposed model. Eq. (3.13) is rewritten as:

$$\begin{aligned} T^* &= \left(\Omega \odot \left(U\alpha^t \right) \right) \Lambda + \hat{H} W_T \\ &= \left(\left(\Omega \odot \left(U\alpha^t \right) \right), \hat{H} \right) \begin{bmatrix} \Lambda \\ W_T \end{bmatrix} \\ &= \mathcal{D} W, \end{aligned} \tag{3.14}$$

$$\text{where } \mathcal{D} = \left(\left(\Omega \odot \left(U\alpha^t \right) \right), \hat{H} \right) \in \mathbb{R}^{M\times(L+J+h)} \tag{3.15}$$

$$\text{and } W = \begin{bmatrix} \Lambda \\ W_T \end{bmatrix} \in \mathbb{R}^{(L+J+h)\times C} \tag{3.16}$$

is the matrix of the knowns and the unknowns of the proposed NF-RVFL model, respectively. Therefore, we need to find the W only for the proposed model. The resultant optimization problem of Eq. (3.14) is formulated as:

$$(W)_{\min} = \underset{W}{\operatorname{argmin}} \frac{\mathfrak{C}}{2}\|\mathcal{D}W - T\|_2^2 + \frac{1}{2}\|W\|_2^2, \tag{3.17}$$

where $\mathfrak{C}$ is a regularization parameter.

The weight matrix W is computed as follows:

$$(W)_{\min} = \begin{cases} \left(\mathcal{D}^t\mathcal{D} + \frac{1}{\mathfrak{C}}I\right)^{-1}\mathcal{D}^tT, & (L+J+h) \leq M, \\ \mathcal{D}^t\left(\mathcal{D}\mathcal{D}^t + \frac{1}{\mathfrak{C}}I\right)^{-1}T, & M < (L+J+h), \end{cases} \tag{3.18}$$

where I is an identity matrix of conformal dimension.

3.2.2 Time Complexity of the Proposed NF-RVFL Models

The time complexity of the proposed NF-RVFL models depends upon three components: (a) clustering technique, (b) Fuzzy layer and hidden layer, and (c) $(W)_{\min}$ calculated on (3.18). Following [172], the time complexity of K-means and Fuzzy C-means are $\mathcal{O}(MLJv)$ and $\mathcal{O}(MLJ^2v)$, respectively, where v is the number of iterations. The time complexity of R-means is $\mathcal{O}(J)$. The time complexity of the generation of the fuzzy and hidden layer is $\mathcal{O}(MJh)$. Time complexity in computing $(W)_{\min}$ is $\mathcal{O}(W)$, where $\mathcal{O}(W) = \mathcal{O}((L+J+h)^2M) + \mathcal{O}((L+J+h)^3)$ if $(L+J+h) \leq M$ or $\mathcal{O}(W) = \mathcal{O}(M^2(L+J+h)) + \mathcal{O}(M^3)$ if $M < (L+J+h)$. Therefore, the time complexity of the NF-RVFL-R is $\mathcal{O}(J) + \mathcal{O}(MJh) + \mathcal{O}(W) \approx \mathcal{O}(MJh) + \mathcal{O}(W)$, the time complexity of NF-RVFL-K is $\mathcal{O}(MLJv) + \mathcal{O}(MJh) + \mathcal{O}(W)$ and the time complexity of NF-RVFL-C

Algorithm 3.1 Algorithm to find the output layer weights of the proposed NF-RVFL model.

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

1: Let u_r be the rth training sample, where $r = 1, 2, \ldots, M$.
2: **Generate**: $\alpha \in \mathbb{R}^{J \times L}$ randomly.
3: **Calculate**: the centers of the jth fuzzy rule, *i.e.*, $c_j = (c_{j1}, c_{j2}, \ldots, c_{jL})$ for $j = 1, 2, \ldots, J$ of the fuzzy layer by employing R-means or K-means, or fuzzy C-means clustering technique.
4: **Calculate**: ω_{rj} using Eq. (3.2).
5: **Calculate**: Y_r using Eq. (3.4).
6: **Calculate**: Y using Eq. (3.5).
7: **Generate**: $W_H \in \mathbb{R}^{J \times h}$ and $\beta_H \in \mathbb{R}^{M \times h}$ randomly with all columns identical in β_H.
8: **Calculate**: H using Eq. (3.6).
9: **Construct**: $\Omega \in \mathbb{R}^{M \times J}$ using Eq. (3.11).
10: **Construct**: $\hat{H}$ using Eq. (3.12).
11: **Construct**: $\mathcal{D}$ using Eq. (3.15).
12: **Find**: the unknown matrix, output layer weights, W using Eq. (3.18).

Source code link: https://github.com/mtanveer1/NeuroFuzzy-RVFL.

is $\mathcal{O}(MLJ^2v) + \mathcal{O}(MJh) + \mathcal{O}(W)$.

3.3 Dataset, Experimental Setup and Hyperparameter Setting

In this section, we give a detailed description of the datasets used for the experiments, experimental setup and the hyperparameter ranges for each model including baseline and proposed.

3.3.1 Binary, Multiclass, and Regression Dataset Description

To assess the effectiveness of the proposed NF-RVFL models, comprehensive experiments are performed on publicly accessible UCI [7] datasets encompassing diverse domains. For binary classification, experiments are conducted over 29 datasets. Among these,

10 datasets comprise sample sizes spanning the range of 101 to 500 instances, 9 datasets exhibit sample sizes ranging from 501 to 1000 instances, and 10 datasets sample sizes extending from 1001 to 130064 instances. For the multiclass classification, 28 datasets are included in the assessment. Among these, 14 datasets are allocated to the sample size ranges of 101 to 1000 instances and 14 datasets whose sample size ranges from 1001 to 58000 instances, respectively. For the regression task, experiments are done over 15 regression datasets. Among these regression datasets, 3 datasets have sizes up to 1000 samples, 7 datasets have sample sizes ranging from 1001 to 10000, and another 5 datasets have sample sizes ranging from 10001 to 40768. The specific dataset sizes for the binary and multiclass are given in Table 3.1, and the regression dataset's sizes are outlined in the first column of the regression Table 3.7.

3.3.2 Experimental Setup and Hyperparameter Settings

The experimental procedures are executed on a computing system possessing MATLAB R2023a software, Intel(R) Xeon(R) Platinum 8260 CPU @ 2.30GHz, 2301 Mhz, 24 Core(s), 48 Logical Processor(s) with 256 GB RAM on a Windows-10 operating platform. We employ the 5-fold cross-validation approach and grid search to fine-tune model hyperparameters. We estimated the testing accuracy for each fold independently for each set of hyperparameters. The mean of these five fold accuracies was then used to get the average testing accuracy for each set of hyperparameters. The models' testing accuracy is determined by selecting the model with the highest average testing accuracy. Now we give the parameter selection range of the proposed NF-RVFL models and the compared baseline models. Each model's regularization parameters are selected from the set: $\mathfrak{C} = \{10^{-5}, 10^{-4}, \cdots, 10^{4}, 10^{5}\}$. The number of hidden layer nodes (h) is selected from the range: $h = 3 : 20 : 203$. For the proposed NF-RVFL model, the number of fuzzy rules

Table 3.1: Details of the datasets used in experiments for binary and multiclass classification sourced from the UCI [7] repository.

Binary			Multiclass			
Dataset	#Samples	#Features	Dataset	#Samples	#Features	#Class
adult	48842	14	abalone	4177	8	3
bank	4521	16	annealing	898	31	5
blood	748	4	cardiotocography_10clases	2126	21	10
breast_cancer	286	9	ecoli	336	7	8
breast_cancer_wisc_diag	569	30	energy_y1	768	8	3
chess_krvkp	3196	36	flags	194	28	8
congressional_voting	435	16	hayes_roth	160	3	3
connect_4	67557	42	heart_cleveland	303	13	5
credit_approval	690	15	led_display	1000	7	10
cylinder_bands	512	35	low_res_spect	531	100	9
haberman_survival	306	3	lymphography	148	18	4
heart_hungarian	294	12	molec_biol_splice	3190	60	3
horse_colic	368	25	nursery	12960	8	5
miniboone	130064	50	oocytes_merluccius_states_2f	1022	25	3
molec_biol_promoter	106	57	optical	5620	62	10
monks_1	556	6	seeds	210	7	3
monks_2	601	6	semeion	1593	256	10
monks_3	554	6	soybean	683	35	18
mushroom	8124	21	statlog_shuttle	58000	9	7
musk_1	476	166	statlog_vehicle	846	18	4
musk_2	6598	166	thyroid	7200	21	3
oocytes_trisopterus_nucleus_2f	912	25	wall_following	5456	24	4
ozone	2536	72	waveform	5000	21	3
parkinsons	195	22	waveform_noise	5000	40	3
pittsburg_bridges_T_OR_D	102	7	wine	178	13	3
spambase	4601	57	wine_quality_red	1599	11	6
statlog_german_credit	1000	24	wine_quality_white	4898	11	7
titanic	2201	3	zoo	101	16	7
vertebral_column_2clases	310	6				

in the fuzzy layer is chosen from the range: $J = 5 : 5 : 50$ and the standard deviation (σ_{js}) is taken 1. μ for the IFRVFL model is chosen from the set $\{2^{-5}, 2^{-3}, \ldots, 2^{5}\}$. λ for the GEELM-LDA, GEELM-LFDA, Total-Var-RVFL and MCVELM models ranges in the set: $\lambda = \{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. For BLS, the number of feature groups is chosen from the range $N_2 = 1 : 2 : 21$; the number of feature nodes in each feature group is chosen from the range $N_1 = 5 : 5 : 50$. For Fuzzy-BLS, the number of fuzzy groups is chosen from the range $N_{fg} = 1 : 2 : 21$; and the number of fuzzy nodes in each fuzzy group is chosen from the range $N_{fn} = 5 : 5 : 50$. The number of enhancement nodes for BLS, Fuzzy-BLS, is chosen from the range $N_3 = 5 : 5 : 50$. The activation function ψ is a critical component in the neural network architecture, playing an essential role in introducing non-linearity and enabling networks to model complex patterns. To keep this thing in mind, we also fine-tune the activation function. We take 5 activation functions, 1 denotes *Sigmoid*, 2 denotes *Sine*, 3 denotes *Tribas*, 4 denotes *Radbas*, 5 denotes *Tansig* and 6 denotes *Relu*.

3.4 Experiments and Results

We start discussing the details of the compared baseline models. Then, we discuss the experimental results and statistical analysis on binary, multiclass classification, and regression datasets. In the last subsection, the interpretability of the proposed NF-RVFL models is discussed, followed by ablation studies on the proposed NF-RVFL models, examining their dependency on key components such as direct link (connecting the input and output layers), fuzzification (fuzzy layer), defuzzification (defuzzified layer), number of fuzzy rules (clusters) and activation functions.

3.4.1 Compared Models

We compare the proposed NF-RVFL models to the other nine RdNNs. The enumerated RdNN models under consideration are as follows:

[1] RVFL: Standard shallow RVFL model [1].

[2] ELM: Extreme learning machine [22].

[3] IFRVFL: Intuitionistic fuzzy RVFL [32].

[4] GEELM-LDA: Graph embedded ELM with linear discriminant analysis (LDA) [173].

[5] GEELM-LFDA: Graph embedded ELM with Local Fisher discriminant analysis (LFDA) [173].

[6] Total-Var-RVFL: Total variance minimization based RVFL [174].

[7] MCVELM: Minimum class variance based ELM [175].

[8] BLS: Standard broad learning system [3].

[9] Fuzzy-BLS: BLS based on neuro-fuzzy system [56].

[10] NF-RVFL-R: The proposed Neuro-fuzzy RVFL based on R-means clustering.

[11] NF-RVFL-K: The proposed Neuro-fuzzy RVFL based on K-means clustering.

[12] NF-RVFL-C: The proposed Neuro-fuzzy RVFL based on fuzzy C-means clustering.

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

Table 3.2 presents experimental results for the proposed NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C models, along with baseline models, namely RVFL, ELM, IFRVFL,

Table 3.2: Classification accuracies of the proposed NF-RVFL-R, NF-RVFL-K and NF-RVFL-C models along with RVFL, ELM, IFRVFL, GEELM-LDA, GEELM-LFDA, Total-Var-RVFL, and MCVELM on binary UCI datasets.

Dataset ↓ | Model →	RVFL [1]	ELM [22]	IFRVFL [32]	GEELM-LDA [173]	GEELM-LFDA [173]	Total-Var-RVFL [174]	MCVELM [175]	NF-RVFL-R†	NF-RVFL-K†	NF-RVFL-C†
adult	84.0342	83.8725	83.5438	83.422	83.2136	84.0178	83.9011	83.2992	84.5879	**84.6218**
bank	89.4051	89.4934	89.1173	89.6158	46.4966	89.6705	89.6703	89.4269	89.4492	**90.9391**
blood	76.5065	76.9056	77.4398	67.1732	76.2398	76.9065	77.4065	77.838	**77.8398**	77.4398
breast_cancer	70.1956	70.1754	71.9298	**89.8246**	84.5796	70.1754	70.5263	72.3351	72.6921	70.8953
breast_cancer_wisc_diag	93.8503	92.2683	89.2719	93.1455	86.988	94.1997	93.4995	94.4753	94.3782	**95.6063**
chess_krvkp	72.0313	71.936	74.2589	76.7814	69.2351	77.8963	75.8278	82.6045	81.7908	**85.2322**
congressional_voting	63.6782	63.2184	58.8506	59.7701	54.2529	63.908	63.4483	63.908	63.926	**64.1379**
connect_4	75.4518	75.4059	75.3407	**76.7316**	75.4281	75.4992	75.4459	75.4844	75.5022	76.2156
credit_approval	85.5674	85.3623	86.5217	85.5072	85.6522	85.5072	85.5072	85.942	85.942	**87.8581**
cylinder_bands	66.4154	65.8081	66.8785	67.5405	67.5385	67.7727	66.0156	68.9606	69.724	**71.2941**
haberman_survival	73.4902	73.8181	75.1348	55.5632	52.2898	74.6398	73.4902	75.2396	75.1296	**75.4574**
heart_hungarian	73.8223	74.4886	76.8849	83.7288	**85.0847**	75.8738	74.1613	76.9024	76.5576	78.2466
horse_colic	86.0369	85.424	85.5979	75.8053	74.7501	86.4124	86.1533	**86.6938**	86.4272	86.6901
miniboone	81.4984	79.2741	84.3698	82.4076	83.3566	81.8659	81.2939	84.3969	84.0217	**89.9858**
molec_biol_promoter	72.7706	74.632	78.3983	71.8182	77.4892	79.5236	72.684	82.0346	**82.12**	82.0346
monks_1	83.7934	82.2685	84.6895	73.5811	71.5476	84.1538	84.8697	81.4527	82.3584	**86.6992**
monks_2	80.0138	81.6804	84.5898	60.7273	57.23	85.8485	82.8499	85.8485	87.3774	**88.8894**
monks_3	91.1564	90.7912	90.0704	82.8419	77.69	91.6986	91.8821	91.8436	91.6486	**92.1744**
mushroom	96.3931	96.0735	93.9684	96.3071	98.7561	97.2919	96.7382	94.2369	95.9132	**99.3351**
musk_1	72.0614	69.7675	71.864	72.4496	65.9518	72.4846	70.1776	75.2149	**76.0482**	74.1645
musk_2	84.5909	84.5909	84.5909	85.1573	84.5909	85.1818	84.909	83.9394	85.501	**85.9545**
oocytes_trisopterus_nucleus_2f	78.9419	77.5182	75.2219	78.0694	79.8889	79.7036	79.6025	79.3863	79.3779	**79.9345**
ozone	97.1217	97.1319	97.1612	97.1217	96.0615	97.1611	97.1358	**97.2005**	97.1612	97.1612
parkinsons	82.2569	80.5128	78.4615	84.1026	83.5897	82.0513	83.5897	83.0769	83.0769	**85.5856**
pittsburg_bridges_T_OR_D	87.1905	87.3685	89.3025	82.1429	80.1905	89.1429	90.1429	**90.1905**	88.2381	**90.1905**
spambase	88.546	87.0472	85.0883	88.9582	88.546	88.6116	87.5819	88.9674	88.9583	**89.3931**
statlog_german_credit	76.9	75.7	76.9	66.7	53.9	77	76.8	77.1	**77.7**	**77.7**
titanic	77.9168	77.9168	**79.0532**	65.5896	61.0231	78.6901	**79.0532**	78.4623	**79.0532**	**79.0532**
vertebral_column_2clases	71.2903	70.6452	85.4839	**94.8387**	92.2581	78.806	74.5161	79.0323	82.366	80.3226
Average Accuracy	80.4458	80.0378	81.0339	78.8766	75.6489	81.7826	80.9959	82.2584	82.5816	**83.5591**
Average Rank	6.931	7.9483	6.2069	6.3966	7.431	4.8103	5.8793	4.069	3.5172	**1.8103**
Average Standard Deviation	9.4944	10.2999	7.9884	11.0356	11.2942	9.116	8.7296	8.0372	**6.9893**	7.3014

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

Table 3.3: Statistical analysis of the proposed models (listed in the first column) with all the models (listed in the first row) for the binary UCI datasets based on the Nemenyi post hoc test.

	RVFL [1]	ELM [22]	IFRVFL [32]	GEELM-LDA [173]	GEELM-LFDA [173]	Total-Var-RVFL [174]	MCVELM [175]	NF-RVFL-R†	NF-RVFL-K†	NF-RVFL-C†
NF-RVFL-R†	✓	✓	0	0	✓	0	0	N/A	0	0
NF-RVFL-K†	✓	✓	✓	✓	✓	0	0	0	N/A	0
NF-RVFL-C†	✓	✓	✓	✓	✓	✓	✓	0	0	N/A

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

Table 3.4: Classification accuracies of the proposed NF-RVFL-R, NF-RVFL-K and NF-RVFL-C models along with the compared baseline models, *i.e.,* RVFL, ELM, IFRVFL, Total-Var-RVFL, MCVELM on multiclass UCI datasets.

Dataset ↓ | Model →	RVFL [1]	ELM [22]	Total-Var-RVFL [174]	MCVELM [175]	NF-RVFL-R†	NF-RVFL-K†	NF-RVFL-C†
abalone	63.4665	63.4419	63.754	63.6578	63.8253	**65.4586**	64.0887
annealing	89.6381	88.3018	89.1937	89.0381	86.6313	87.0782	**90.8591**
cardiotocography_10clases	66.2324	65.2894	66.1351	66.2307	64.8151	62.8425	**69.2395**
ecoli	60.9175	61.2116	51.0667	51.0667	60.619	60.6277	**62.5405**
energy_y1	88.6716	88.2777	**89.7106**	89.0604	87.979	88.5434	89.0595
flags	53.6302	51.5789	53.1174	54.197	**55.2687**	55.1957	55.2092
hayes_roth	61.875	60.625	63.5	61.875	61.875	64.075	**65**
heart_cleveland	59.7268	59.377	61.0328	59.3607	61.0383	**61.3388**	60.0437
led_display	72.6	72.5	73.5	73	72.7	72.2	**75.6**
low_res_spect	**87.7605**	86.6285	73.9852	74.1851	87.3849	87.5666	**87.7605**
lymphography	86.4138	85.4745	86.4138	86.4368	85.7701	87.1034	**88.5057**
molec_biol_splice	56.0949	51.8809	54.2633	51.8809	**69.2163**	68.652	68.3072
nursery	70.3935	70.1775	65.693	62.8627	66.7824	66.3889	**71.9444**
oocytes_merluccius_states_2f	91.2846	90.8948	91.8737	91.633	91.3845	91.1894	**92.067**
optical	95.4626	94.3594	95.4626	94.8754	93.3808	94.0214	**97.2317**
seeds	89.0476	87.1429	90	89.0476	**90.4762**	90.3653	**90.4762**
semeion	87.0673	85.4363	87.2562	83.0492	88.0726	**88.7725**	87.3187
soybean	87.6954	86.9809	88.4264	88.306	87.3991	88.1387	**90.4723**
statlog_shuttle	98.6483	98.6241	98.681	98.6793	98.6345	98.6638	**98.8764**
statlog_vehicle	82.0327	81.5593	82.0313	81.3227	79.1974	79.5475	**82.2666**
thyroid	95.625	95.4722	95.9917	95.5833	94.7617	94.7639	**96.0278**
wall_following	75.6057	75.239	75.734	75.1845	71.518	72.1223	**80.683**
waveform	86.54	86.08	86.56	86.26	86.88	86.68	**87.22**
waveform_noise	86.28	85.02	86.42	85.42	86.28	86.3	**86.58**
wine	97.2063	96.0952	96.6508	96.7896	97.2063	**97.2363**	97.2063
wine_quality_red	59.4769	59.4755	59.9134	60.4759	60.5106	**60.8533**	60.5388
wine_quality_white	52.3989	52.5332	52.472	52.4103	52.4106	52.6349	**53.4925**
zoo	95	96	97	96	97	95	**97.6**
Average Accuracy	78.4569	77.7028	77.7085	77.0675	78.5363	78.6914	**80.222**
Average Rank	4.1786	5.7143	3.6429	4.75	4.375	3.8036	**1.5357**
Average Standard Deviation	5.4311	7.0908	6.8053	8.7373	4.8882	5.1418	**4.5872**

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

GEELM-LDA, GEELM-LFDA, Total-Var-RVFL, and MCVELM. Among them, NF-RVFL-C achieves the highest average accuracy of 83.5591%, followed by NF-RVFL-K at 82.5816%, and NF-RVFL-R at 82.2584%, outperforming other state-of-the-art models. Notably, NF-RVFL-K and NF-RVFL-C exhibit the lowest standard deviations (6.9893 and 7.3014, respectively), indicating the high certainty of their predictions. Following the Statistical tests discussion from Section 2.11, Table 3.2 reports the average ranks of all models across the datasets. The proposed models achieve the lowest average ranks, with NF-RVFL-C (1.8103), NF-RVFL-K (3.5172), and NF-RVFL-R (4.069) securing the top three positions among all competing models, indicating their superior performance.

Table 3.5: Statistical analysis of the proposed (listed in the first column) with all the models (listed in the first row) for multiclass UCI datasets based on the Nemenyi post hoc test.

	RVFL [1]	ELM [22]	Total-var-RVFL [174]	MCVELM [175]	NF-RVFL-R†	NF-RVFL-K†	NF-RVFL-C†
NF-RVFL-R†	0	0	0	0	N/A	0	✗
NF-RVFL-K†	0	✓	0	0	0	N/A	✗
NF-RVFL-C†	✓	✓	✓	✓	✓	✓	N/A

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

The Friedman test is conducted to examine whether the differences among the models are statistically significant. For $\mathscr{M} = 10$ models and $\mathscr{D} = 29$ datasets, the obtained values are $\chi_F^2 = 105.3045$ and $F_F = 18.9378$. At the 5% significance level, the critical value is $F_F(9, 252) = 1.9171$. Since $18.9378 > 1.9171$, the null hypothesis is rejected, confirming that significant performance differences exist among the models.

Subsequently, the Nemenyi post hoc test is applied for pairwise comparisons, with the critical difference $(C.D.) = 2.5157$ at $\alpha = 0.05$. Table 3.3 shows that NF-RVFL-C is statistically superior to all baseline models. NF-RVFL-K significantly outperforms all models except Total-Var-RVFL and MCVELM. NF-RVFL-R shows statistically significant improvement over RVFL, ELM, and GEELM-LFDA, while its performance remains comparable with IFRVFL, GEELM-LDA, Total-Var-RVFL, and MCVELM. Overall, the lower average ranks indicate the stronger generalization capability of the proposed NF-RVFL models compared to the baseline methods.

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

Table 3.4 showcases the results related to multiclass classification datasets. The IFRVFL model is designed exclusively for binary classification, making it ineligible for inclusion in the multiclass comparisons. The GEELM-LDA and GEELM-LFDA are the worst-

performing models in binary classification as per the Subsection 3.4.2, so they are excluded from the multiclass classification comparison.

The proposed NF-RVFL-C, NF-RVFL-K, and NF-RVFL-R secured the first, second, and third spots with an average accuracy of 80.222%, 78.6914%, and 78.5363%, respectively. With a standard deviation of 4.5872, 4.8882, and 5.1418, the proposed NF-RVFL-C, NF-RVFL-R, and NF-RVFL-K, respectively, have the lowest standard deviations among all the models. This showcases that the proposed NF-RVFL models have the highest prediction capabilities with high certainty. Among all models, the proposed NF-RVFL-C has the lowest rank of 1.5357, while NF-RVFL-K ranks third, highlighting their superiority over baseline models. With an average rank of 4.375, NF-RVFL-R demonstrates strong competitiveness. We get $\chi_F^2 = 59.4811$ and $F_F = 14.7992$ for $\mathscr{M} = 7$ and $\mathscr{D} = 28$. Also, $F_F(6, 252) = 2.1549$ at a 5% significance level. $F_F = 14.7992 > F_F(6, 252)$ demonstrates substantial differences across the compared models; therefore, the null hypothesis is rejected. Table 3.5 presents Nemenyi post hoc test results for all proposed and baseline models. The proposed NF-RVFL-C is statistically superior to all existing models, as well as the proposed NF-RVFL-R and NF-RVFL-K models. NF-RVFL-K is statistically better than ELM. MCVELM and standard ELM get the last and second last spots in the ranking scheme, highlighting the RVFL-based family's superior generalization performance. Statistical tests confirm the proposed NF-RVFL models' top performance, attributed to enhanced generalization performance via Neuro-fuzzy system integration.

3.4.4 Comparision of RVFL and BLS-based Neuro-Fuzzy Models

Now, we conduct a comparative analysis of two neuro-fuzzy-based models: Fuzzy-BLS and our proposed NF-RVFL models. The evaluation is performed on both binary and multiclass UCI datasets, and the comparison is based on the outcomes outlined in Table 3.6.

Table 3.6: Comparision of Neuro-Fuzzy based RVFL (proposed) and BLS models in terms of accuracy, standard deviation, rank, number of fuzzy rules for binary and multiclass classification.

	Model →	RVFL [1]	BLS [3]	Fuzzy-BLS [56]	NF-RVFL-R†	NF-RVFL-K†	NF-RVFL-C†
Binary	Avg. Accuracy	80.4458	81.9585	82.4748	82.2584	82.5816	**83.5591**
	Avg. Standard Deviation	9.4944	8.2629	8.8137	8.0372	**6.9893**	7.3014
	Avg. Rank	5.4138	4.3448	3.1552	3.3621	3.069	**1.6552**
	# Avg. Fuzzy Rules	N/A	N/A	330.52	**23.28**	27.76	26.38
Multiclass	Avg. Accuracy	78.4569	79.183	79.3606	78.5363	78.6914	**80.222**
	Avg. Standard Deviation	5.4311	4.9593	6.6522	4.8882	5.1418	**4.5872**
	Avg. Rank	4.2679	3.6429	3.0179	4.3393	3.8036	**1.9286**
	# Avg. Fuzzy Rules	N/A	N/A	450.36	30.18	36.25	**30**

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

Analysis:

1. The standard BLS outperforms RVFL in terms of average accuracy in both binary and multiclass classification.

2. Neuro-fuzzy integration significantly enhances model performance. In binary classification, RVFL and BLS exhibit average accuracies of 80.4458% and 81.9585%, respectively. On the other hand, NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C achieve average accuracies of 82.2584%, 82.5816%, and 83.5591%, respectively. Similarly, Fuzzy-BLS attains an average accuracy of 82.4748%, surpassing the accuracy of BLS at 81.9585%. This pattern holds true for multiclass classification as well.

3. While standard RVFL models may not match the performance of BLS, notable advancements are observed with NF-RVFL models. NF-RVFL-K and NF-RVFL-C surpass both BLS and Fuzzy-BLS in binary classification. NF-RVFL-R outperforms BLS and competes closely with Fuzzy-BLS in binary classification. Remarkably, the proposed NF-RVFL-C model outshine all baseline and neuro-fuzzy-based models for multiclass classification. This underscores the effectiveness of our proposed NF-RVFL-C in incorporating a fuzzy inference system within its structure.

4. In binary classification, all proposed NF-RVFL models exhibit the lowest standard deviation, and in multiclass classification, NF-RVFL-C, NF-RVFL-R, and NF-RVFL-K secure

the first, second, and fourth positions with the lowest standard deviation, respectively. This emphasizes that the proposed NF-RVFL models not only demonstrate superior prediction capabilities but also achieve heightened certainty.

5. NF-RVFL-C secures the top rank with 1.6552 and 1.9286 in binary and multiclass classification, respectively, showcasing a substantial lead over other models. NF-RVFL-K outperforms both BLS and Fuzzy-BLS in binary classification.

6. The crucial factor of the number of fuzzy rules significantly influences the simplicity and interpretability of neuro-fuzzy models. Notably, the optimal number of fuzzy rules (average) for NF-RVFL models consistently falls within the range of 20-35, whereas Fuzzy-BLS utilizes 330.52 and 450.36 fuzzy rules for binary and multiclass classification, respectively—roughly 15-20 times more than that of NF-RVFL. This highlights the superior structural fuzzy adaptability of our proposed NF-RVFL models, indicating a less complex yet high-performing alternative than Fuzzy-BLS.

Main takeaways from the Analysis:

1. While RVFL's generalization performance may not rival that of BLS, the proposed NF-RVFL consistently outperforms BLS and Fuzzy-BLS and sometimes demonstrates competitive results.

2. NF-RVFL exhibits better interpretability than Fuzzy-BLS due to its reduced complexity and enhanced structural adaptability, particularly in fuzzy rule utilization.

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

The significance of regression tasks becomes evident within the RVFL domain, where current research is relatively constrained. Here, we compare the proposed NF-RVFL models

with the standard RVFL and ELM models. We fix the value of the regularization parameter $\mathfrak{C} = 10^5$. The $RMSE$ value is calculated to measure the performance of the proposed models against the baseline models. $RMSE = \sqrt{\frac{1}{M}\sum_{r=1}^{M}(t_r - t_r{}^*)}$, where M, t_r and $t_r{}^*$ is the total number of samples, real output and predicted output of the rth sample, respectively. A lower $RMSE$ value signifies superior model performance. Table 3.7 provides an overview of the $RMSE$ values associated with each model across the respective datasets. After analysis, it becomes evident that the proposed NF-RVFL-R model exhibits the most favorable outcome, with an average $RMSE$ value of 8.0×10^{-6}. Following this, the proposed NF-RVFL-C and NF-RVFL-K models demonstrate competitive performance, registering 8.4×10^{-6} and 9.4×10^{-6} $RMSE$ values, respectively. The standard RVFL model occupies the fourth position, whereas the standard ELM model showcases the worst performance, evidenced by $RMSE$ value of 5.8×10^{-2}. The $RMSE$ value of the standard ELM is approximately 10000 higher than the RVFL family's model. ELM can be seen as a variant of RVFL without direct links. In this context, ELM can be referred to as RVFLwoDL (RVFL without Direct Links). Thus, we conclude that the generalization performance of RVFL with direct links is better than without direct links. Win-tie-loss for individual models is summarized in Table 3.8. In comparison with the ELM model, the proposed NF-RVFL-R and NF-RVFL-C models secure victories across all datasets, whereas NF-RVFL-K attains victory on 14 instances and experiences defeat in 1 out of a total of 15 datasets. This shows the dominance of the proposed NF-RVFL models over the existing baseline models. Thus, the emergence of the proposed NF-RVFL models introduces a promising avenue, demonstrating its potential to excel in addressing regression challenges effectively.

Table 3.7: The *RMSE* values and the corresponding best hyperparameters of the proposed NF-RVFL-R, NF-RVFL-K and NF-RVFL-C models along with the compared baseline models, *i.e.,* RVFL, ELM on regression UCI datasets.

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

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

Table 3.8: Comparison of win-tie-loss results on UCI regression datasets.

	RVFL [1]	ELM [22]	NF-RVFL-R†	NF-RVFL-K†
ELM [22]	$[0, 1, 14]$			
NF-RVFL-R†	$[8, 1, 6]$	$[15, 0, 0]$		
NF-RVFL-K†	$[4, 1, 10]$	$[14, 0, 1]$	$[4, 4, 7]$	
NF-RVFL-C†	$[6, 0, 9]$	$[15, 0, 0]$	$[2, 1, 12]$	$[8, 0, 7]$

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

3.4.6 Interpretability of the Proposed NF-RVFL Models

Interpretability plays a pivotal role in system explanation, aiming to render the internal mechanisms of a system comprehensible to humans. As per [176], two primary categories of interpretability methods exist: intrinsic ante-hoc explanations, centered on transparent and straightforward models such as linear regression and decision trees, and post-hoc explanations, employing methods like PDP [171] and LIME [177] after model training. NF-RVFL, a network model, falls within the local post-hoc interpretability category, extracting fuzzy rules on the RVFL architecture for prediction. Due to the absence of a systematic definition and standardized evaluation methods for interpretability in machine learning, we adopt the established post-hoc method, *i.e.,* PDP and also following the guidance of articles [169, 178, 179], we assess the interpretability of the proposed NF-RVFL models. The interpretability of NF-RVFL is evaluated based on two factors: (i) PDP-based feature importance interpretability, and (ii) feature-importance-based interpretability considering weights assigned to each feature during prediction (following [169, 178]).

3.4.6.1 Dependence Plot (PDP)-based feature importance interpretability

To validate the interpretability of the proposed NF-RVFL models, we employed Partial Dependence Plots (PDP) [171], a standard technique for understanding the feature-

prediction relationship. PDP enables the exploration of feature influence on projected outcomes and their importance according to the model. Considering decision trees as one of

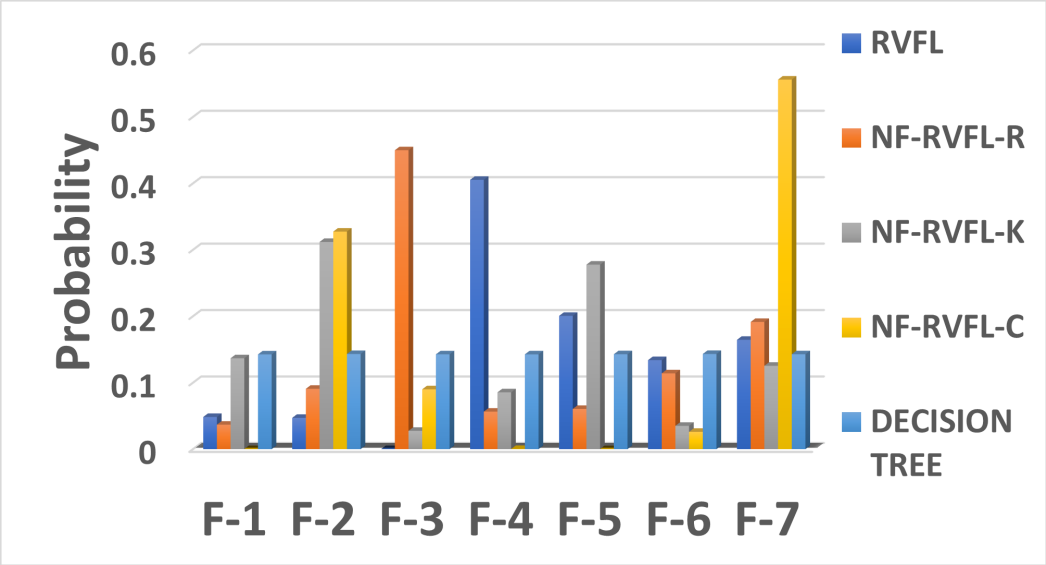


Figure 3.2: Partial Dependence Plot (PDP) of each feature of the dataset pittsburg_bridges_T_OR_D *w.r.t.* the proposed NF-RVFL models, RVFL and Decision Tree. F-i in the X-axis denotes the ith feature of the dataset.

the best interpretable models [180], we compare the NF-RVFL and standard RVFL models with the decision tree. We depict PDP plots in Figure 3.2 for all 7 features of the dataset pittsburg_bridges_T_OR_D. For feature ranking, each feature is assigned ranks based on probability scores, with lower ranks indicating higher importance. In this context, NF-RVFL-C assigns rank 1 to feature-7 (F-7), rank 2 to feature-2 (F-2) and so on. Comparing rank errors (in $RMSE$) between decision tree and NF-RVFL-R, NF-RVFL-K, NF-RVFL-C, and RVFL, the errors are 2.14, 2.62, 2.39, and 2.93 respectively. The lower rank error of the proposed NF-RVFL models with the decision tree (compared to the standard RVFL model) reveals that the proposed NF-RVFL models align more closely with the decision tree in terms of feature importance ranking. Consequently, this heightened congruence implies that the decision-making process within the proposed NF-RVFL models closely resembles that of the decision tree (which is one of the best interpretable models). Hence, the inter-

pretability of the proposed NF-RVFL models is enhanced to some extent in comparison to the RVFL model.

3.4.6.2 Feature importance-based interpretability considering weights assigned to each feature during prediction

This study presents the interpretability of the proposed NF-RVFL models using a feature-importance-based approach, wherein weights assigned to each feature during prediction are considered. The step-by-step learning process of the NF-RVFL models is demonstrated using the breast cancer dataset.

Following [169], we consider the Breast Cancer dataset, which aims to predict the likelihood of breast cancer recurrence based on diagnostic measurements (features). Each data point in the dataset is categorized as either class 1 (no recurrence) or class 2 (recurrence). It consists of nine features labeled as x_1 to x_9.

Breast Cancer Dataset

Each feature in the provided Breast Cancer dataset and their meanings are given in Table 3.9. Recurrence events indicate that after initial treatment for Breast Cancer, the disease has returned or reappeared at the same site or has spread to other parts of the body, and no-recurrence-events, on the other hand, indicate that after the initial treatment, there is no evidence of cancer returning or spreading.

Learning Process of the Proposed NF-RVFL-K Model

To provide a transparent illustration of interpretability, we first illustrate, how our proposed NF-RVFL-K learns fuzzy rules and assigns weights to the original features during the training process at the minimum number of hidden nodes and fuzzy rules. Then, following [178], we show the interpretability of all the proposed NF-RVFL models at the

Table 3.9: Description of features in the Breast Cancer dataset.

Feature	**Description**	**Values**
Class	Target variable indicating cancer recurrence	"No recurrence," "Recurrence"
Feature-1	Age (x_1)	Ranges from 10-99, grouped into age categories
Feature-2	Menopause (x_2)	"lt40" (<40), "ge40" (≥ 40), "premeno" (premenopausal)
Feature-3	Tumor Size (x_3)	Ranges from 0-59, grouped into size categories
Feature-4	Inv-Nodes (x_4)	Ranges from 0-39, grouped into categories
Feature-5	Node Caps (x_5)	"yes" or "no"
Feature-6	Deg Malig (x_6)	1, 2, 3
Feature-7	Breast (x_7)	"left" or "right"
Feature-8	Breast-Quad (x_8)	"left-up," "left-low," "right-up," "right-low," "central"
Feature-9	Irradiat (x_9)	"yes" or "no"

best hyperparameter setting.

Step-1: The Breast Cancer dataset's input vector $X = [x_1, x_2, \ldots, x_9]$ is converted into five fuzzy features through fuzzification. and for a sample u_1, *i.e.,* $u_1 = [1.61, -0.76, 0.77, -0.27, 0.75, -1.3, -0.79, -0.81, 0.8]$, we demonstrate the process.

Step-2: Centre for each fuzzy rule is generated using K-means clustering and is listed below:

$$\text{Centre} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0.27 & -1.33 & 0.45 & 0.32 & -0.39 & 0.93 & -0.28 \\ -1.41 & -0.05 & 0.15 & 0.35 & -0.47 & 1.19 & 0.24 \\ 0.95 & 0.62 & 0.01 & -0.47 & -0.84 & -0.59 & 0.35 \\ 0.62 & -1.29 & 0.23 & -0.40 & 0.83 & -0.68 & 0.70 \\ -1.28 & 0.60 & 0.47 & -0.03 & 0.52 & -0.58 & 0.29 \end{bmatrix}$$

and standard deviation *i.e.,* $\sigma = 1$ for each fuzzy nodes.

Step-3: The consequent part of each fuzzy rule is established through the output of a linear function, which is determined by the membership degrees of the input variables and the corresponding random coefficient arrays α, and it is given below:

Table 3.10: Fuzzy vector generation through fuzzification.

If	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u_{17}	u_{18}	u_{19}
	1.61	-0.76	0.77	-0.27	0.75	-1.3	-0.79	-0.81	0.8
Since $y_{1j} = \sum_{s=1}^{9} \alpha_{js} u_{1s}$									
	$y_1 1$	$y_1 2$	$y_1 3$	$y_1 4$	$y_1 5$				
	0.67	1.26	-0.02	-0.29	-0.42				
The weighted fire strength									
	$\omega_1 1$	$\omega_1 2$	$\omega_1 3$	$\omega_1 4$	$\omega_1 5$				
	1.00E-04	0	0.0005	0.9994	0				
Fuzzy vector of the first sample is $Y_1 = (\omega_{11} y_{11}, \omega_{12} y_{12}, \ldots, \omega_{15} y_{15})$									
then	$Y_1 =$	0.5645	0	0	-0.0463	-1.00E-04			

$$\alpha = \begin{bmatrix} 1 & 0.24 & 0.92 & 0.32 & 0.55 \\ 0.72 & 0.08 & 0.64 & 0.37 & 0.58 \\ 0.65 & 0.96 & 0.57 & 0.96 & 0.24 \\ 0.28 & 0.68 & 0.47 & 0.32 & 0.94 \\ 1 & 0.7 & 0.37 & 0.58 & 0.67 \\ 0.86 & 0.07 & 0.36 & 0.49 & 0.85 \\ 0.33 & 0.22 & 0.65 & 0.82 & 0.83 \\ 0.38 & 0.33 & 0.95 & 0.78 & 0.11 \\ 0.16 & 0.49 & 0.18 & 0.37 & 0.68 \end{bmatrix}.$$

Step-4: The fuzzified feature vector of the sample u_1 (through fuzzification) is presented in Table 3.10.

Step-5: For a given number of hidden nodes $h = 3$, subsequent to the feature enhancement process, the concatenated feature vector for sample u_1 is represented as: $[0.16, 0.05, 0.51, 1.61, -0.76, 0.77, -0.27, 0.75, -1.3, -0.79, -0.81, 0.8]$. The augmented dataset $\mathcal{D} = \left((\Omega \odot (U\alpha^t)), \hat{H} \right)$ is formed by processing the original feature matrix U and

concatenating it with the enhanced feature vector $\hat{H}$. Consequently, the initial nine columns are derived from the original feature matrix U. Subsequently, the output layer weights, denoted as $(W)_{\min}$, are computed, and

$$\beta = \begin{bmatrix} -0.51 & -0.6 \\ -0.94 & -0.41 \\ -0.47 & -0.36 \\ -0.54 & -0.36 \\ -0.83 & -0.31 \\ 1.17 & 0.72 \\ 0.65 & 0.16 \\ 1.15 & 0.82 \\ -0.01 & -0.04 \\ -0.06 & 0.01 \\ -0.11 & -0.03 \\ 0.09 & 0.06 \\ 0.09 & -0.05 \\ -0.05 & 0.14 \\ 0.06 & -0.06 \\ -0.07 & -0.06 \\ 0.07 & 0.01 \end{bmatrix}.$$

Given the output representation $T^* = \mathcal{D}W$, it follows that the initial nine columns of $\mathcal{D}$, directly stemming from the features of the original dataset, corresponding to the first nine rows of the weight matrix β. Thus, the weights assigned to the features of the original matrix U can be extracted. Specifically, the weights for x_1 are denoted by $[-0.51, -0.6]$, the weights for x_2 are represented as $[-0.94, -0.41]$, and so forth, with the weights for x_9 given by $[-0.01, -0.04]$. Consequently, a higher magnitude of weights signifies greater importance attributed to the respective feature by the NF-RVFL-K model during the learning process. The norm vector representing the weights of the first 9 rows of β is delineated as: $[0.79, 1.03, 0.59, 0.64, 0.89, 1.38, 0.67, 1.41, 0.04]$. This signifies that Feature-8 (Breast-

Quad) and Feature-6 (Deg Malig) play pivotal roles in predicting "no-recurrence-events (Class-1)" or "recurrence-events (Class-2)." The proposed NF-RVFL-K model exhibits a human-like interpretability, elucidating the importance of individual features in the learning process. From a biomedical perspective, the prominence of Feature-6, namely, the degree of malignancy (cancer severity), underscores its crucial role in characterizing the status of cancer patients and predicting recurrence. The proposed model thus offers interpretability in terms of feature importance, emphasizing the clinical relevance of certain attributes. It is imperative to note that the number of hidden nodes and fuzzy rules, while not optimized for maximal performance in this illustration, are presented with minimal values for enhanced readability. A more exhaustive exploration of parameter settings during training would yield more accurate and nuanced interpretability.

Feature Importance-Based Interpretability of all the Proposed NF-RVFL Models

Now, we demonstrate the feature importance of each feature of the Breast Cancer dataset for NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C models. We compute the norm of the β vector for each model at the best hyperparameter setting across all nine features. The norms for all features are listed in Table 3.11, and the discussion follows below.

Analysis and main takeaways from the learning process and interpretability of NF-RVFL: In examining the interpretability of our proposed NF-RVFL models, we analyze the following:

- For NF-RVFL-C, feature emphasis is notably placed on features x_9 (Irradiat), x_6 (Deg Malig), and x_3 (Tumor Size). These features are associated with radiation therapy, cancer severity, and tumor size. As per biomedical knowledge [181], these are some prominent features in predicting whether breast cancer has no recurrence or recurrence. Align with biomedical knowledge; the proposed NF-RVFL-C gives signifi-

Table 3.11: The norm of the weights (from β) linking Breast Cancer features to the output layer for the final prediction in the proposed NF-RVFL models. The left column indicates the features, while the corresponding row entries for each feature denote the weights assigned by each model.

Features	NF-RVFL-R	NF-RVFL-K	NF-RVFL-C
x_1	0.3023×10^{-3}	0.6352×10^{-3}	0.0742×10^{-3}
x_2	0.4373×10^{-3}	0.4561×10^{-3}	0.0921×10^{-3}
x_3	0.4829×10^{-3}	0.9991×10^{-3}	0.0943×10^{-3}
x_4	0.3598×10^{-3}	0.5046×10^{-3}	0.073×10^{-3}
x_5	0.4728×10^{-3}	0.9835×10^{-3}	0.061×10^{-3}
x_6	0.3388×10^{-3}	0.4657×10^{-3}	0.1006×10^{-3}
x_7	0.4548×10^{-3}	0.6586×10^{-3}	0.0482×10^{-3}
x_8	0.3337×10^{-3}	1.03×10^{-2}	0.0441×10^{-3}
x_9	0.4001×10^{-3}	0.317×10^{-3}	0.1047×10^{-3}

cance to the above features in predicting breast cancer recurrence.

- For NF-RVFL-R, the model predominantly prioritizes x_3 (Tumor Size). This concurs with biomedical understanding, highlighting the pivotal role of tumor size in predicting cancer diseases.

- For NF-RVFL-K, the model assigns primary weightage to x_8 (Breast-Quad) and secondary importance to x_3 (Tumor Size). NF-RVFL-K adeptly identifies and prioritizes features, aligning with the final output and showcasing its ability to discern meaningful patterns.

In summary, our study demonstrates the learning rules and feature importance-based interpretability of the proposed NF-RVFL models. The models exhibit nuanced prioritization of features, aligning with biomedical knowledge and enhancing the understanding of their predictive capabilities in the context of breast cancer recurrence. In accordance with Friedman [171], "visualization stands as one of the most potent tools for interpretation". Since the neuro-fuzzy IF-THEN rule is based on the human-like visualization approach and from the

above two interpretability discussions, the proposed model gets closer to interpretability.

3.5 Ablation Study and Further Analysis

In this section, we conduct an ablation study on the proposed NF-RVFL models, examining their dependency on key components. The essential elements under investigation include the direct link (connecting the input and output layers), fuzzification (Fuzzy layer), defuzzification (defuzzified layer), number of fuzzy rules (clusters), and activation functions. The ablation study specifically focuses on the direct link, fuzzification, and defuzzification in the first subsection. In subsequent subsections, separate analyses are conducted on the impact of the number of fuzzy rules (clusters) and activation functions.

3.5.1 Ablation Study

The abbreviations used for the proposed NF-RVFL models with and without key components are presented in Table 3.12. For our ablation study, we selected six datasets, encompassing two from binary classification, two from multiclass classification, and two from regression. To ensure impartiality in our model evaluations, we include one dataset from each family (binary, multiclass, and regression) where none of our proposed NF-RVFL models emerged as a top performer. Additionally, we take one dataset from each family where at least one of our proposed NF-RVFL models demonstrated superior performance. Further details about the selected datasets are reported in Table 3.13. The performance of the proposed NF-RVFL models is reported in Table 3.14.

Analysis for the classification task:

[1] The consistently strong performance of the proposed NF-RVFL-C model, both with and without key components, positions it as a perennial top performer.

Table 3.12: Abbreviation used for the NF-RVFL models with and without key components for the ablation analysis. ✓and ✗ denote the proposed model (row) with and without key component (column), respectively.

Models	Key Components			Abbreviation Given
	Direct Link	Fuzzification Layer	Defuzzification Layer	
NF-RVFL	✗	✗	✗	ELM
NF-RVFL	✓	✗	✗	RVFL
NF-RVFL-R	✓	✓	✓	R-M1 (NF-RVFL-R)
NF-RVFL-R	✓	✓	✗	R-M2
NF-RVFL-R	✗	✓	✓	R-M3
NF-RVFL-R	✗	✓	✗	R-M4
NF-RVFL-K	✓	✓	✓	K-M1 (NF-RVFL-K)
NF-RVFL-K	✓	✓	✗	K-M2
NF-RVFL-K	✗	✓	✓	K-M3
NF-RVFL-K	✗	✓	✗	K-M4
NF-RVFL-C	✓	✓	✓	C-M1 (NF-RVFL-C)
NF-RVFL-C	✓	✓	✗	C-M2
NF-RVFL-C	✗	✓	✓	C-M3
NF-RVFL-C	✗	✓	✗	C-M4

Table 3.13: The details of the datasets used for the ablation study.

	Binary		Multiclass		Regression	
Datasets	heart_hungarian	pittsburg_bridges_T_OR_D	energy_y1	flags	Triazines	Yacht_Hydrodynamics
The proposed NF-RVFL model is top performer or not (Yes/No)	No	Yes	No	Yes	Yes	No

Table 3.14: Performance (in terms of accuracy) of the proposed NF-RVFL models with and without key components for ablation analysis. The top three models are boldfaced *w.r.t.* each dataset.

Dataset →	Classification					Regression		
Model ↓	energy_y1	flags	heart_hungarian	pittsburg_bridges_T_OR_D	Average	Triazines	Yacht_Hydrodynamics	Average
ELM	88.2777	51.5789	74.4886	87.3685	75.4284	0.686413	0.000206	0.3433
RVFL	**88.6716**	53.6302	73.8223	87.1905	75.8287	3.50E-05	4.00E-06	1.95E-05
R-M1	87.979	**55.2687**	76.9024	**90.1905**	**77.5852**	2.16E-05	5.77E-06	1.37E-05
R-M2	87.7591	53.6032	77.9252	90.1429	77.3576	**1.56E-05**	**2.90E-07**	**7.97E-06**
R-M3	85.9426	47.4359	72.7294	90.1429	74.0627	0.88472001	0.0006817	0.4427
R-M4	86.4621	46.9906	73.8983	90.1429	74.3735	0.90749518	0.00081535	0.4542
K-M1	**88.5434**	55.1957	76.5576	88.2381	77.1337	2.55E-05	8.02E-06	1.68E-05
K-M2	87.3703	54.6559	**79.8948**	90.1429	78.016	**1.61E-05**	**2.90E-07**	**8.20E-06**
K-M3	86.8483	48.5425	73.4717	**91.1905**	75.0133	0.87304478	0.00074612	0.4369
K-M4	86.1998	49.0283	76.1952	89.2381	75.1654	0.82710537	0.00077904	0.4139
C-M1	**89.0595**	**55.2092**	78.2466	**90.1905**	**78.1765**	2.20E-05	7.01E-06	1.45E-05
C-M2	88.1564	**55.722**	78.2466	**91.1429**	**78.317**	**1.50E-05**	**2.90E-07**	**7.64E-06**
C-M3	88.1487	54.17	**79.5733**	90.1429	78.0087	0.5484	0.0006775	0.2745
C-M4	87.6326	54.1565	**79.4915**	**90.1905**	77.8678	0.53088702	0.00067188	0.2658

[2] In terms of average accuracy, the leading performers are the proposed NF-RVFL-C (C-M1), C-M2, and NF-RVFL-R (R-M1). This underscores the substantial impact of

the fuzzification layer and direct links on the models' performance.

[3] Notably, for the proposed NF-RVFL-R model, the presence of all key components significantly influences its performance.

[4] A noteworthy observation is that the proposed models lacking a direct link (and/or defuzzification layer, and/or fuzzification layer) rank among the least-performing models.

Analysis for the regression task:

[1] For each dataset, the proposed NF-RVFL models without a defuzzification link (R-M2, K-M2, and C-M2) demonstrate superior performance.

[2] An interesting observation is that models without defuzzification links and direct links consistently rank among the lowest performers.

These analyses highlight the critical role of key components in the proposed NF-RVFL models, showcasing their impact on classification and regression tasks.

3.5.2 Number of Fuzzy Rules (clusters) V/S Performance of NF-RVFL Models

To assess the influence of fuzzy rules (clusters) on the performance of the proposed NF-RVFL models, experiments are conducted varying the number of fuzzy rules (J), and the accuracy of the NF-RVFL models are investigated. The experimental outcomes are depicted in Figs. 3.3a, 3.3b, 3.3c, 3.4a, 3.4b, and 3.4c.

The noteworthy analyses are as follows:

[1] Regression Datasets: The optimal performance of all proposed NF-RVFL models is consistently achieved with a minimal number of fuzzy rules, $J = 5$. This underscores

the models' proficiency at lower complexity and improved interpretability. Thus, for regression tasks, the recommended choice for the number of fuzzy rules is 5 for NF-RVFL models.

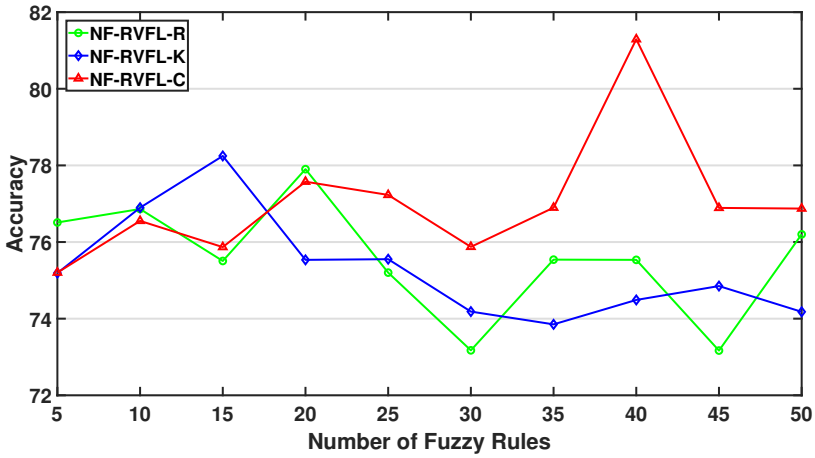
[2] Binary Datasets: For NF-RVFL-R and NF-RVFL-K in binary datasets, the optimal J is consistently less than or equal to 20, highlighting the models' adeptness at adapting to neuro-fuzzy structures with a limited number of fuzzy rules. NF-RVFL-C, however, exhibits optimal values between 20 and 40.

[3] Multiclass Classification: Mixed patterns are observed for multiclass classification. For the energy_y1 dataset, the performance of proposed models improves with an increasing number of fuzzy rules. Whereas, for the flag dataset, no discernible pattern is observed.

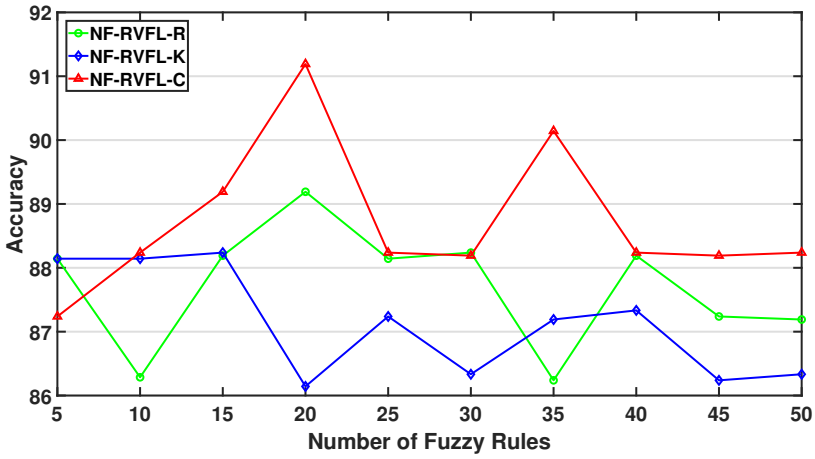
Broadly, the observations suggest that the proposed NF-RVFL models exhibit optimal training with a modest number of fuzzy rules. However, performance varies depending on the dataset and application, adhering to the No Free Lunch Theorem [182]. Therefore, tuning fuzzy rules is recommended for enhanced generalization performance of the proposed NF-RVFL model.

3.5.3 Role of an Activation Function in the Proposed NF-RVFL Models

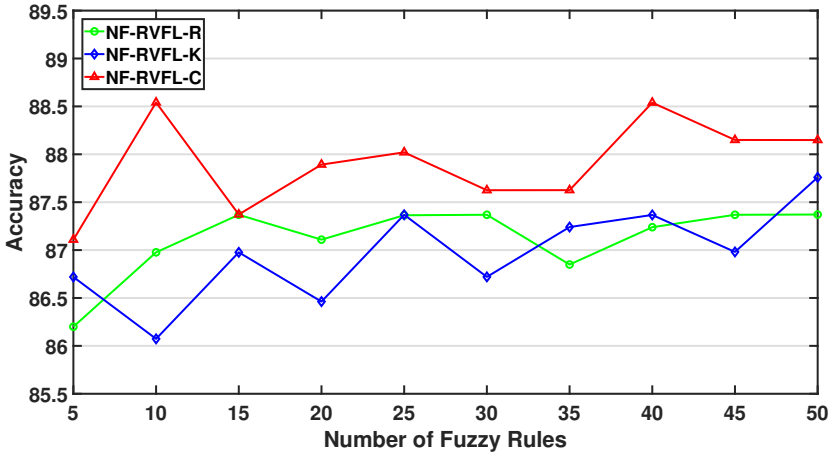
Activation functions play a pivotal role in neural networks by introducing non-linearity, enabling models to capture complex relationships within data. These functions determine the output of individual neurons, facilitating information propagation, feature extraction and ultimately enhancing the network's capacity to learn and generalize from diverse inputs [183]. Since activation functions are important in how neural networks work, we



(a) Binary data: heart_hungarian

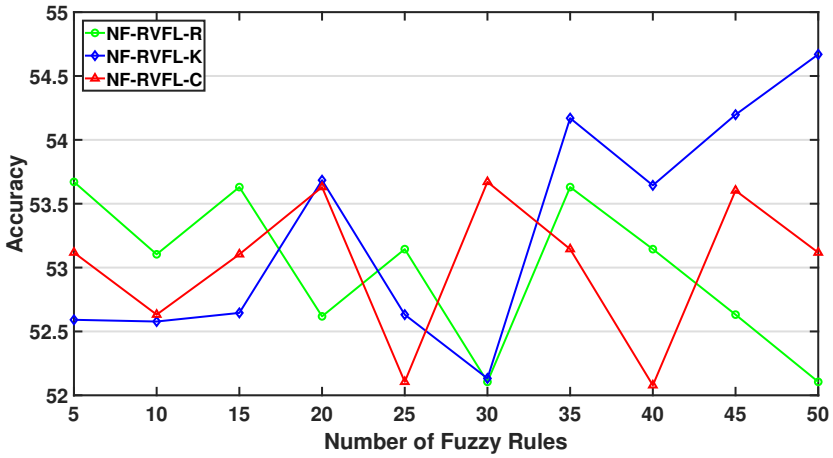


(b) Binary data: pittsburg_bridges_T_OR_D

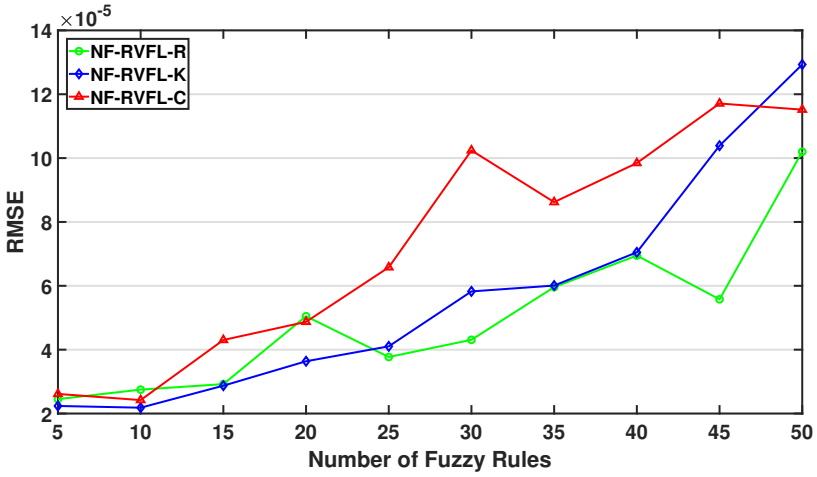


(c) Multiclass data: energy_y1

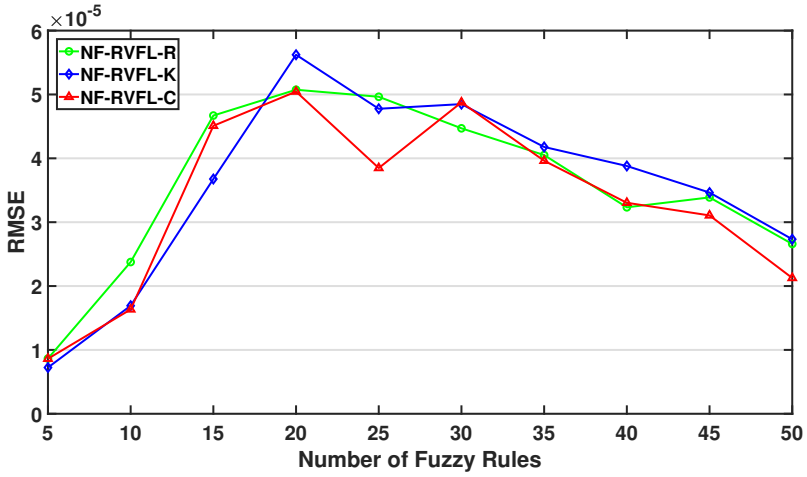
Figure 3.3: (a) and (b) showcase the accuracy of the proposed NF-RVFL models on binary and multiclass datasets, respectively, whereas (c) showcases the $RMSE$ of the proposed NF-RVFL models on regression datasets with respect to activation functions.



(a) Multiclass data: flags

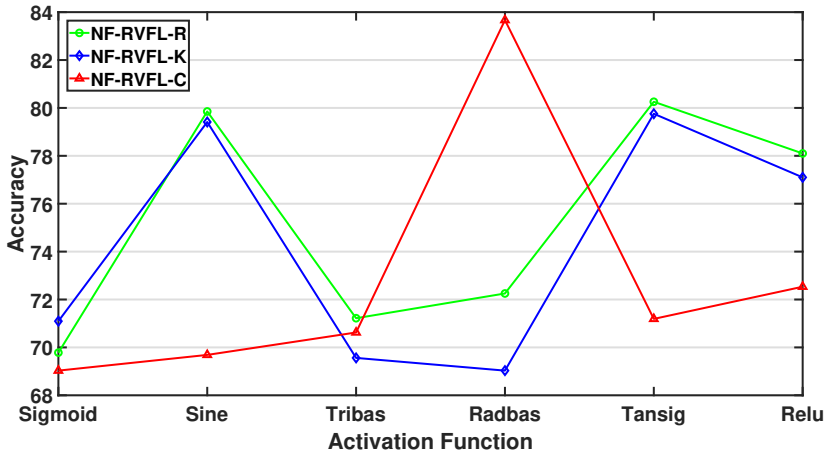


(b) Regression data: Triazines

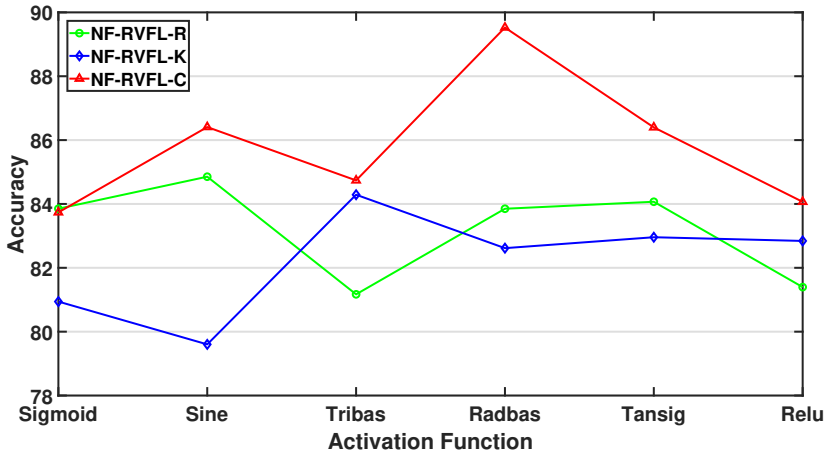


(c) Regression data: Yacht_Hydrodynamics

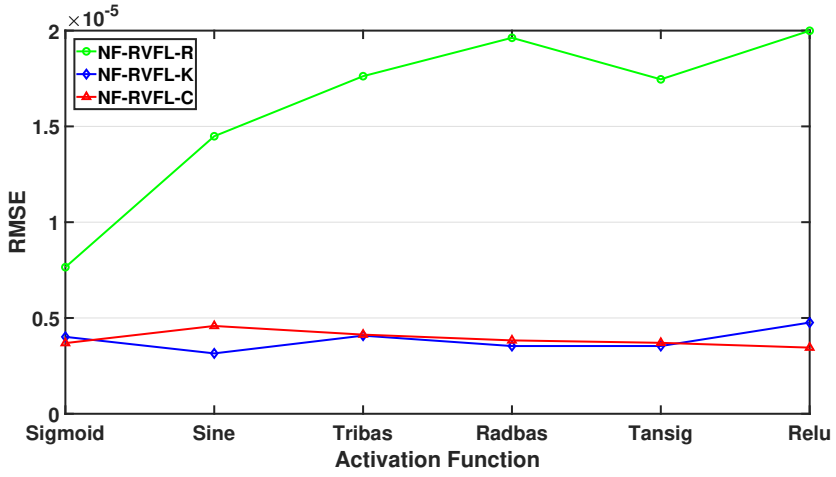
Figure 3.4: (a) and (b) showcase the accuracy of the proposed NF-RVFL models on binary and multiclass datasets, respectively, whereas (c) showcases the $RMSE$ of the proposed NF-RVFL models on regression datasets with respect to activation functions.



(a) Binary data: chess_krvp



(b) Multiclass data: annealing



(c) Regression data: forest_fires

Figure 3.5: (a) and (b) showcase the accuracy of the proposed NF-RVFL models on binary and multiclass datasets, respectively, whereas (c) showcases the $RMSE$ of the proposed NF-RVFL models on regression datasets with respect to activation functions.

focus on fine-tuning the activation functions. We explore five distinct activation functions: $Sigmoid$ (1), $Sine$ (2), $Tribas$ (3), $Radbas$ (4), $Tansig$ (5) and $Relu$ (6). Figures 3.5a, 3.5b and 3.5c illustrate NF-RVFL model performance on binary, multiclass, and regression datasets (chess_krvp, annealing, and forest_fires) based on different activation functions. The graphs highlight how deliberate activation function choices significantly improve model performance, enhancing its capacity to capture intricate patterns and optimize overall effectiveness. Thus, proper selection of activation functions is crucial for optimizing model performance and achieving successful outcomes. In this regard, an examination of the number of activation functions provides insight into the relative significance of each during the learning phase of NF-RVFL models. The activation function counts are presented in Table 3.15.

Key Insights from Table 3.15:

[1] In binary classification, the $Tansig$ activation function, followed by $Sine$, is most frequently observed. For multiclass classification, $Radbas$ prominently dominates, while $Relu$ and $Tribas$ emerge as optimal choices for regression tasks.

[2] In the NF-RVFL-R model, $Sine$, $Radbas$, and $Relu$ activation functions significantly impact optimal performance across binary, multiclass, and regression tasks, respectively. Overall, $Sine$, $Relu$, and $Radbas$ play a substantial role in the NF-RVFL-R model.

[3] For the NF-RVFL-K model, $Tansig$ and $Radbas$ are optimal choices for binary and multiclass classification. Notably, $Radbas$ is not identified as an optimal choice for regression tasks where a mixed influence of other activation functions is observed. Overall, $Relu$ and $Tansig$ majorly contribute to the NF-RVFL-K model's enhanced performance.

Table 3.15: Count of activation functions for proposed NF-RVFL models.

	Activation Function	NF-RVFL-R	NF-RVFL-K	NF-RVFL-C	Total
Binary	$Sigmoid$	3	4	1	8
	$Sine$	9	6	5	20
	$Tribas$	2	2	1	5
	$Radbas$	4	3	4	11
	$Tansig$	7	9	8	24
	$Relu$	4	5	7	16
Multiclass	$Sigmoid$	5	2	4	11
	$Sine$	5	3	2	10
	$Tribas$	4	4	2	10
	$Radbas$	7	9	8	24
	$Tansig$	2	4	6	12
	$Relu$	5	6	6	17
Regression	$Sigmoid$	2	3	1	6
	$Sine$	1	2	4	7
	$Tribas$	3	4	4	11
	$Radbas$	2	0	1	3
	$Tansig$	2	2	2	6
	$Relu$	5	4	3	12
All	$Sigmoid$	10	9	6	25
	$Sine$	15	11	11	37
	$Tribas$	9	10	7	26
	$Radbas$	13	12	13	38
	$Tansig$	11	15	16	42
	$Relu$	14	15	16	45

The last box (row) represents the count of activations *w.r.t.* the proposed models for all the datasets, and the last column denotes the combined count of each activation functions *w.r.t.* for all the proposed NF-RVFL-L models.

[4] In the NF-RVFL-C model, $Tansig$ and $Relu$ activation functions significantly influence optimal performance, and the same pattern can be observed for binary classification. Meanwhile, for multiclass classification, $Radbas$ excels. In regression tasks, $Sine$ and $Tribas$ demonstrate substantial impact.

[5] Further analyzing the last column of the last row-box, $Relu$, and $Tansig$ occur most frequently as an optimal choice for the proposed NF-RVFL model.

In conclusion, tuning the activation function during the training process of NF-RVFL mod-

els can enhance model performance. However, $Tansig$ and $Relu$ would be preferable if one wants to optimize the model's generalization performance.

3.6 Summary

This chapter proposed the NF-RVFL model by integrating RVFL with a neuro-fuzzy system to improve the interpretability of the learning and decision-making process. The proposed framework introduces a fuzzy layer that enables human-understandable reasoning through an IF–THEN rule mechanism, allowing the model to provide more transparent predictions. The fuzzified representations are combined with random projections and defuzzified values, along with the original input features, to produce the final prediction. Three clustering strategies, namely randomly initialized centers (R-means), K-means clustering, and fuzzy C-means clustering, were employed to construct the fuzzy layer, resulting in three model variants: NF-RVFL-R, NF-RVFL-K, and NF-RVFL-C.

The proposed models were evaluated on 29 binary and 28 multiclass classification UCI datasets, along with 15 regression datasets. The experimental results show that NF-RVFL-C achieves the best overall performance, while NF-RVFL-K and NF-RVFL-R secure the second and third positions, respectively, based on average accuracy and ranking. In regression tasks, the NF-RVFL models consistently achieve lower RMSE values and higher numbers of wins compared to the baseline models, demonstrating their strong generalization capability.

From the interpretability perspective, the NF-RVFL model provides meaningful insights into the prediction process through partial dependence plot (PDP)-based feature importance on the Pittsburgh Bridge dataset and weight-based feature importance analysis on the Breast Cancer dataset. These analyses highlight the contribution of individual features to the model predictions and reveal the role of fuzzy rules in shaping the decision boundaries. Compared to the standard RVFL model, the proposed NF-RVFL offers improved transparency in the

learning process. In addition, NF-RVFL demonstrates better interpretability than Fuzzy-BLS due to its simpler structure and the use of fewer fuzzy rules. An ablation study further investigates the impact of important components, including direct links, fuzzification, defuzzification, number of fuzzy rules, and activation functions, confirming their contribution to both predictive performance and interpretability.

Chapter 4

Robust and Scalable Randomized Neural Networks based on Granularity and Loss Function

Outcome of this Chapter

1. Publication: M. Sajid, A. Quadir and M. Tanveer (2025). "GB-RVFL: Fusion of Randomized Neural Network and Granular Ball Computing," in ***Pattern Recognition, Elsevier***, v. 159, p. 111142, **DOI:** https://doi.org/10.1016/j.patcog.2024.111142 [**SCI Indexed with Impact Factor.:** 9.1, Q1]

Source Code: https://github.com/mtanveer1/GB-RVFL

2. Publication: M. Sajid, A. Quadir and M. Tanveer (2024). "Wave-RVFL: A Randomized Neural Network Based on Wave Loss Function," in ***The 31st International Conference on Neural Information Processing (ICONIP)***, v. 15287, pp. 242-257,

DOI: https://doi.org/10.1007/978-981-96-6579-2_17 [**Core Rank B Conference**]

Source Code: https://github.com/mtanveer1/Wave-RVFL

In the previous chapter, we proposed NF-RVFL, which successfully addresses the limitations of interpretability; however, conventional RdNNs suffer from several other limitations, including sensitivity to noise and outliers, reliance on the square loss function, and limited scalability due to the need to invert the entire training dataset. To address these issues, this chapter proposes two extensions of the RVFL framework. First, we introduce the granular ball RVFL (GB-RVFL), which uses granular balls (GBs) instead of individual samples for classifier construction. This improves scalability by requiring the inversion of only the GB center matrix and enhances robustness to noise. To further preserve the intrinsic geometric structure of the data, we extend this model to graph embedded GB-RVFL (GE-GB-RVFL). Second, we propose Wave-RVFL, which incorporates the Wave loss function into the RVFL framework. The optimization problem is solved using the Adam algorithm, eliminating matrix inversion and improving robustness against noise and outliers. Experimental results on benchmark and biomedical datasets demonstrate the effectiveness of the proposed models compared to baseline methods.

4.1 Background

Although RVFL has demonstrated promising performance across various applications, it still faces several challenges. First, the computation of output weights in conventional RVFL requires the inversion of a matrix formed from the entire training dataset, which may become computationally expensive or intractable for large-scale problems. Second, the traditional RVFL treats each training sample equally during optimization, making it sensitive to noise and outliers present in real-world datasets. Third, the standard formulation typically relies on the square error loss function, which assumes Gaussian-distributed errors and can excessively penalize large deviations, leading to instability and reduced robustness when noisy data or outliers are present. Finally, traditional RVFL does not explicitly exploit

the intrinsic geometric structure of the data during model construction [25].

Several improved variants of RVFL have been proposed to address these limitations. For example, intuitionistic fuzzy RVFL (IFRVFL) [32] assigns fuzzy membership and non-membership scores to samples to mitigate the effect of noise and outliers. Graph-based variants such as GE-IFWRVFL [35] and GE-IFRVFL-CIL [39] incorporate graph embedding frameworks to preserve the intrinsic geometric structure of the dataset. Other approaches include SP-RVFL [27], which integrates sparse autoencoders to stabilize randomized feature representations, and discriminative manifold RVFL (DMRVFL) [184], which leverages manifold learning to enhance discriminative capability. While these models improve certain aspects of RVFL, many of them still rely on the inversion of matrices constructed from the entire dataset, limiting their scalability for large-scale problems.

Motivated by these challenges, this chapter explores two complementary directions to enhance the robustness and scalability of RVFL: granular computing-based learning and robust loss function design.

Granular computing has emerged as an effective framework for handling large-scale and noisy datasets. Inspired by human cognitive processes that follow a "large scope first" principle, granular computing represents data using information granules of different sizes [43, 44]. Larger granules provide improved scalability and robustness, while smaller granules capture finer structural details. Recently, granular ball computing has been proposed as an efficient data representation technique, where datasets are represented using granular balls (GBs) that summarize groups of samples [45]. Based on this idea, granular ball support vector machines (GBSVM) [46] and granular ball twin support vector machine (GB-TSVM) [69] demonstrated improved robustness to noise and outliers by performing learning on GB centres instead of individual data samples.

Inspired by the effectiveness of granular learning, we integrate granular ball computing

with RVFL and propose the granular ball RVFL (GB-RVFL). In GB-RVFL, granular balls are used as inputs instead of individual data samples, significantly reducing the computational burden and improving scalability since the model only requires the inversion of the matrix formed by GB centers. Furthermore, to preserve the intrinsic geometric structure of the dataset, we extend this approach by incorporating graph embedding (GE) [36], resulting in the graph embedded granular ball RVFL (GE-GB-RVFL). The integration of GE allows the model to exploit the topological relationships among data points while maintaining the robustness benefits of granular representation.

Another important direction to enhance the robustness of RVFL is to modify its loss function. The conventional RVFL relies on the square error loss function, which is highly sensitive to outliers and large deviations. To address this limitation, recent studies have explored robust loss functions that provide better resistance to noisy samples. In particular, the Wave loss function [71] has recently shown promising performance in support vector machines due to its smooth behavior, reduced sensitivity to noise, and improved robustness to outliers. This raises an intriguing question: can the Wave loss function be effectively utilized in the optimization function of the RVFL network? If so, can we formulate a solution to this optimization problem, and will the integrated model (RVFL with Wave loss function) exhibit the same robustness against noise and outliers?

Inspired by these promising properties of the Wave loss function and motivated by the potential answers to these questions, we propose the Wave-RVFL model, an RVFL model based on the Wave loss function. In developing Wave-RVFL, we aimed to preserve the inherent optimization formulation of the RVFL as much as possible, simply replacing the square loss function with the Wave loss function. The proposed optimization problem of Wave-RVFL is solved using the adaptive moment estimation (Adam) algorithm [73]. This approach not only eliminates the need for matrix inversion but also improves scalability for

large datasets while enhancing robustness against noise and outliers.

In summary, this chapter presents two novel extensions of the RVFL framework. The first approach introduces granular ball computing and graph embedding to improve scalability, structural learning, and robustness. The second approach incorporates a robust Wave loss function into the RVFL optimization framework to mitigate the influence of noise and outliers while enabling scalable learning through gradient-based optimization. Extensive experiments on benchmark datasets and real-world biomedical applications demonstrate the effectiveness of the proposed approaches.

The key contributions of this chapter are summarized as follows:

[1] We propose the granular ball RVFL (GB-RVFL) model, which utilizes granular balls (GBs) instead of individual samples for classifier construction. This representation improves scalability by requiring the inversion of the GB centers matrix rather than the entire training matrix and enhances robustness against noise and outliers through coarse-grained data representation.

[2] To further preserve the intrinsic geometric structure of the dataset, we propose the graph embedded GB-RVFL (GE-GB-RVFL), which integrates graph embedding with subspace learning criteria. The incorporation of graph regularization enables the model to maintain the structural relationships of the data in the projection space while retaining the advantages of GB-RVFL.

[3] We introduce Wave-RVFL, which integrates the Wave loss function into the RVFL framework to improve robustness against noise and outliers. The proposed formulation prevents excessive penalization of large deviations and leads to a more balanced learning process.

[4] The optimization problem of Wave-RVFL is solved using the adaptive moment esti-

mation (Adam) algorithm, eliminating the need for matrix inversion and significantly improving scalability for large-scale datasets.

[5] Extensive experiments are conducted on benchmark KEEL and UCI datasets under clean and noisy conditions to evaluate the effectiveness of the proposed models. Additionally, experiments on large-scale datasets and real-world biomedical datasets, including BreakHis for breast cancer classification and ADNI for Alzheimer's disease classification, demonstrate the practical applicability and effectiveness of the proposed approaches.

Section 4.2 presents the GB-RVFL and GE-GB-RVFL models along with their validation. Section 4.3 discusses the Wave-RVFL model and its experimental analysis, while Section 4.5 provides the summary of the chapter.

4.2 Proposed Work: Granular Ball-based RVFLs

Following the discussion on Section 2.8, let $\{GB_1, GB_2, \ldots, GB_k\}$ be the set of GBs for the training dataset $\mathcal{T} = (U, T)$. Let o_j and w_j be the center and label of the granular ball GB_j, respectively. Let $\mathcal{O} = [o_1^t, o_2^t, \ldots, o_k^t]^t \in \mathbb{R}^{k \times L}$ and $\mathcal{W} = [w_1^t, w_2^t, \ldots, w_k^t]^t \in \mathbb{R}^{k \times C}$ be the matrix of centers and classes of GBs, respectively. $\otimes$ and $\oplus$ denote the Kronecker product and the Concatenation operator, respectively

4.2.1 Granular Ball Random Vector Functional Link (GB-RVFL) Network

Through the fusion of RVFL and GB computing, we propose GB-RVFL with the aim of achieving greater scalability and robustness compared to the RVFL model. We first highlight

the rationale behind the scalability and robustness of our model, and then we delve into the mathematical framework that fuses GB and RVFL to form GB-RVFL.

- ***Robustness:*** Our proposed GB-RVFL model fuses the concept of GBs with RVFL. During the training, the proposed GB-RVFL captures information from either the entire sample space or from subsets of the sample space (in the form of GBs). These GBs, derived from the training dataset, are inherently coarse and represent a small fraction of the total data points. By leveraging the coarse nature of GBs, specifically focusing on their centers, we effectively harness the bulk of the information situated around these centers. This strategy renders our proposed GB-RVFL model less susceptible to noise and outliers, which are typically situated farther away from the central data distribution or clusters. A comprehensive examination of the robustness of our proposed GB-RVFL model, in comparison to baseline models, has been conducted in Section 4.2.4.5.

- ***Scalability:*** By training the proposed GB-RVFL on GBs instead of the entire training dataset, and considering that the number of GBs is significantly smaller than the total training data points, we enhance the scalability of our model. For a deeper understanding and justification, please refer to Section 4.2.3 for the complexity analysis of our proposed model and Section 4.2.4.6 for the experiments on scalability investigation.

Let the hidden layer matrix is denoted by $\mathfrak{G}$, acquired through the random projection of the matrix of centers $\mathcal{O}$ of GBs followed by applying the activation function ψ, defined as:

$$\mathfrak{G} = \psi(\mathcal{O}\mathcal{X} + \mathbf{1}^t \otimes \zeta) \in \mathbb{R}^{k \times g}, \tag{4.1}$$

where $\mathcal{X} \in \mathbb{R}^{L \times g}$ is the randomly initialized weights matrix and $\zeta \in \mathbb{R}^{1 \times g}$ is the bias vector.

The output layer's weights are calculated as follows:

$$[\mathfrak{G} \oplus \mathcal{O}]\Omega = D\Omega = \hat{\mathcal{W}}. \tag{4.2}$$

Here, Ω in $\mathbb{R}^{(L+g)\times C}$ represents the weight matrix that links the input layer and hidden layer (GBs) to the output layer (GBs). $\hat{\mathcal{W}}$ is the anticipated output and $D = [\mathfrak{G} \oplus \mathcal{O}] \in \mathbb{R}^{k\times(L+g)}$.

The proposed optimization problem for the proposed GB-RVFL model is formulated as:

$$\begin{aligned} &\Omega_{\min} = \underset{\Omega}{\operatorname{argmin}} \frac{\mathcal{C}}{2}\|\phi\|_2^2 + \frac{1}{2}\|\Omega\|_2^2, \\ &\text{s.t. } D\Omega - \mathcal{W} = \phi, \end{aligned} \tag{4.3}$$

where ϕ refers to the error matrix and $\mathcal{C}$ is the regularization parameter. Problem (4.3) is the convex quadratic programming problem (QPP) and hence possesses a unique solution. The Lagrangian of (4.3) is written as:

$$\mathcal{L}(\Omega, \phi, \lambda) = \frac{\mathcal{C}}{2}\|\phi\|_2^2 + \frac{1}{2}\|\Omega\|_2^2 - \lambda^t(D\Omega - \mathcal{W} - \phi), \tag{4.4}$$

where λ is the Lagrangian multiplier. Differentiating $\mathcal{L}$ partially w.r.t. each parameters, $i.e.,\Omega, \phi$ and λ; and equating them to zero, we obtain

$$\frac{\partial \mathcal{L}}{\partial \Omega} = 0 \Rightarrow \Omega - D^t\lambda = 0 \Rightarrow \Omega = D^t\lambda, \tag{4.5}$$

$$\frac{\partial \mathcal{L}}{\partial \phi} = 0 \Rightarrow \mathcal{C}\phi + \lambda = 0 \Rightarrow \lambda = -\mathcal{C}\phi, \tag{4.6}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \Rightarrow D\Omega - \mathcal{W} - \phi = 0 \Rightarrow \phi = D\Omega - \mathcal{W}. \tag{4.7}$$

Substituting Eq. (4.7) in (4.6), we get

$$\lambda = -\mathcal{C}(D\Omega - \mathcal{W}). \tag{4.8}$$

On substituting the value of λ obtained in Eq. (4.5), we obtain

$$\Omega = D^t\left(-\mathcal{C}(D\Omega - \mathcal{W})\right), \tag{4.9}$$

$$\Rightarrow \Omega = -\mathcal{C}D^tD\Omega + \mathcal{C}D^t\mathcal{W}, \tag{4.10}$$

$$\Rightarrow \left(I + \mathcal{C}D^tD\right)\Omega = \mathcal{C}D^t\mathcal{W}, \tag{4.11}$$

$$\Rightarrow \Omega = \left(D^tD + \frac{1}{\mathcal{C}}I\right)^{-1}D^t\mathcal{W}, \tag{4.12}$$

where I is the identity matrix of the appropriate dimension. Substituting the values of (4.7) and (4.5) in (4.6), we get

$$\lambda = -\mathcal{C}(DD^t\lambda - \mathcal{W}), \tag{4.13}$$

$$\Rightarrow \lambda + \mathcal{C}DD^t\lambda = \mathcal{C}\mathcal{W}, \tag{4.14}$$

$$\Rightarrow \mathcal{C}\left(\frac{1}{\mathcal{C}}I + DD^t\right)\lambda = \mathcal{C}\mathcal{W}, \tag{4.15}$$

$$\Rightarrow \lambda = \left(\frac{1}{\mathcal{C}}I + DD^t\right)^{-1}\mathcal{W}. \tag{4.16}$$

Putting the obtained value of λ from Eq. (4.16) in (4.5), we get

$$\Omega = D^t\left(\frac{1}{\mathcal{C}}I + DD^t\right)^{-1}\mathcal{W}. \tag{4.17}$$

We get two distinct formulas, (4.12) and (4.17), that can be utilized to determine Ω. It is worth noting that both formulas involve the calculation of the matrix inverse. If the number of features $(L + g)$ in D is less than or equal to the number of GBs (k), we employ the formula (4.12) to compute Ω. Otherwise, we opt for the formula (4.17) to calculate Ω. As a result, we possess the advantage of calculating the matrix inverse either in the feature or sample space, contingent upon the specific scenario. Therefore, the optimal solution of (4.3)

is given as:

$$\Omega = \begin{cases} \left(D^t D + \frac{1}{\mathcal{C}} I\right)^{-1} D^t \mathcal{W}, & \text{if } (L+g) \leq k, \\ D^t \left(\frac{1}{\mathcal{C}} I + D D^t\right)^{-1} \mathcal{W}, & \text{if } k < (L+g). \end{cases} \tag{4.18}$$

4.2.2 Graph Embedded Granular Ball Random Vector Functional Link (GE-GB-RVFL) Network

This subsection delves into the proposed GE-GB-RVFL model, commencing with the establishment of its foundational mathematical framework. The GE-GB-RVFL model inherits all the properties of the proposed GB-RVFL model (i.e., robustness and scalability) along with preserving the inherent geometrical structure of GBs.

To preserve the intrinsic geometric structure of the granular balls (GBs), the proposed GE-GB-RVFL model incorporates the Graph Embedding (GE) framework into the learning process. Unlike the conventional RVFL model, which treats each training sample independently, graph embedding explicitly models the relationships among the GB centers by constructing graphs that characterize their intrinsic and discriminative structures. Consequently, the output weights are learned not only by minimizing the prediction error but also by preserving the local and global geometric relationships among the GB centers.

Let

$$D = [d_1, d_2, \ldots, d_M]^T \in \mathbb{R}^{M \times (L+g)}$$

denote the concatenated feature matrix of the M granular balls, where each row corresponds to the representation of a GB center after concatenating the original input attributes and the hidden-layer outputs. Two weighted graphs are constructed on D, namely the intrinsic graph

$$\mathcal{U}^{int} = \{D, \Theta^{int}\}$$

and the penalty graph

$$\mathcal{U}^{pen} = \{D, \Theta^{pen}\}.$$

The intrinsic graph preserves the similarity among GB centers belonging to the same class, whereas the penalty graph enhances class separability by penalizing undesirable relationships between different classes.

The similarity matrices

$$\Theta^{int}, \Theta^{pen} \in \mathbb{R}^{M \times M}$$

encode the pairwise relationships among GB centers. Specifically, the (i, j)-th entry of Θ^{int} represents the intrinsic similarity between the i-th and j-th GB centers, while the corresponding entry of Θ^{pen} quantifies the penalty imposed on that pair. Based on these similarity matrices, the corresponding degree matrices are constructed as

$$F_{ii} = \sum_{j=1}^{M} \Theta_{ij}^{int},$$

and

$$F_{ii}^{pen} = \sum_{j=1}^{M} \Theta_{ij}^{pen},$$

where F and F^{pen} are diagonal matrices.

The intrinsic and penalty graph Laplacians are subsequently defined as

$$\mathcal{L} = F - \Theta^{int},$$

and

$$\mathcal{S} = F^{pen} - \Theta^{pen},$$

respectively. The intrinsic Laplacian encourages neighboring GB centers to remain close after projection, whereas the penalty Laplacian enlarges the distance between GB centers

belonging to different classes, thereby improving class discrimination.

Following the general graph embedding framework [5], the projection matrix is obtained by minimizing the intrinsic graph energy while constraining the penalty graph energy, resulting in the generalized eigenvalue problem

$$U_{int}s = \lambda U_{pen}s,$$

where

$$U_{int} = D^T \mathcal{L} D,$$

and

$$U_{pen} = D^T \mathcal{S} D.$$

Accordingly, the unified graph embedding matrix is defined as

$$U_{uni} = U_{pen}^{-1} U_{int},$$

which simultaneously captures the intrinsic neighborhood relationships and the discriminative information among the GB centers. This unified graph matrix serves as the graph regularizer in the proposed GE-GB-RVFL optimization problem, ensuring that the learned output weights preserve the original geometric structure of the granular balls while maintaining strong discriminative capability.

- ***Preservation of the original geometrical structure of granular balls:*** The preservation of the original geometrical structure of GBs is achieved through the optimization process (see Eq. 4.19) for determining the GE-GB-RVFL network's output weights. This process incorporates subspace learning (SL) criteria, utilizing intrinsic and penalty SL within the GE and granular computation frameworks. The GE

framework adeptly manages the geometric relationships among the centers of GBs. In contrast, the standard RVFL model lacks this capability, leading to a failure in preserving the original geometrical structure of datasets and, thus, losing valuable information during training.

The proposed optimization problem of the GE-GB-RVFL is articulated as follows:

$$\begin{aligned} &\min_{\Omega} \ \frac{1}{2}\|\Omega\|_2^2 + \frac{\mathcal{C}}{2}\|\xi\|_2^2 + \frac{\alpha}{2}\|U^{\frac{1}{2}}\Omega\|_2^2 \\ &s.t. \ \ D\Omega - \mathcal{W} = \xi. \end{aligned} \tag{4.19}$$

Here, Ω denotes the weights matrix that establishes connections between the input layer and hidden layer (GBs) and the output layer (GBs), ξ is the error matrix, and $\mathcal{C}$ is the regularization parameter. Eq. (4.19) assigns distinct significance to each center of the GBs by incorporating weights and considering the geometric relationships among GBs by including the GE term.

Drawing from the literature, the weights for intrinsic and penalty graphs in the context of linear discriminant analysis (LDA) [185] are given as follows:

$$\Theta_{ij}^{int} = \begin{cases} \frac{1}{G_{w_i}}, & \text{if } \ w_i = w_j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.20}$$

$$\Theta_{ij}^{pen} = \begin{cases} \frac{1}{k} - \frac{1}{G_{w_i}}, & \text{if } \ w_i = w_j, \\ \frac{1}{k}, & \text{otherwise.} \end{cases} \tag{4.21}$$

Here, G_{w_i} denotes the number of GBs (centers) with class label w_i.

The Lagrangian of (4.19) is written as:

$$\mathcal{L}_{GE} = \frac{1}{2}\|\Omega\|_2^2 + \frac{\mathcal{C}}{2}\|D\Omega - \mathcal{W}\|_2^2 + \frac{\alpha}{2}\|U_{uni}^{\frac{1}{2}}\Omega\|_2^2. \tag{4.22}$$

Applying the Karush-Kuhn-Tucker (K.K.T.) condition in (4.22), we get following:

$$\frac{\partial \mathcal{L}_{GE}}{\partial \Omega} = \mathcal{C}D^t(D\Omega - \mathcal{W}) + \Omega + \alpha U_{uni}\Omega = 0. \tag{4.23}$$

Upon computation, the output layer parameters are obtained as follows:

$$\Omega = \left(D^tD + \frac{1}{\mathcal{C}}I + \frac{\alpha}{\mathcal{C}}U_{uni}\right)^{-1} D^t\mathcal{W}. \tag{4.24}$$

The matrix U_{uni} can be defined in two cases:

[1] Case 1: When U_{uni} represents intrinsic training data relationships ($U_{pen} = I$), it indicates that no penalties are imposed on relationships among the nodes (GB centers). This approach solely relies on intrinsic relationships to extract graphical information from the GBs.

[2] Case 2: When U_{uni} encompasses both intrinsic and penalty training data relationships ($U_{pen} \neq I$), specific penalties are applied to certain relationships among the nodes (GB centers). Here, the emphasis is on utilizing both intrinsic and penalty-specific relationships to extract graphical information from the GBs.

Remarks

[1] In our proposed GE-GB-RVFL model, the embedding space of the Graph U_{uni} is in the GB-RVFL space $\mathbb{R}^{(L+g)}$, rather than the input space $\mathbb{R}^L$, provides a more accurate representation of both linear and nonlinear relationships among the nodes/centers of GBs.

[2] Incorporating graph embedding (GE) increases the complexity of the proposed GE-GB-RVFL model. Therefore, future research could explore alternative methods, such

as sparse GE techniques, to preserve the geometrical structure of datasets while reducing computational overhead.

[3] In our experiments, we used LDA for the graph embedding, as it is particularly effective for maximizing class separability in supervised learning tasks. However, other techniques such as, local Fisher discriminant analysis (LFDA) or marginal Fisher analysis (MFA) [36], can also be explored, depending on the task requirements.

The algorithms of the proposed GB-RVFL and GE-GB-RVFL models are given in Algorithm 4.2.

4.2.3 Time and Space Complexity Analysis of the Proposed Models

Here, we discuss the time and space complexity of the proposed GB-RVFL and GE-GB-RVFL models.

Time Complexity of GB-RVFL: The complexity of the proposed GB-RVFL model primarily depends on (a) the requirement of matrix inversion in (4.18), and (b) GB computation. Following [2], time complexity in computing inverse in RVFL is $\mathcal{O}(M^3)$ if $M \leq (L+g)$ or $\mathcal{O}((L+g)^3)$ if $(L+g) < M$. Therefore, the time complexity to find the inverse in (4.18) is $\mathcal{O}(k^3)$ if $k \leq (L+g)$ or $\mathcal{O}((L+g)^3)$ if $(L+g) < k$. In the generation of GB, we use 2-means clustering; therefore, in 2-means clustering, the time complexity is $\mathcal{O}(2M(iter))$ [186], where M represents the number of samples in the training dataset, and $iter$ is the number of iterations. Our approach starts with the training dataset, which we treat as the initial granular ball (GB) set. Using the 2-means clustering method, we divide GB into two GBs initially, with a computational complexity of $\mathcal{O}(2M)$. In subsequent phases, if both GBs are impure, they undergo further division into four GBs, maintaining a maximum computational complexity of $\mathcal{O}(2M)$ each time. This iterative process continues for a total of $iter$ iterations. The overall computational complexity of generating GBs is

Algorithm 4.2 Algorithm of the proposed GB-RVFL and GE-GB-RVFL models.

Input: training dataset (denoted as $\mathcal{T} = (U, T)$), and the threshold purity ρ.
Output: Output weights of the GB-RVFL.

1: Assume the entire dataset $\mathcal{T}$ as a granular ball GB and set of GBs, G, to be empty set, i.e., $GB = \mathcal{T}$ and $G = \{\ \}$.
2: $Temp = \{GB\}$.
3: $for\ i = 1 : |Temp|$
4: $if\ pur(GB_i) < \rho$
5: Split GB_i into GB_{i1} and GB_{i2}, using 2-means clustering algorithm.
6: $Temp \leftarrow GB_{i1}, GB_{i2}$.
7: $end\ if$.
8: $else\ pur(GB_i) \geq \rho$
9: Calculate the center $o_i = \frac{1}{p}\sum_{j=1}^{p} v_j$ of GB_i, where $v_j \in GB_i$, $j = 1, 2, \ldots, p$, and p is the number of training sample in GB_i.
10: Calculate the label w_i of GB_i, where w_i is assigned the label of majority class samples within GB_i.
11: Put $GB_i = \{(o_i, w_i)\}$ in G.
12: $end\ else$.
13: $end\ for$.
14: $if\ Temp \neq \{\ \}$
15: Go to step 3 (for further splitting).
16: $end\ if$.
17: Set $G = \{GB_i,\ \ i = 1, 2, \ldots, k\} = \{(o_i, w_i),\ \ i = 1, 2, \ldots, k\}$, where o_i signifies the center of the GB, w_i is the label of GB_i and k is the number of generated GBs.
18: Find the hidden layer features using (4.1).
19: Create the enhanced features using (4.2).
20: **For the output weights of GB-RVFL model:** Calculate the output layer weights using (4.18).
21: **For the output weights of GE-GB-RVFL model:** Calculate intrinsic and penalty graphs using (5.24) and (5.25), respectively, then the output layer weights are calculated using (4.24).

$\mathcal{O}(2M(iter))$ or less, considering the maximum computational complexity per iteration and the total number of iterations $iter$. Thus, the time complexity of the proposed GB-RVFL is $\mathcal{O}(k^3)$+$\mathcal{O}(2M(iter))$ if $k \leq (L + g)$ or $\mathcal{O}((L + g)^3)$+$\mathcal{O}(2M(iter))$ if $(L + g) < k$. Since, $k << M$, therefore, $\mathcal{O}(k^3) << \mathcal{O}(M^3)$. This implies that if the inverse of the matrix is calculated in the sample space, then the time complexity of the proposed GB-RVFL is much less than the RVFL model.

Time Complexity of GE-GB-RVFL: Additionally, the time complexity of the proposed

GE-GB-RVFL depends upon one more factor, which is the computation of the GE matrix U_{uni}. This matrix accounts for both intrinsic and penalty graph structures. According to [39], the time complexity for this computation is $\mathcal{O}((L+g)^3 + (L+g)^2 k)$. Thus, the time complexity of the proposed GB-RVFL is $\mathcal{O}(k^3)$+$\mathcal{O}(2M(iter))$+$\mathcal{O}((L+g)^3 + (L+g)^2 k)$ if $k \leq (L+g)$ or $\mathcal{O}((L+g)^3)$+$\mathcal{O}(2M(iter))$+$\mathcal{O}((L+g)^3 + (L+g)^2 k)$ if $(L+g) < k$.

Space complexity of GB-RVFL: The space complexity of the GB-RVFL model can be understood by considering the storage requirements for its components. First, the model requires space to store the centers of the GBs, which involves k GBs, each with L features, resulting in a complexity of $\mathcal{O}(kL)$. Additionally, the hidden layer weights matrix, involving g hidden nodes and P input features, contributes $\mathcal{O}(Lg)$. Furthermore, the inversion of the GB center matrix, which is of size $k \times g$, requires $\mathcal{O}(kg)$. Therefore, the overall space complexity of the GB-RVFL model is $\mathcal{O}(kL + Lg)$.

Space complexity of GE-GB-RVFL: For the proposed GE-GB-RVFL model, which extends GB-RVFL by incorporating graph embedding (GE), additional storage is required. The graph embedding involves an adjacency matrix to capture relationships between data points of $L + g$ features of the matrix D, leading to a potential space complexity of $\mathcal{O}((L+g)^2)$. Consequently, the total space complexity of the GE-GB-RVFL model is $\mathcal{O}(kL + Lg)$ + $\mathcal{O}((L+g)^2)$.

4.2.4 Numerical Experiments and Results

This section presents comprehensive details of the experimental setup, datasets, and compared models. Subsequently, we delve into the experimental results and conduct statistical analyses. We also examine the influence of noise on the performance of the proposed GB-RVFL and GE-GB-RVFL models.

4.2.4.1 Datasets

To evaluate the efficacy of the proposed GB-RVFL and GE-GB-RVFL models, we employ 30 benchmark datasets from UCI [7] and KEEL [8]. To test the scalability of the proposed model, we use NDC [187] dataset, generated using David Musicant's NDC Data Generator, encompasses samples ranging from 50 thousand to 100 million, consistently featuring 32 features. Moreover, as an application, we employ our proposed models to classify the various stages of Alzheimer's disease using the ADNI[1] dataset and breast cancer detection using the BreakHis dataset [181].

4.2.4.2 Compared Models

We conduct comparisons among the proposed GB-RVFL and GE-GB-RVFL; and several benchmarks, including the standard RVFL [1], RVFL without direct link (RVFLwoDL) (also known as ELM) [22], Intuitionistic Fuzzy RVFL (IF-RVFL) [32], Neuro-fuzzy RVFL (NF-RVFL) [17], and RVFL based on Wave loss function (Wave-RVFL) [72]. Further, we formulate and propose GB-RVFLwoDL (granular ball RVFL without direct link) to test the significance of direct links. The formulation of GB-RVFLwoDL is the same as GB-RVFL, except the direct links connection is missing in GB-RVFLwoDL.

4.2.4.3 Experimental Setup and Hyperparameter Setting

The experimental configuration includes a PC equipped with an Intel(R) Xeon(R) Gold 6226R CPU running at a speed of 2.90GHz and featuring 128 GB of RAM. This system runs on the Windows 11 platform and executes tasks using Python 3.11. The dataset is randomly divided into training and testing subsets at a ratio of $70 : 30$, respectively. We utilize a five-fold cross-validation technique combined with a grid search approach to optimize the

[1] https://adni.loni.usc.edu/

hyperparameters of the models within specified ranges: $\mathcal{C} = \alpha = \{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. The number of hidden nodes is selected within the range $3 : 20 : 203$. Following [17], for the NF-RVFL model, the neuro-fuzzy layer's fuzzy rules are chosen from the range $J = 5 : 5 : 50$ and k-means clustering is used to generate centres of the neuro-fuzzy layer. μ for the membership function in the IFRVFL model is chosen from the set $\{2^{-5}, 2^{-3}, \ldots, 2^{5}\}$. The Wave loss parameters used for the Wave-RVFL model are the same as those in [72]. For all models, we tune 10 activation functions with indices ranging from 1 to 10. The details are provided in Table 4.1.

Table 4.1: Index of activation functions.

Index	Activation Functions
1	Scaled Exponential Linear Unit (SELU)
2	Rectified Linear Units (ReLU)
3	Sigmoid
4	Sine (Sin)
5	Hard Limit Transfer Function (Hardlim)
6	Triangular Basis Transfer Function (Tribas)
7	Radial Basis Transfer Function (Radbas)
8	Signum (Sgn)
9	Leaky Rectified Linear Unit, or (Leaky ReLU)
10	Hyperbolic Tangent Sigmoid Transfer Function (Tansig)

4.2.4.4 Results and Discussions on UCI and KEEL Datasets

The proposed GB-RVFL and GE-GB-RVFL models are evaluated against standard RVFL, RVFLwoDL, GB-RVFLwoDL, IF-RVFL, Wave-RVFL and NF-RVFL models. The accuracy and rank of the models are reported in Table 4.2. We compare the performance of the proposed models using accuracy and statistical tests (following recommendations of Demšar [153]), including the ranking, Friedman, and Nemenyi post hoc tests.

Accuracy

As per Table 4.2, the proposed GE-GB-RVFL achieved the highest average accuracy at 85.03%, with the proposed GB-RVFL following closely at 84.44%. The number of fuzzy-based models, i.e., IF-RVFL and NF-RVFL, as well as RVFL, RVFLwoDL, Wave-RVFL, and GB-RVFLwoDL models, comes later than the proposed models. This finding suggests that the proposed GB-RVFL and GE-GB-RVFL models mitigate the detrimental effect of noise and outliers and simultaneously leverage the original topological structure of the center of the GBs.

Table 4.2: Accuracy and rank of the proposed GB-RVFL and GE-GB-RVFL models against the baseline models on real-world datasets, i.e., KEEL and UCI.

Model →	RVFL [1]	RVFLwoDL [22]	IF-RVFL [32]	NF-RVFL [17]	Wave-RVFL [72]	GB-RVFLwoDL★	GB-RVFL★	GE-GB-RVFL★
Dataset ↓	(ACC, Rank)	(ACC, Rank)	(ACC, Rank)	(ACC, Rank)	(ACC, Rank)	(ACC, Rank)	(ACC, Rank)	(ACC, Rank)
acute_inflammation	(100, 3)	(100, 3)	(83.33, 6)	(75.56, 7)	(100, 3)	(75, 8)	(100, 3)	(100, 3)
aus	(86.94, 6)	(87.98, 1.5)	(86.54, 8)	(87.98, 3)	(87.98, 1.5)	(87.02, 4.5)	(87.02, 4.5)	(86.54, 7)
bank	(89.17, 3)	(89.31, 2)	(88.14, 8)	(88.95, 4)	(88.52, 6)	(88.87, 5)	(88.43, 7)	(89.83, 1)
chess_krvkp	(90.41, 4)	(90.2, 5)	(90.62, 3)	(80.95, 8)	(87, 7)	(88.95, 6)	(91.45, 2)	(92.49, 1)
cleve	(81.11, 3)	(80, 4)	(78.89, 5)	(82.22, 1)	(75.46, 8)	(76.67, 6)	(75.56, 7)	(82.22, 2)
conn_bench_sonar_mines_rocks	(74.6, 3.5)	(71.43, 7)	(70.95, 8)	(74.6, 6)	(74.63, 1)	(74.6, 3.5)	(74.6, 3.5)	(74.6, 3.5)
crossplane130	(97.44, 6.5)	(97.44, 6.5)	(94.87, 8)	(100, 3)	(100, 3)	(100, 3)	(100, 3)	(100, 3)
echocardiogram	(85, 6)	(87.5, 2.5)	(81, 7)	(80, 8)	(86.5, 5)	(87.5, 2.5)	(87.5, 2.5)	(87.5, 2.5)
ecoli-0-1-4-6_vs_5	(98.81, 4.5)	(98.81, 4.5)	(98.81, 2)	(94.05, 8)	(97.93, 7)	(100, 1)	(98.81, 4.5)	(98.81, 4.5)
ecoli0137vs26	(85.74, 7)	(86.81, 5)	(86.62, 6)	(88.68, 2)	(87.18, 4)	(58.51, 8)	(87.23, 3)	(89.36, 1)
ecoli2	(82.08, 7)	(81.09, 8)	(85.12, 5.5)	(85.12, 5.5)	(86.38, 3)	(85.15, 4)	(87.13, 1.5)	(87.13, 1.5)
fertility	(90, 2)	(80, 7)	(83.33, 6)	(73.33, 8)	(86.33, 5)	(90, 2)	(86.67, 4)	(90, 2)
haberman_survival	(76.09, 5)	(78.26, 3)	(73.91, 7)	(71.74, 8)	(74.47, 6)	(78.26, 3)	(78.26, 3)	(79.35, 1)
heart-stat	(81.89, 3.5)	(81.89, 3.5)	(79.78, 5.5)	(79.78, 5.5)	(72.19, 8)	(77.78, 7)	(82.72, 1)	(82.72, 2)
heart_hungarian	(72.78, 7)	(72.65, 8)	(77.53, 3)	(87.65, 2)	(96.61, 1)	(73.03, 6)	(74.16, 5)	(75.28, 4)
ionosphere	(82.79, 5)	(82.45, 6)	(87.74, 1.5)	(87.74, 1.5)	(74, 8)	(75.47, 7)	(83.96, 3)	(83.02, 4)
led7digit-0-2-4-5-6-7-8-9_vs_1	(94.74, 3.5)	(94.74, 3.5)	(86.47, 8)	(92.48, 7)	(92.85, 6)	(95.49, 1)	(93.99, 5)	(94.99, 2)
mammographic	(82.01, 4)	(82.35, 3)	(81.66, 6)	(80.28, 7.5)	(81.97, 5)	(80.28, 7.5)	(83.74, 1)	(83.39, 2)
monk1	(42.52, 6)	(42.1, 7)	(49.1, 1)	(44.31, 5)	(44.31, 3.5)	(41.32, 8)	(44.31, 3.5)	(47.9, 2)
monks_3	(90.41, 2.5)	(90.41, 2.5)	(87.9, 4)	(85.63, 6.5)	(85.63, 6.5)	(85.63, 6.5)	(85.63, 6.5)	(91.02, 1)
new-thyroid1	(90, 5.5)	(90, 5.5)	(98.46, 2)	(100, 1)	(89.44, 7)	(70.77, 8)	(90.77, 4)	(93.85, 3)
pima	(74.03, 5)	(73.59, 6)	(75.76, 4)	(76.19, 3)	(70.8, 8)	(71.86, 7)	(79.7, 1)	(76.62, 2)
statlog_heart	(80.89, 5.5)	(80.89, 5.5)	(81.36, 4)	(87.65, 1)	(80.37, 7)	(77.78, 8)	(82.72, 2)	(82.72, 3)
tic_tac_toe	(96.65, 7)	(96.65, 7)	(99.65, 1.5)	(99.65, 1.5)	(96.65, 7)	(97.22, 5)	(97.57, 4)	(99.31, 3)
transfusion	(76.44, 2)	(71.11, 8)	(74.56, 7)	(79.56, 1)	(74.89, 6)	(75.11, 5)	(76, 3.5)	(76, 3.5)
vehicle2	(90.85, 6)	(90.03, 8)	(97.64, 1.5)	(97.64, 1.5)	(90.64, 7)	(91.34, 5)	(91.73, 4)	(92.52, 3)
vertebral_column_2clases	(81.4, 7)	(82.25, 6)	(87.1, 3)	(84.95, 4)	(94.84, 1)	(82.37, 5)	(74.19, 8)	(90.32, 2)
wpbc	(69.49, 5.5)	(70.97, 4)	(76.27, 3)	(79.66, 1)	(68.36, 7)	(62.71, 8)	(76.27, 2)	(69.49, 5.5)
yeast-0-2-5-7-9_vs_3-6-8	(95.35, 6)	(95.68, 5)	(96.03, 2)	(95.7, 4)	(95.9, 3)	(94.37, 7)	(98.01, 1)	(77.81, 8)
yeast1	(74.01, 8)	(74.78, 6.5)	(75.34, 3)	(74.78, 6.5)	(75.09, 5)	(75.78, 2)	(75.11, 4)	(76.23, 1)
Average (ACC, Rank)	(83.79, 4.98)	(83.38, 5.13)	(83.82, 4.65)	(83.89, 4.37)	(83.9, 5.18)	(80.63, 5.32)	(84.44, 3.57)	(**85.03**, **2.8**)

The best-performing model in terms of average accuracy is indicated by boldface in the last row. ACC is an acronym for accuracy. ★ denotes the proposed models.

Statistical rank

Following the Statistical tests in Section 2.11, Table 4.2 presents the ranking of models across all datasets. The proposed GE-GB-RVFL and GB-RVFL models obtain the lowest

Table 4.3: Comparison of the proposed GE-GB-RVFL and other compared models using Nemenyi post hoc test.

	RVFL [1]	RVFLwoDL [22]	IF-RVFL [32]	NF-RVFL [17]	Wave-RVFL [72]	GB-RVFLwoDL*
Rank Difference	2.18	2.33	1.85	1.57	2.38	2.52
Significance	Yes	Yes	Yes	No	Yes	Yes

The first row presents the difference in ranking between the proposed GE-GB-RVFL model and the models listed in the respective columns, whereas the second row shows the significant superiority of the proposed GE-GB-RVFL model listed in the respective columns. * denotes the proposed model.

average ranks of 2.8 and 3.57, respectively, securing the top two positions among all models. This indicates the strong generalization capability of the proposed approaches across the datasets. In contrast, GB-RVFLwoDL obtains the highest average rank of 5.32, demonstrating comparatively weaker performance. These results highlight the effectiveness of the proposed models and also emphasize the importance of direct links in RVFL-based architectures.

Friedman test

For $\mathfrak{M} = 8$ models and $\mathfrak{D} = 30$ datasets, the Friedman statistics are obtained as $\chi_F^2 = 27.78$ and $F_F = 4.42$. At the 5% significance level, the critical value is $F_F(7, 203) = 2.05$. Since $4.42 > 2.05$, the null hypothesis is rejected, confirming that statistically significant differences exist among the compared models.

Nemenyi post hoc test

The Nemenyi post hoc test is further applied with a critical difference of $C.D. = 1.76$ at the 0.1 significance level. The rank differences between GE-GB-RVFL and RVFL, RVFLwoDL, IF-RVFL, Wave-RVFL, and GB-RVFLwoDL are 2.18, 2.33, 1.85, 2.38, and 2.52, respectively, all exceeding the $C.D.$ value. This confirms that GE-GB-RVFL significantly outperforms these models. Although GE-GB-RVFL is not statistically superior to NF-RVFL, its lower average rank indicates better overall performance. These results demonstrate that the use of GBs as inputs contributes to the improved generalization perfor-

mance of the proposed GB-RVFL and GE-GB-RVFL models.

Direct link significance

Notably, the rank difference between GB-RVFL and GB-RVFLwoDL is 1.75; this affirms the significance of direct links in the model's generalization performance.

Main takeaways from the above discussions:

- The proposed GB-RVFL and GE-GB-RVFL models excel in both average accuracy and ranks.
- The superior accuracy and rank of GB-RVFL over GB-RVFLwoDL indicates that the direct links significantly contribute to the model's performance.
- Statistical analyses, including the Friedman and Nemenyi post hoc test, validate the statistical superiority of the proposed models compared to the existing baseline models.

4.2.4.5 Results and Discussions on UCI and KEEL Datasets in Noisy Environment

We contaminate the label noise in the UCI and KEEL datasets at values of 5%, 10%, 20%, 30%, and 40% to disturb the label of the samples in order to evaluate how robust the proposed GB-RVFL and GE-GB-RVFL models are against noise. In total, we take 7 datasets for this experiment.

Comparision among proposed, RVFL and RVFLwoDL models

First of all, we compare our proposed models with baseline models, i.e., RVFL and RVFLwoDL using 4 datasets to test the robustness of the models. Among the 4 datasets, two datasets (ecoli-0-1-4-6_vs_5 and yeast-0-2-5-7-9_vs_3-6-8) are chosen in such a manner

that the proposed GE-GB-RVFL model does not top at 0% noise. In the other two datasets, i.e., statlog_heart and conn_bench_sonar_mines_rocks, the proposed GE-GB-RVFL model tops at 0% noise.

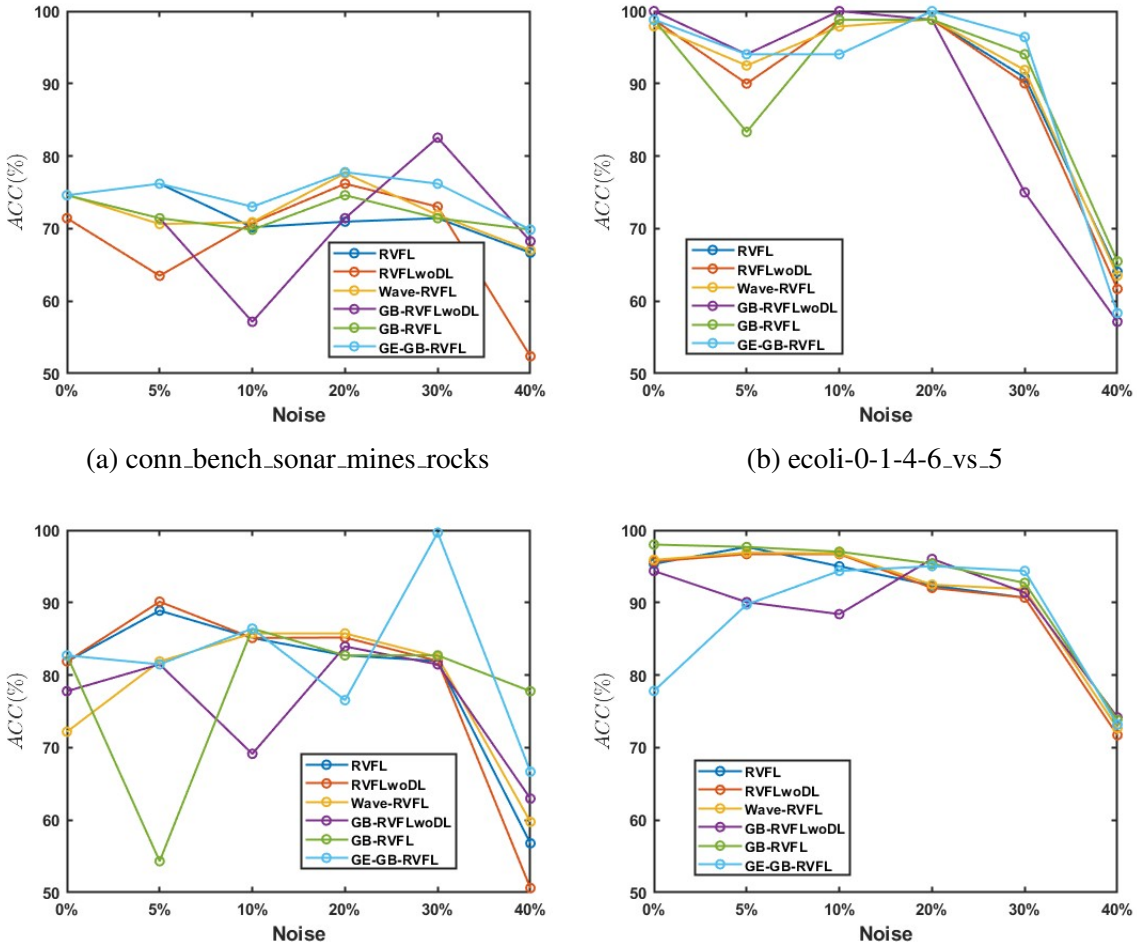
Table 4.4: Classification accuracies of the proposed GB-RVFL and GE-GB-RVFL models against the baseline models on UCI and KEEL datasets with contaminated label noise.

Model →		RVFL [1]	RVFLwoDL [22]	Wave-RVFL [72]	GB-RVFLwoDL★	GB-RVFL★	GE-GB-RVFL★
Dataset ↓	Noise	ACC	ACC	ACC	ACC	ACC	ACC
conn_bench_sonar_mines_rocks	5%	76.1905	63.4921	70.63	71.4286	71.4286	76.1905
	10%	70.1905	70.7778	70.88	57.1429	69.8413	73.0159
	20%	70.9524	76.1905	77.63	71.4286	74.6032	77.7778
	30%	71.4286	73.0159	71.88	82.5397	71.4286	76.1905
	40%	66.6667	52.381	67	68.254	69.8413	69.8413
Average		71.0857	67.1714	71.604	70.1587	71.4286	**74.6032**
ecoli-0-1-4-6_vs_5	5%	90	90	92.5	94.0476	83.3333	94.0476
	10%	98.8095	98.8095	97.88	100	98.8095	94.0476
	20%	98.8095	98.8095	98.88	98.8095	98.8095	100
	30%	90.8571	90.0476	91.88	75	94.0476	96.4286
	40%	64.0476	61.6667	63.5	57.1429	65.4762	58.3333
Average		88.5048	87.8667	88.928	85	88.0952	**88.5714**
statlog_heart	5%	88.8889	90.1235	81.88	81.4815	84.321	81.4815
	10%	85.1235	85.1235	85.75	69.1358	86.4198	86.4198
	20%	82.716	85.1852	85.75	83.9506	82.716	76.5432
	30%	81.9506	81.9506	82.5	81.4815	82.716	99.6528
	40%	56.7901	50.6173	59.75	62.963	77.7778	66.6667
Average		79.0938	78.6	79.126	75.8025	**82.7901**	82.1528
yeast-0-2-5-7-9_vs_3-6-8	5%	97.6821	96.6887	96.89	90.0662	97.6821	89.7351
	10%	95.0132	96.6887	96.78	88.4106	97.0199	94.3709
	20%	92.3576	92.0265	92.48	96.0265	95.3642	95.0331
	30%	90.6887	90.6887	91.84	91.3907	92.7152	94.3709
	40%	71.7219	71.7219	72.68	74.1722	73.8411	73.1788
Average		89.4927	89.5629	90.134	88.0132	**91.3245**	89.3377
Overall Average ACC		84.015	82.7865	82.848	81.5653	**85.7265**	85.2408

The best-performing model in terms of accuracy is indicated by boldface in each row. ACC is an acronym for accuracy.
★ denotes the proposed models.

Table 4.4 and Fig. 4.1 present experimental results and graphs, respectively, across different noise levels. The proposed GB-RVFL and GE-GB-RVFL models consistently outperform baseline models, showcasing superior accuracy. In each noise level, the proposed GB-RVFL and GE-GB-RVFL emerge as best performers, achieving average accuracies (overall) of 85.7265 and 85.2408, respectively.

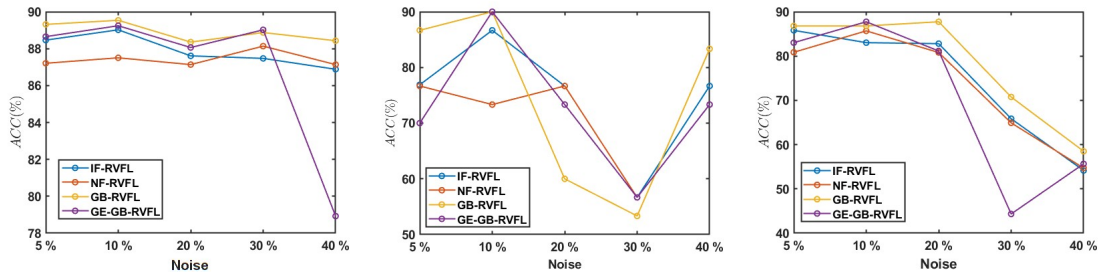
Compared with the proposed models, the GB-RVFLwoDL model lags with an overall average accuracy of 81.5653. This indicates a significant performance gap of approximately 4% between the GB-RVFLwoDL and the proposed models. This emphasizes the advanced



(a) conn_bench_sonar_mines_rocks (b) ecoli-0-1-4-6_vs_5

(c) statlog_heart (d) yeast-0-2-5-7-9_vs_3-6-8

Figure 4.1: Effect of different labels of noise on the performance of the proposed GB-RVFL and GE-GB-RVFL models.



(a) bank (b) fertility (c) ionosphere

Figure 4.2: Effect of different labels of noise on the performance of the proposed GB-RVFL and GE-GB-RVFL model with the baseline IF-RVFL and NF-RVFL models.

noise resilience exhibited by the proposed GB-RVFL and GE-GB-RVFL models. Additionally, the presence of at least one proposed model (sometimes both) among the top two performers in each dataset underscores the robustness of these models due to the fusion of GB computation in handling noise.

Comparision among proposed and fuzzy based models

The robustness of fuzzy-based models is widely acknowledged [32], prompting us to evaluate the robustness of our proposed models against state-of-the-art baseline robust models, namely IF-RVFL and NF-RVFL. We conduct our analysis using three datasets: bank, fertility, and ionosphere, with corresponding graphs illustrated in Fig. 4.2. Notably, our findings consistently demonstrate the robust performance of the proposed GB-RVFL models compared to baseline models. Specifically, when subjected to increasing noise percentage levels in the bank dataset, the proposed GB-RVFL model maintains its accuracy, displaying insensitivity to noise fluctuations. Similarly, for the fertility and ionosphere datasets, the GB-RVFL models consistently exhibit the highest accuracy at a noise label of 40%, highlighting their superior robustness over state-of-the-art baseline models. The competitive performance of the GE-GB-RVFL model raises intriguing possibilities. One potential explanation for this phenomenon could be that while preserving the intrinsic geometrical structure of the datasets, the model may, to some extent, sacrifice a portion of its robustness. This delicate balance between structural fidelity and robustness could contribute to the GE-GB-RVFL model's ability to achieve competitive results.

Thus, by utilizing the coarse nature of GBs and focusing on their centers, the proposed GB-RVFL and GE-GB-RVFL models effectively capture the core information while reducing susceptibility to noise located further from central data distribution or clusters.

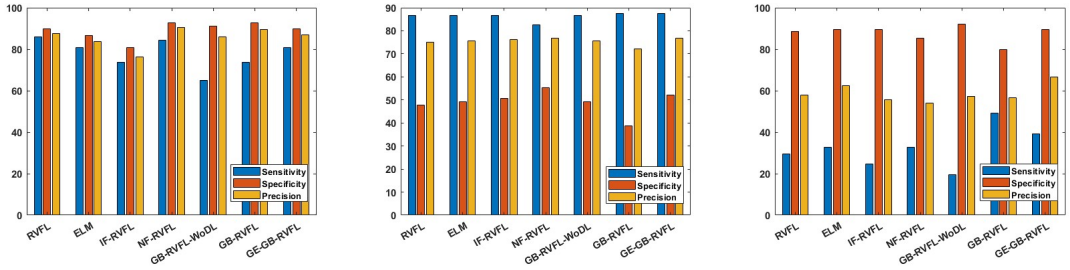
Table 4.5: Accuracies of the proposed models versus the baseline models on NDC datasets.

Model →	RVFL [1]	RVFLwoDL [22]	GB-RVFLwoDL★	GB-RVFL★	GE-GB-RVFL★
Dataset ↓	ACC	ACC	ACC	ACC	ACC
NDC-50K	80.9435	80.1134	82.1687	84.4884	**85.3785**
NDC-100K	78.2968	78.6267	88.2494	**88.4044**	83.8742
NDC-500K	82.3814	81.1474	80.7758	83.2042	**85.3393**
NDC-1M	81.6562	81.5717	83.1739	85.2399	**86.1404**
NDC-3M	80.8727	80.4434	80.7375	82.6701	**83.9588**
NDC-5M	84.0435	83.608	83.9787	84.8743	**85.3805**
NDC-10M	82.8764	82.7896	86.1019	**86.6179**	85.7042
NDC-30M	a	a	82.6621	83.4257	**83.4441**
NDC-50M	a	a	80.9622	83.5944	**83.7941**
NDC-100M	a	a	80.9646	84.4071	**85.0702**
Average ACC	81.5815	81.1857	82.9775	84.6926	**84.8084**

The best-performing model in terms of accuracy is indicated by boldface in each row.
"a" denotes the code terminated due to out of memory. ACC is an acronym for accuracy.
★ denotes the proposed models.

4.2.4.6 Results and Discussions on NDC Dataset

We conducted extensive experiments on large datasets generated using David Musicant's NDC Data Generator [187], ranging from 50 thousand to 100 million data samples. For example, NDC-50K and NDC-1M indicate that the dataset comprises $50,000$ and 1 million data samples, respectively. The results in Table 4.5 provide a comprehensive view of the performance of the proposed GB-RVFL and GE-GB-RVFL models in comparison to the baseline RVFL and RVFLwoDL models. Analysis of Table 4.5 reveals that the GE-GB-RVFL model outperforms as the top performer, achieving the highest average accuracy at 84.8084%, followed closely by GB-RVFL with 84.6926% average accuracy. Notably, RVFL and RVFLwoDL encounter memory issues with larger datasets (e.g., NDC-30M, NDC-50M and NDC-100M), while our proposed GB-based models handle these datasets seamlessly. This emphasizes the adept adaptation and leveraging of the GB and GE frameworks by our proposed GB-RVFL and GE-GB-RVFL models. Furthermore, it attests to the scalability and efficacy exhibited by proposed GB-RVFL and GE-GB-RVFL models, demonstrating a



(a) CN_vs_AD (b) CN_vs_MCI (c) MCI_vs_AD

Figure 4.3: Performance comparison of the proposed GB-RVFL and GE-GB-RVFL models against baseline models on ADNI data using sensitivity, specificity, and precision.

notable capacity to generalize effectively, especially when confronted with large and intricate datasets such as NDC. Scalability in the proposed GB-RVFL and GE-GBRVFL models stems from their training process, which employs GBs instead of the entire dataset. Because the number of GBs is much smaller than the total data points, the models' scalability is improved.

4.2.4.7 Application on Biomedical Domain

To show the applicability of the proposed GB-RVFL and GE-GB-RVFL models in real-world scenarios, we employ them in biomedical datasets, namely, the BreakHis dataset (for breast cancer classification) and ADNI datasets (for AD classification). To thoroughly evaluate the effectiveness of the proposed GB-RVFL and GE-GB-RVFL models, we assess their performance across multiple metrics, including Accuracy (ACC), Sensitivity, Specificity, and Precision.

- ***BreakHis dataset***: BreakHis dataset comprises histopathology scans, as detailed in reference [181]. Our study included 1240 scans, all at 400 times magnification. These scans primarily fell into two categories: malignant and benign. Among the malignant scans, there were subclasses such as lobular carcinoma, ductal carcinoma, papillary carcinoma and mucinous carcinoma with respective scan counts of 137, 208, 138, and

Table 4.6: Accuracies of the proposed GB-RVFL and GE-GB-RVFL models against the baseline models on Biomedical datasets.

Dataset	RVFL [1] (ACC, Sensitivity) (Specificity, Precision)	RVFLwoDL [22] (ACC, Sensitivity) (Specificity, Precision)	IF-RVFL [32] (ACC, Sensitivity) (Specificity, Precision)	GB-RVFLwoDL* (ACC, Sensitivity) (Specificity, Precision)	GB-RVFL* (ACC, Sensitivity) (Specificity, Precision)	GE-GB-RVFL* (ACC, Sensitivity) (Specificity, Precision)
adenosis_vs_ductal_carcinoma	(85.26, 76.67)	(77.89, 63.33)	(88.42, 80)	(85.26, 86.67)	(91.58, 92.59)	(81.05, 60)
	(89.23, 76.67)	(84.62, 65.52)	(92.31, 82.76)	(84.62, 72.22)	(91.18, 80.65)	(90.77, 75)
adenosis_vs_lobular_carcinoma	(54.79, 42.86)	(46.58, 42.86)	(52.05, 0)	(50.68, 42.86)	(53.42, 17.14)	(58.9, 45.71)
	(65.79, 53.57)	(50, 44.12)	(100, 0)	(57.89, 48.39)	(86.84, 54.55)	(71.05, 59.26)
adenosis_vs_mucinous_carcinoma	(66.27, 18.18)	(60.24, 0)	(61.45, 15.15)	(67.47, 45.45)	(57.83, 54.55)	(61.45, 43.75)
	(98, 85.71)	(100, 0)	(92, 55.56)	(82, 62.5)	(60, 47.37)	(72.55, 50)
adenosis_vs_papillary_carcinoma	(55.41, 42.42)	(52.7, 36.36)	(55.41, 0)	(54.05, 60.61)	(51.35, 37.93)	(55.41, 40)
	(65.85, 50)	(65.85, 46.15)	(100, 0)	(48.78, 48.78)	(60, 37.93)	(69.23, 53.85)
CN_vs_AD	(88, 85.96)	(84, 80.7)	(77.6, 73.68)	(79.2, 64.91)	(84, 73.68)	(85.6, 80.7)
	(89.71, 87.5)	(86.76, 83.64)	(80.88, 76.36)	(91.18, 86.05)	(92.65, 89.36)	(89.71, 86.79)
CN_vs_MCI	(72.87, 86.78)	(73.4, 86.78)	(73.94, 86.78)	(73.4, 86.78)	(70.21, 87.6)	(75, 87.6)
	(47.76, 75)	(49.25, 75.54)	(50.75, 76.09)	(49.25, 75.54)	(38.81, 72.11)	(52.24, 76.81)
fibroadenoma_vs_ductal_carcinoma	(79.85, 77.92)	(79.1, 77.92)	(79.85, 75.32)	(70.9, 66.23)	(79.85, 79.22)	(78.36, 76.62)
	(82.46, 85.71)	(80.7, 84.51)	(85.96, 87.88)	(77.19, 79.69)	(80.7, 84.72)	(80.7, 84.29)
fibroadenoma_vs_lobular_carcinoma	(65.49, 100)	(64.6, 98.65)	(65.49, 100)	(49.56, 64.86)	(66.37, 83.78)	(60.18, 68.92)
	(0, 65.49)	(0, 65.18)	(0, 65.49)	(20.51, 60.76)	(33.33, 70.45)	(43.59, 69.86)
fibroadenoma_vs_mucinous_carcinoma	(50, 52.44)	(51.64, 58.54)	(56.56, 69.51)	(59.84, 58.54)	(55.74, 57.89)	(61.48, 64.63)
	(45, 66.15)	(37.5, 65.75)	(30, 67.06)	(62.5, 76.19)	(52.17, 66.67)	(55, 74.65)
fibroadenoma_vs_papillary_carcinoma	(61.95, 77.63)	(55.75, 76.32)	(59.29, 72.37)	(53.98, 73.68)	(64.6, 93.42)	(66.37, 97.37)
	(29.73, 69.41)	(13.51, 64.44)	(32.43, 68.75)	(13.51, 63.64)	(5.41, 66.98)	(2.7, 67.27)
MCI_vs_AD	(68.18, 29.51)	(69.89, 32.79)	(67.05, 24.59)	(67.05, 19.67)	(69.32, 49.18)	(72.16, 39.34)
	(88.7, 58.06)	(89.57, 62.5)	(89.57, 55.56)	(92.17, 57.14)	(80, 56.6)	(89.57, 66.67)
phyllodes_tumour_vs_ductal_carcinoma	(84.54, 85)	(81.44, 92.5)	(84.54, 82.5)	(60.82, 20)	(83.51, 88.57)	(90.72, 94.12)
	(84.21, 79.07)	(73.68, 71.15)	(85.96, 80.49)	(89.47, 57.14)	(80.65, 72.09)	(88.89, 82.05)
phyllodes_tumour_vs_lobular_carcinoma	(56.58, 57.14)	(48.68, 48.57)	(57.89, 34.29)	(50, 37.14)	(59.21, 42.86)	(59.21, 71.43)
	(56.1, 52.63)	(48.78, 44.74)	(78.05, 57.14)	(60.98, 44.83)	(73.17, 57.69)	(48.78, 54.35)
phyllodes_tumour_vs_mucinous_carcinoma	(47.67, 55.56)	(51.16, 66.67)	(50, 44.44)	(48.84, 62.96)	(53.49, 24.32)	(36.05, 48.15)
	(44.07, 31.25)	(44.07, 35.29)	(52.54, 30)	(42.37, 33.33)	(75.51, 42.86)	(30.51, 24.07)
phyllodes_tumour_vs_papillary_carcinoma	(44.74, 34.29)	(60.53, 45.71)	(44.74, 11.43)	(60.53, 22.86)	(56.58, 57.14)	(53.95, 41.67)
	(53.66, 38.71)	(73.17, 59.26)	(73.17, 26.67)	(92.68, 72.73)	(56.1, 52.63)	(65, 51.72)
tubular_adenoma_vs_ductal_carcinoma	(68.63, 53.19)	(67.65, 51.06)	(67.65, 46.81)	(66.67, 40.43)	(66.67, 42.55)	(67.65, 40.43)
	(81.82, 71.43)	(81.82, 70.59)	(85.45, 73.33)	(89.09, 76)	(87.27, 74.07)	(90.91, 79.17)
tubular_adenoma_vs_lobular_carcinoma	(45.68, 33.33)	(56.79, 47.62)	(51.85, 45.24)	(45.68, 57.14)	(56.79, 40.54)	(50.62, 23.08)
	(58.97, 46.67)	(66.67, 60.61)	(58.97, 54.29)	(33.33, 48)	(70.45, 53.57)	(76.19, 47.37)
tubular_adenoma_vs_mucinous_carcinoma	(50, 23.26)	(52.22, 23.26)	(53.33, 18.6)	(57.78, 23.26)	(63.33, 17.95)	(61.11, 43.59)
	(74.47, 45.45)	(78.72, 50)	(85.11, 53.33)	(89.36, 66.67)	(98.04, 87.5)	(74.51, 56.67)
tubular_adenoma_vs_papillary_carcinoma	(55.56, 41.46)	(49.38, 70.73)	(54.32, 21.95)	(40.74, 39.02)	(60.49, 46.34)	(46.91, 89.47)
	(70, 58.62)	(27.5, 50)	(87.5, 64.29)	(42.5, 41.03)	(75, 65.52)	(9.3, 46.58)
Average ACC	63.23	62.3	63.23	60.13	**65.49**	64.32
Average Rank	3.37	4	3.45	4.55	2.95	**2.68**

The best-performing model in terms of average accuracy and average ranks are indicated by boldface in the second last and last rows, respectively. ACC is an acronym for accuracy.
* denotes the proposed models.

169. Similarly, the benign class had subcategories like tubular adenoma, phyllodes tumour, adenosis, and fibroadenoma with corresponding scan counts of 130, 115, 106, and 237. Our approach for feature extraction from these histopathological scans mirrored the methodology outlined in the references [188]. We employed a total of 16 combinations involving both benign and malignant classes to differentiate between the various subclasses of benign and malignant cancers, as summarized in Table 4.6.

- ***ADNI dataset***: Alzheimer's disease (AD) stands out as a prevalent neurodegenerative condition [189], accounting for approximately 70% of dementia cases, according to Khojaste-Sarakhsi et al. [190]. Despite ongoing research efforts, the exact cause of AD remains elusive, and an effective cure is yet to be discovered [14]. This underscores the critical need for early detection methods. In our study, we utilized scans

from the ADNI dataset, accessible via *https://adni.loni.usc.edu/*. The ADNI project was initiated with the aim to investigate neuroimaging modalities like positron emission tomography (PET), magnetic resonance imaging (MRI), and other diagnostic assessments for AD, particularly focusing on mild cognitive impairment (MCI) stages. Our approach for feature extraction aligns with the methodology described in reference [39]. The dataset we worked with encompasses three primary comparisons: MCI versus AD (MCI_vs_AD), control normal (CN) versus MCI (CN_vs_MCI), and CN versus AD (CN_vs_AD).

Results and Discussion

Table 4.6 presents the accuracy, sensitivity, specificity, and precision values for both the proposed models and the baseline models. Our analysis demonstrates that among all the models evaluated, the proposed GB-RVFL achieves the highest accuracy at 65.49%, followed by the GE-GB-RVFL with an accuracy of 64.32%, ranking as the second-highest. From the last row of the table, we observe that the average rank of the proposed GE-GB-RVFL is minimal, followed by GB-RVFL. This also indicates the superior performance of our proposed models. For a more nuanced understanding of performance, we examined sensitivity, specificity, and precision through graphs plotted with respect to the ADNI datasets, as depicted in Fig. 4.3. As noted in the literature [191], distinguishing the MCI_vs_AD case poses a significant challenge. However, our findings indicate that the proposed GB-RVFL and GE-GB-RVFL models showcase the highest and second-highest sensitivity levels, a crucial metric in biomedical data analysis. This trend is similarly observed in the CN_vs_MCI case. In summary, our proposed models demonstrate robust performance across various metrics, showcasing their efficacy in real-world datasets, i.e., breast cancer and AD classification.

4.2.5 Interpretability of the Proposed Models Through Graph Embedding and Granular Computing

A central claim in the development of the proposed GE-GB-RVFL model is to maintain the geometric structure of the original data, an aspect that randomised models often struggle to preserve. This ability to leverage the inherent geometry of the data is particularly valuable, as it can lead to more informed and accurate decision-making. Thus, ensuring that the model makes decisions based on the actual features of the dataset—rather than functioning as a black-box model—becomes even more critical. In this section, we provide both a theoretical framework and visual evidence to demonstrate that the interpretability of the proposed GE-GB-RVFL and GB-RVFL models is superior to that of traditional baseline models. By doing so, we aim to show not only how the model performs but also why it arrives at its decisions. We begin by exploring which features each model relies on when making its final predictions, offering insights into their decision-making processes.

For GE-GB-RVFL, from Eq. (4.24), we observe that the weight matrix is defined as: $\Omega = \left(D^tD + \frac{1}{\mathcal{C}}I + \frac{\alpha}{\mathcal{C}}U_{uni}\right)^{-1} D^t\mathcal{W}$. This implies that the features used to make the final decision are influenced by the inverse of the matrix $\left(D^tD + \frac{1}{\mathcal{C}}I + \frac{\alpha}{\mathcal{C}}U_{uni}\right)^{-1} D^t$, i.e., $(D^t)^\dagger \left(D^tD + \frac{1}{\mathcal{C}}I + \frac{\alpha}{\mathcal{C}}U_{uni}\right) =: \mathcal{E}_1 \in \mathbb{R}^{(L+g)\times k}$, where $\dagger$ is the Moore-Penrose inverse. Furthermore, for the GB-RVFL model (from (4.18)), the features used to predict the final output can be expressed as $(D^t)^\dagger \left(D^tD + \frac{1}{\mathcal{C}}I\right) =: \mathcal{E}_2 \in \mathbb{R}^{(L+g)\times k}$. Similarly, for the features related to the original GB matrix $\mathcal{O}$, the corresponding feature matrix can be defined as $(\mathcal{O}^t)^\dagger \left(\mathcal{O}^t\mathcal{O} + \frac{1}{\mathcal{C}}I\right) =: \mathcal{E}_3 \in \mathbb{R}^{L\times M}$.

For the RVFL, the output weights $\mathcal{Q}$ (for hidden layer matrix $\mathcal{G}$) are defined as: $\mathcal{Q} = [U \oplus \mathcal{G}]^\dagger T = \left([U \oplus \mathcal{G}]^t[U \oplus \mathcal{G}] + \frac{1}{\mathcal{C}}I\right)^{-1} [U \oplus \mathcal{G}]^t T$. Thus, the features contributing to the final decision of the RVFL can be represented as: $([U \oplus \mathcal{G}]^t)^\dagger \left([U \oplus \mathcal{G}]^t[U \oplus \mathcal{G}] + \frac{1}{\mathcal{C}}I\right) =: \mathcal{E}_4 \in \mathbb{R}^{(L+g)\times M}$. Similarly, the features associated with the RVFLwoDL and for the original

Table 4.7: Feature interpretability results for the datasets conn_bench_sonar_mines_rocks and acute_inflammation, based on the Frobenius norm distance of the feature distance matrices.

Dataset ↓ / Distance →	$\mathcal{DDE}_1$	$\mathcal{DDE}_2$	$\mathcal{DDE}_4$	$\mathcal{DDE}_5$
acute_inflammation	$3.49E-05$	$3.60E-05$	0.2559	92.3832
conn_bench_sonar_mines_rocks	$6.03E-05$	$6.13E-05$	$7.79E-05$	2186.1229

input features are given by: $(\mathcal{G}^t)^\dagger \left(\mathcal{G}^t\mathcal{G} + \frac{1}{\mathcal{C}}I\right) =: \mathcal{E}_5 \in \mathbb{R}^{g\times M}$ and $(U^t)^\dagger \left(U^tU + \frac{1}{\mathcal{C}}I\right) =: \mathcal{E}_6 \in \mathbb{R}^{L\times M}$, respectively.

We adjust the dimensions of each $\mathcal{E}_i$ (for $i = 1, 2$) to match that of $\mathcal{E}_3$ by cropping the extra rows and/or columns. Similarly, we modified the dimensions of each $\mathcal{E}_i$ (for $i = 4, 5$) to align with $\mathcal{E}_6$ through the same cropping process. These modified matrices are denoted as $\mathcal{E}_i'$ (for $i = 1, 2, 4, 5$). For consistency, we also renamed $\mathcal{E}_3$ and $\mathcal{E}_6$ as $\mathcal{E}_3'$ and $\mathcal{E}_6'$, respectively.

The next step involves calculating the pairwise distance between each row of the feature matrices $\mathcal{E}_i'$ (for $i = 1, 2, \ldots, 6$), denoted as $\mathcal{DE}_i$, where each entry represents the distance between features within the matrix. Note that $\mathcal{DE}_i \in \mathbb{R}^{k\times k}$ for $i = 1, 2, 3$ and $\mathcal{DE}_i \in \mathbb{R}^{M\times M}$ for $i = 4, 5, 6$. Once we obtain feature distance matrices $\mathcal{DE}_i$, we can compute the distance between each feature distance matrix $\mathcal{DE}_i$ (for $i = 1, 2$) and the feature distance matrix $\mathcal{DE}_3$ using the Frobenius norm. This is defined as:

$$\mathcal{DDE}_i = \|\mathcal{DE}_i - \mathcal{DE}_3\|_F, \quad \text{for} \quad i = 1, 2.$$

Similarly, we define $\mathcal{DDE}_4$ and $\mathcal{DDE}_5$ as:

$$\mathcal{DDE}_i = \|\mathcal{DE}_i - \mathcal{DE}_6\|_F, \quad \text{for} \quad i = 4, 5,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, which provides a measure of the difference between the distance matrices.

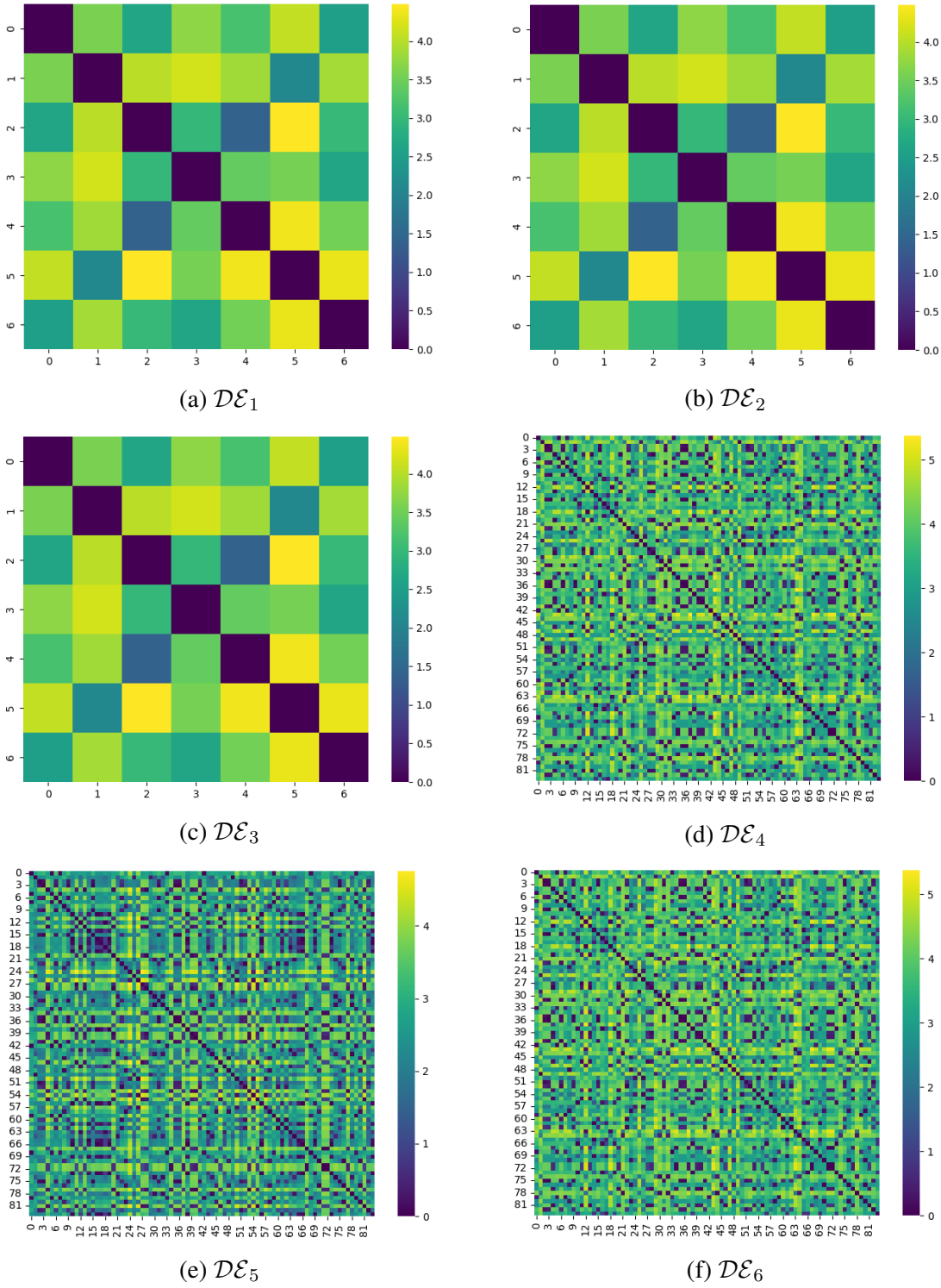
The core concept is that smaller values of $\mathcal{DDE}_i$ signify a closer alignment between the features in $\mathcal{E}_i$ and the original features (V) or the original GB features ($\mathcal{O}$). Therefore, lower $\mathcal{DDE}_i$ values reflect superior preservation of the original feature structure within the learned representations of the model. This suggests that the model effectively captures and retains the essential characteristics of the original data, enhancing interpretability and reliability in decision-making processes.

To assess the feature interpretability of our proposed models, we conducted experiments on two datasets: "conn_bench_sonar_mines_rocks" and "acute_inflammation". The results are summarized in Table 4.7. Notably, the distance $\mathcal{DDE}_1$ is the smallest among the measured values, indicating that the features associated with the GE-GB-RVFL model are more closely aligned with the original GB features. This alignment is attributed to the use of graph embedding, which effectively preserves the geometric relationships inherent in the data, along with a slight contribution from granular computing techniques.

Following this, the GB-RVFL model shows a slightly larger distance, reinforcing the enhanced feature interpretability of our proposed model. Furthermore, the distance associated with RVFL features is smaller than that of the RVFLwoDL features, highlighting the significance of direct links in feature representation.

These findings demonstrate that the feature alignment of the proposed models is significantly improved—by factors ranging from 10^4 to 10^9—compared to the baseline RVFL and RVFLwoDL models. This enhancement implies a greater degree of feature interpretability for both proposed models.

Additionally, we visualize the distances among the features of different matrices $\mathcal{E}_i$ by generating heat maps for each $\mathcal{DE}_i$, as depicted in Figs. 4.4 and 4.5. Since the dimension of $\mathcal{DE}_i$ is influenced by the sample space's dimension, i.e., M (the number of original samples) and k (the number of GBs), with $k << M$. Consequently, the dimensions of the

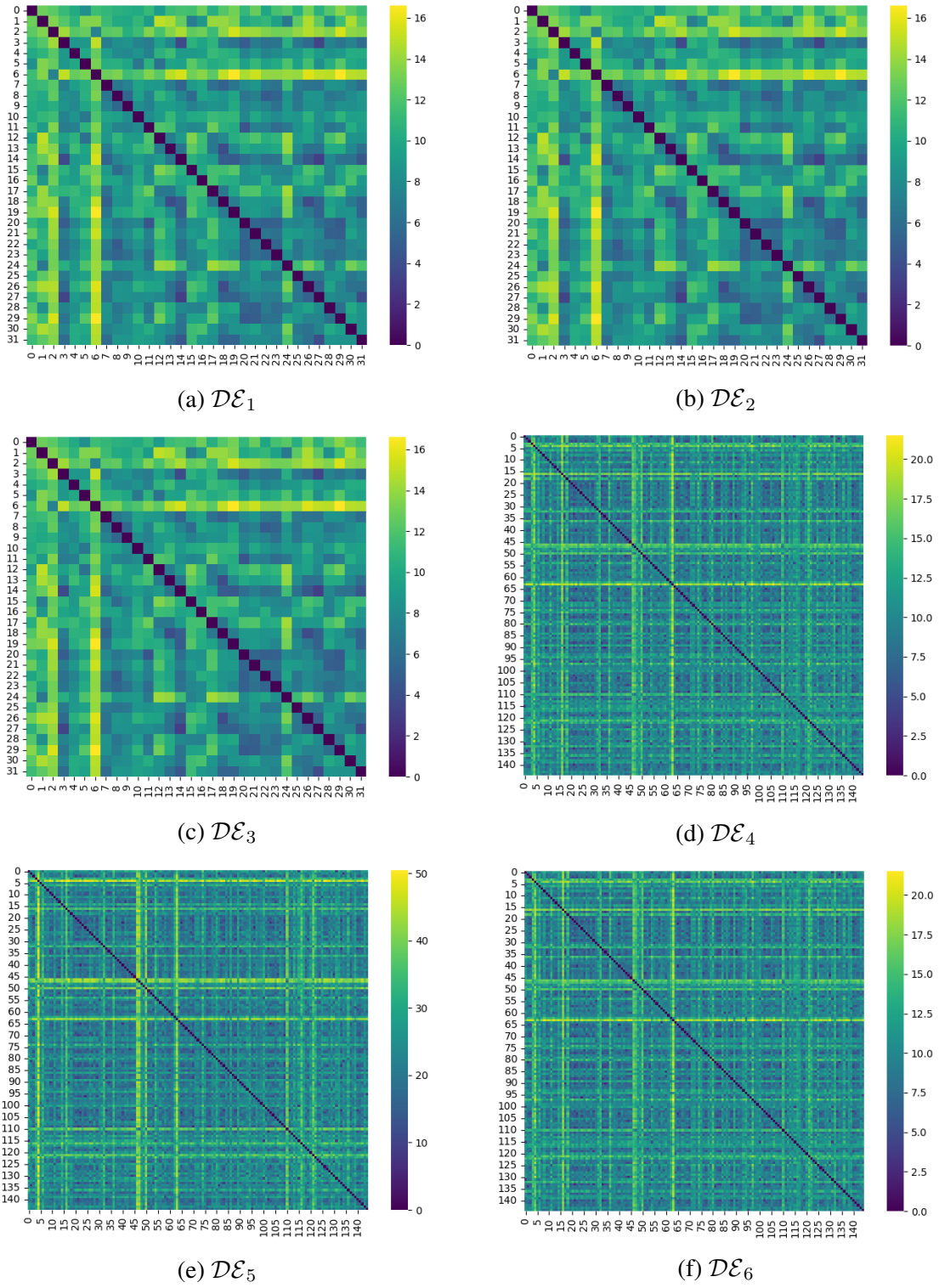


(a) $\mathcal{DE}_1$ (b) $\mathcal{DE}_2$

(c) $\mathcal{DE}_3$ (d) $\mathcal{DE}_4$

(e) $\mathcal{DE}_5$ (f) $\mathcal{DE}_6$

Figure 4.4: Visualization of the feature distance matrices $\mathcal{DE}_i$ through Heat map on "acute_inflammation" dataset.



(a) $\mathcal{DE}_1$ (b) $\mathcal{DE}_2$

(c) $\mathcal{DE}_3$ (d) $\mathcal{DE}_4$

(e) $\mathcal{DE}_5$ (f) $\mathcal{DE}_6$

Figure 4.5: Visualization of the feature distance matrices $\mathcal{DE}_i$ through Heat map on "conn_bench_sonar_mines_rocks" dataset.

heat maps vary accordingly in the figures. These visual representations further illustrate the relationships between features, providing a clearer understanding of how effectively our models capture the original feature structure. By combining theoretical insights with visual evidence, we demonstrate that our proposed GE-GB-RVFL and GB-RVFL models not only outperform traditional baselines but also provide clarity on their decision-making processes, enhancing their interpretability and reliability.

4.2.6 Discussion

This section delves into a fundamental inquiry: do the proposed models achieve a harmonious balance between scalability and accuracy within the realm of granular computing? Subsequently, we conduct various sensitivity analyses of the hyperparameters of the proposed models to assess the effect of various hyperparameters on the performance of the proposed GB-RVFL and GE-GB-RVFL models. These hyperparameters include pur (purity threshold), num (# minimum GBs), $Actfun$ (activation function), $\mathcal{C}$ (regularization parameter) and graph regularization parameter (α).

4.2.6.1 Balance Between Number of GBs Generated and the Resulting Accuracy of the Proposed GB-RVFL and GE-GB-RVFL Models with Different Purities

Purity significantly impacts the formation of GBs, thus influencing the overall performance of GB-RVFL and GE-GB-RVFL models. Our objective is to explore the correlation between the number of GBs generated ($number(GB)$) and the threshold purity (pur), and how this correlation affects our models' performance. We've chosen five diverse UCI and KEEL datasets to showcase the adaptability and effectiveness of GB-RVFL and GE-GB-RVFL.

Tables 4.8 and 4.9 present the number of GBs and the corresponding accuracy achieved

Table 4.8: The number of GBs and the corresponding Accuracies by the proposed GB-RVFL model under different purities.

pur	1	0.97	0.94	0.91	0.88	0.85	0.82	0.79
Dataset	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$
blood	78.22	76	77.33	76.44	75.11	76.89	76.89	56.89
	349	354	355	340	314	245	211	172
breast_cancer	74.42	74.42	77.91	69.77	74.42	67.44	80.23	66.28
	139	130	142	132	137	119	119	90
breast_cancer_wisc_prog	63.33	68.33	60	70	73.33	68.33	55	66.67
	94	88	84	88	85	73	71	52
chess_krvkp	94.99	93.95	93.85	93.64	94.16	90.51	91.14	89.68
	1168	1195	1131	1062	969	858	757	619
cleve	83.33	72.22	84.44	82.22	71.11	68.89	65.56	73.33
	108	113	114	100	87	65	45	34

Table 4.9: The number of granular balls and the corresponding Accuracies by the proposed GE-GB-RVFL model under different purities.

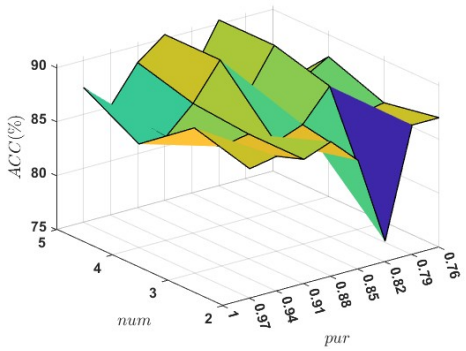
pur	1	0.97	0.94	0.91	0.88	0.85	0.82	0.79
Dataset	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$	$number(GB)$
blood	76	77.33	77.33	75.11	74.67	76.89	51.11	76.44
	349	354	355	340	314	249	219	180
breast_cancer	74.42	65.12	74.42	74.42	70.93	72.09	66.28	52.33
	139	130	142	132	137	119	119	90
breast_cancer_wisc_prog	66.67	58.33	65	56.67	63.33	50	58.33	61.67
	94	96	93	92	87	75	71	52
chess_krvkp	92.18	90.82	90.93	89.47	93.01	90.09	90.93	89.57
	1168	1195	1131	1062	969	858	757	619
cleve	85.56	78.89	77.78	67.78	75.56	74.44	74.44	70
	108	113	114	111	107	68	43	37

by GB-RVFL and GE-GB-RVFL models across different purity levels from 0.79 to 1.0. Notably, a clear pattern emerges: as purity decreases, the number of GBs also decreases, indicating a refinement in granularity with higher purity levels. Additionally, there's a consistent trend of increased accuracy with higher purity levels across both models.

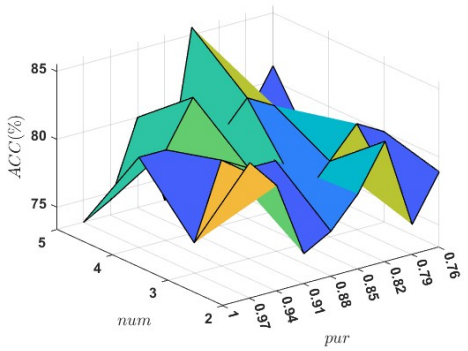
This leads us to answer the very fundamental question in GB research: can we strike a balance between scalability and accuracy in granular computing models? Our models' scalability has already been demonstrated through experiments on NDC datasets. Furthermore, our above analyses show that our models effectively handle varying purity levels, highlighting their adaptability and effectiveness in granular computing applications.

4.2.6.2 Effect of Granular Ball Parameters "pur" and "num" on the Performance of the Proposed GB-RVFL and GE-GB-RVFL Models

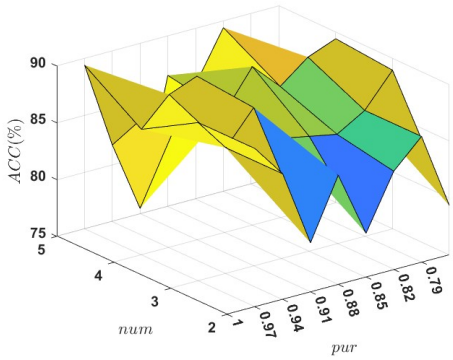
We evaluate the impact of GB parameters pur and num on the performance of GB-RVFL and GE-GB-RVFL models using datasets aus and clever, as shown in Fig. 4.6.



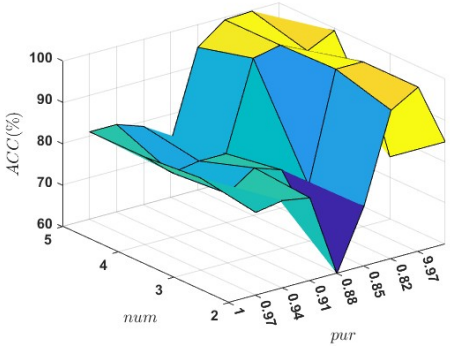
(a) aus (GB-RVFL)



(b) cleve (GB-RVFL)



(c) aus (GE-GB-RVFL)



(d) cleve (GE-GB-RVFL)

Figure 4.6: Effect of granular ball parameters pur and num on the performance of the proposed GB-RVFL and GE-GB-RVFL models.

- For GB-RVFL: Examining Figs. 4.6(a) and 4.6(b), we note that higher values of num, coupled with pur in the vicinity of 0.85, yield optimal performance for the model.

- For GE-GB-RVFL: Analyzing Figs. 4.6(c) and 4.6(d), we observe that increasing num values, particularly with lower pur, result in superior performance for the model.

These observations unveil a discernible pattern regarding the impact of pur and num on the proposed models. This pattern identifies a subset of parameters that, when utilized in training our model, significantly enhances its efficiency.

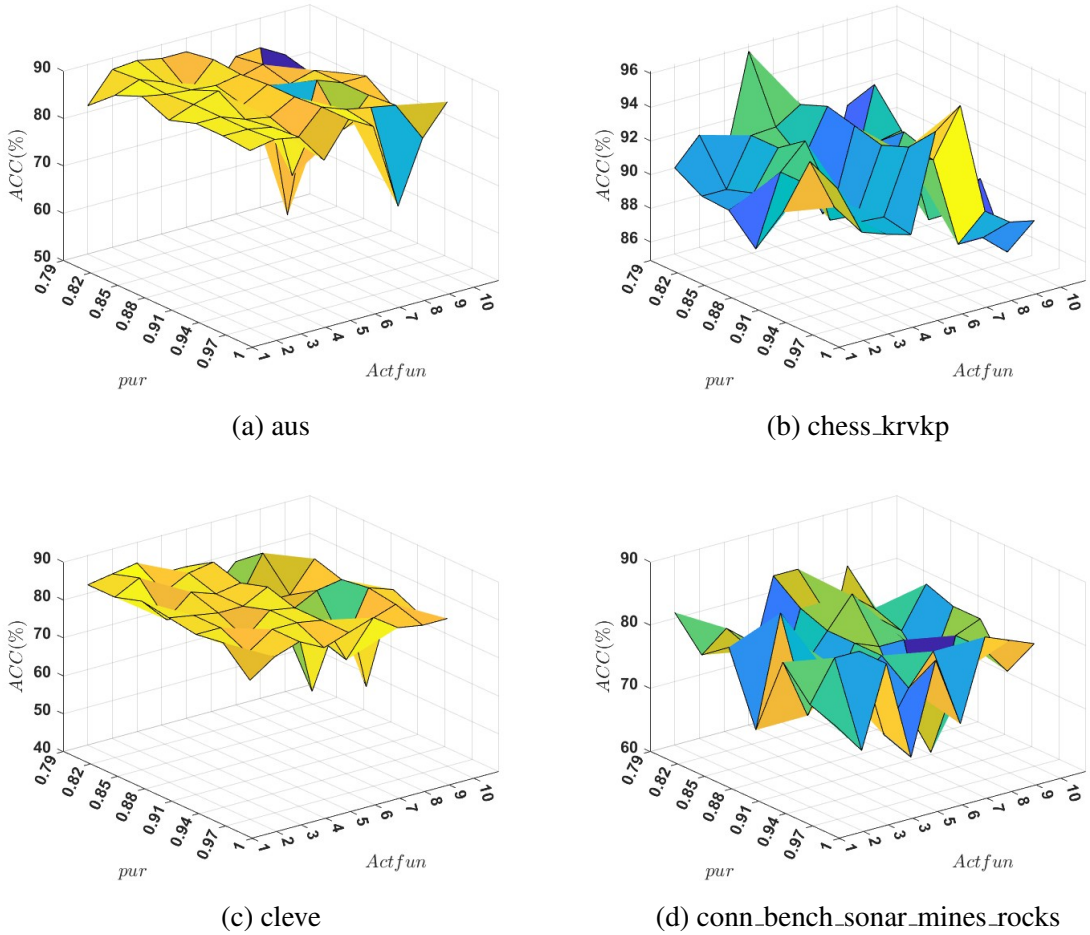
4.2.6.3 Effect of Parameter "pur" and "$Actfun$" on the Performance of the Proposed GB-RVFL Model

The activation function plays a crucial role in the GB-RVFL model's performance, particularly in its interaction with granular ball generation and its overall impact on model performance. We analyze this relationship using Fig. 4.7 across datasets aus, chess_krvkp, clever, and conn_bench_sonar_mines_rocks. In these 3-D graphs, pur and $Actfun$ serve as variables, with ACC representing the final output.

Our observations reveal a nuanced sensitivity to hyperparameters. For example, in the aus dataset, activation function 9 exhibits the lowest performance within the purity range of 0.88 to 0.91. Conversely, in the chess_krvkp dataset, activation function 9 demonstrates superior performance within the purity range of 0.85 to 0.88. Thus, we observe the mix performance with respect to $Actfun$, and no particular patterns have been found. These findings highlight the need for fine-tuning the activation function to achieve better results.

4.2.6.4 Effect of Parameters "$\mathcal{C}$" and "$Actfun$" on the performance of the proposed GB-RVFL and GE-GB-RVFL models

To evaluate the impact of the regularization parameter $\mathcal{C}$ on the performance of GB-RVFL and GE-GB-RVFL models, we present graphs in Fig. 4.8 with $\mathcal{C}$ and $Actfun$ as independent variables and ACC as the dependent variable across the aus and bank datasets. Our analysis reveals interesting insights. In the bank datasets, both GB-RVFL and GE-GB-RVFL models demonstrate consistent performance, except for instances where $\mathcal{C}$ equals 10^{-3} and/or 10^{-1}, where some variability is observed. However, on the aus dataset, the performance of the proposed models shows sensitivity to hyperparameters, indicating a dataset-dependent behavior. This underscores the importance of tuning the parameter $\mathcal{C}$ to achieve optimal results based on the specific dataset characteristics. In conclusion, our



(a) aus (b) chess_krvkp

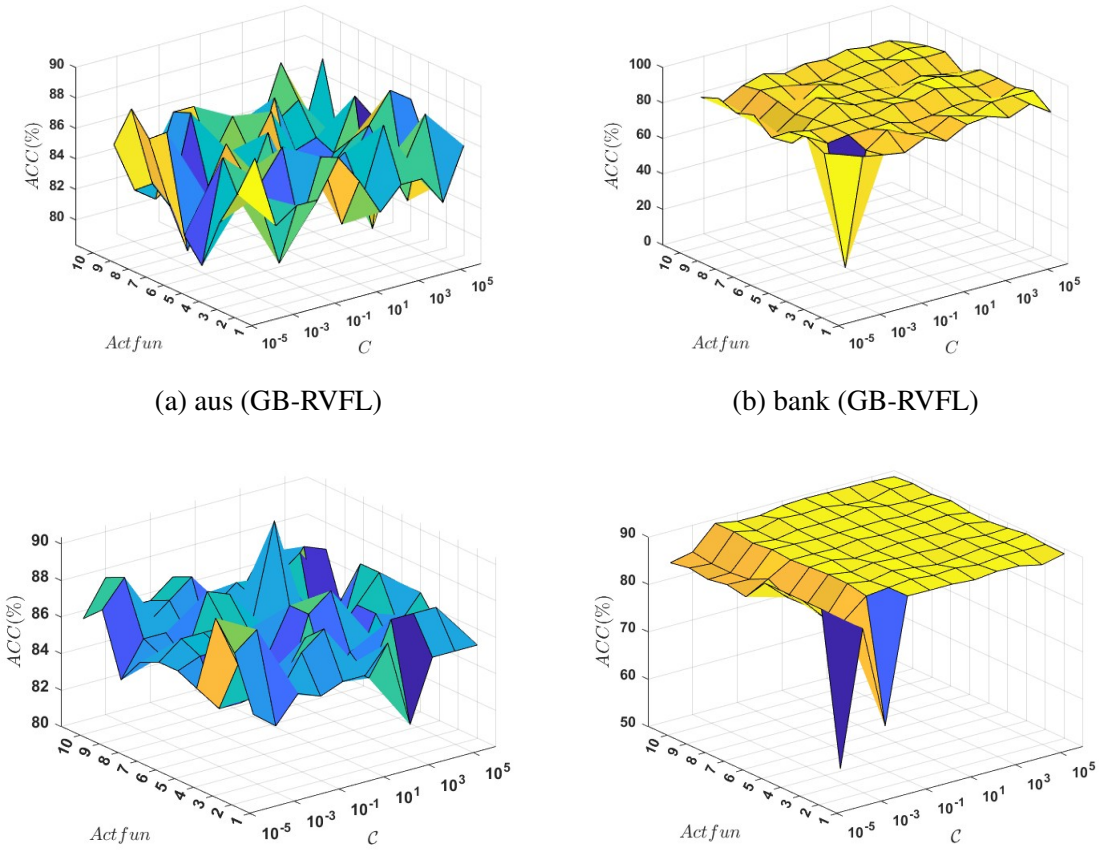
(c) cleve (d) conn_bench_sonar_mines_rocks

Figure 4.7: Effect of parameters purity (pur) and activation function $(Act\ fun)$ on the performance of the proposed GB-RVFL model.

results underscore the impact of dataset characteristics on the performance of the proposed models, stressing the importance of fine-tuning the regularization parameter $\mathcal{C}$ for optimal model performance.

4.2.6.5 Effect of Parameters "α" on the performance of the proposed GE-GB-RVFL model

The effect of the hyperparameter α on the GE-GB-RVFL model's performance is depicted in Fig. 4.9. Our analysis demonstrates a clear pattern: as the lowest value of α, there is a significant improvement in the model's performance, particularly when $\alpha = 10^{-5}$.



(a) aus (GB-RVFL)

(b) bank (GB-RVFL)

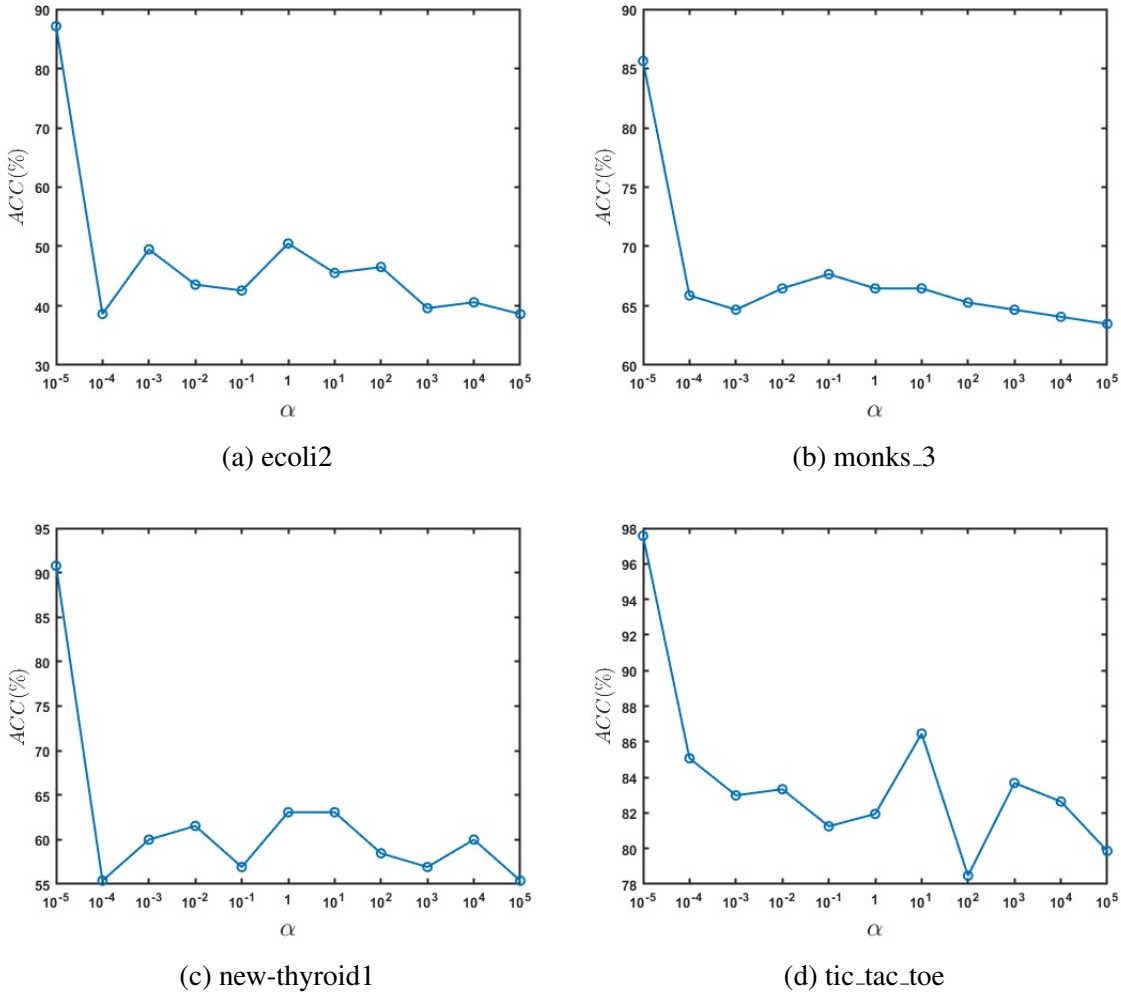
(c) aus (GE-GB-RVFL)

(d) bank (GE-GB-RVFL)

Figure 4.8: Effect of parameters $\mathcal{C}$ and activation function $(Act\ fun)$ on the performance of the proposed GB-RVFL and GE-GB-RVFL models.

Beyond this point, however, the performance tends to level off, as indicated by the accuracy (ACC) metrics. This plateau suggests diminishing returns in performance gains for larger values of α.

The results emphasize that the optimal performance of the GE-GB-RVFL model is highly dependent on the careful selection and fine-tuning of α. Small values of α contribute positively to performance, but further increases beyond a certain threshold offer minimal additional benefit. This sensitivity of the model to α highlights the importance of considering the characteristics of the dataset when tuning the model. Different datasets may respond differently to variations in α, meaning that optimal performance may vary depending on the



(a) ecoli2 (b) monks_3

(c) new-thyroid1 (d) tic_tac_toe

Figure 4.9: Effect of parameters α on the performance of the proposed GE-GB-RVFL model.

specific features and patterns present in the data. Thus, fine-tuning α is crucial for ensuring that the GE-GB-RVFL model reaches its full potential in terms of accuracy and robustness across diverse datasets.

4.3 Proposed Work: Wave-RVFL

Recently, the Wave loss function, introduced in [71], has been employed in support vector machines (SVM) with some desired claims. It is said to have a trend towards smoothness, insensitivity to noise, and resilience against outliers. This raises an intriguing ques-

tion: can the Wave loss function be effectively utilized in the optimization function of the RVFL network? If so, can we formulate a solution to this optimization problem, and will the integrated model (RVFL with Wave loss function) exhibit the same robustness against noise and outliers? Inspired by the promising properties of the Wave loss function (refer Section 4.3.1.1) and motivated by the potential answers to these questions, we propose the Wave-RVFL model, an RVFL model based on the Wave loss function. In developing Wave-RVFL, we aimed to preserve the inherent optimization formulation of the RVFL as much as possible, simply replacing the square loss function with the Wave loss function. The proposed optimization problem of Wave-RVFL is solved using the adaptive moment estimation (Adam) algorithm [73].

4.3.1 Proposed Method

In this Section, we propose a novel Wave-RVFL model and provide the mathematical formulation, its solution using Adam algorithm. Wave-RVFL capitalizes on the asymmetry inherent in the Wave loss function to dynamically apply penalties at the instance level for misclassified samples. The proposed Wave-RVFL demonstrates robustness against noise and outliers by preventing excessive penalization of deviations, thereby ensuring a balanced treatment of noise and outliers. We start by reviewing some of the properties of the Wave loss function and then dive into the formulation of Wave-RVFL.

4.3.1.1 Wave loss Function

We present the formulation of the Wave loss function, engineered to be resilient to outliers, robust against noise, and smooth in its properties, which can be formulated as:

$$\mathcal{L}_{wave}(v) = \frac{1}{\eta}\left(1 - \frac{1}{1+\eta v^2 \exp\{(\gamma v)\}}\right), \ \ \forall\, v \in \mathbb{R}, \tag{4.25}$$

where $\eta \in \mathbb{R}^+$ represents the bounding parameter and $\gamma \in \mathbb{R}$ denotes the shape parameter. Figure 4.10 visually illustrates the Wave loss function. The Wave loss function exhibits the following properties [71]:

[1] It is bounded, smooth, and non-convex function.

[2] The Wave loss function introduces two essential parameters: the shape parameter γ, dictating the shape of the loss function, and the bounding parameter η, determining the loss function threshold values.

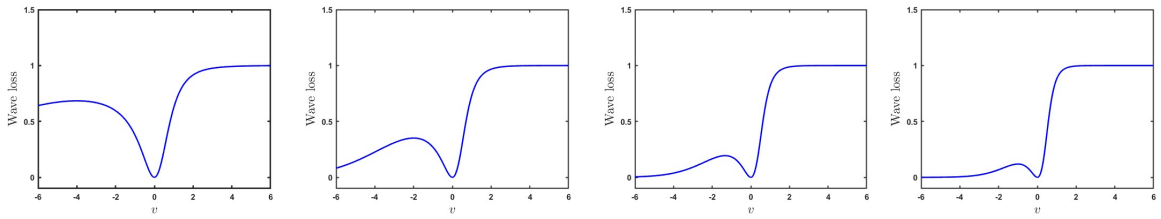
[3] The Wave loss function is infinitely differentiable and hence continuous.

[4] The Wave loss function showcases resilience against outliers and noise insensitivity. With loss bounded to $\frac{1}{\gamma}$, it handles outliers robustly while assigning loss to samples with $v \leq 0$, displaying resilience to noise.

[5] As γ tends to infinity, for a fixed η, the Wave loss function converges point-wise to the $0 - \frac{1}{\eta}$ loss, expressed as:

$$\mathcal{L}_{0-1/\eta}(v) = \begin{cases} 0, & \text{if } v \leq 0, \\ \frac{1}{\eta}, & \text{if } v > 0. \end{cases} \tag{4.26}$$

Furthermore, the Wave loss converges to the "$0-1$" loss when $\eta = 1$.



(a) $\gamma = 0.5$ (b) $\gamma = 1$ (c) $\gamma = 1.5$ (d) $\gamma = 2$

Figure 4.10: A graphical representation of the Wave loss function is provided, with η set to 1, and varying values of γ.

4.3.1.2 Optimization Problem: Wave-RVFL

In this subsection, we give the mathematical formulation of the proposed Wave-RVFL as follows.

$$\begin{aligned} &\min f(\beta) = \min \frac{1}{2}\|\beta\|^2 + \frac{\mathcal{C}}{2}\sum_{i=1}^{M}\mathcal{L}_{wave}(\xi_i) \\ &s.t.\ \ z_i\beta = [u_i, h(u_i)]\beta - T_i = \xi_i,\ \forall i \in \{1, 2, \ldots, M\}, \end{aligned} \tag{4.27}$$

where $z_i = [u_i, h(u_i)]$, $h(u_i)$ is the hidden layer corresponding to the sample u_i, $\mathcal{C}$ represents the tunable parameter, β is the output layer matrix and ξ_i represents the error variable, allowing tolerance for misclassifications in situations of overlapping distributions. Putting the values of ξ_i in the optimization function of (4.27), we obtain

$$\min f(\beta) = \min \frac{1}{2}\|\beta\|^2 + \frac{\mathcal{C}}{2}\sum_{i=1}^{M}\mathcal{L}_{wave}(z_i\beta - T_i), \tag{4.28}$$

where $\mathcal{L}_{wave}$ is the Wave loss function defined in (4.25).

Then, the optimization is reduced to the following form:

$$\min f(\beta) = \min \frac{1}{2}\|\beta\|^2 + \frac{\mathcal{C}}{2\eta}\sum_{i=1}^{M}\left(1 - \frac{1}{1 + \eta(z_i\beta - T_i)^2 e^{\gamma(z_i\beta - T_i)}}\right). \tag{4.29}$$

Optimizing the Wave-RVFL model presents challenges due to the non-convexity of the loss function. Nevertheless, by leveraging the inherent smoothness of Wave-RVFL, gradient-based algorithms can be employed for model optimization.

4.3.1.3 Soultion: Wave-RVFL

Now, we employ the adaptive moment estimation (Adam) algorithm for solving the optimizing problem (4.29). The (4.29) can be written as:

$$\min f(\beta) = \min \frac{1}{2}\|\beta\|^2 + \frac{\mathcal{C}}{2\eta}\sum_{i=1}^{M}\left(1 - \frac{1}{1+\eta\xi_i^2 e^{\gamma\xi_i}}\right), \tag{4.30}$$

where $\xi_i = z_i\beta - T_i$.

At t^{th} iteration (maximum number of iterations is denoted as "Itr"), s samples are selected randomly and used to find the gradient and the following steps. Taking the gradient with respect to β, we obtain:

$$\frac{\partial f(\beta)}{\partial \beta} = \beta + \frac{\mathcal{C}}{2}\sum_{i=1}^{s}\left(\frac{\xi_i z_i^T e^{\gamma\xi_i}(2+\xi_i\gamma)}{(1+\eta\xi_i^2 e^{\gamma\xi_i})^2}\right). \tag{4.31}$$

Following [182], we construct the first moment vector (g_t) and second moment vector ($\sqcap_t$) as follows:

$$g_t = \lambda_1 g_{t-1} + (1-\lambda_1)\frac{\partial f(\beta_t)}{\partial \beta_t}. \tag{4.32}$$

$$\sqcap_t = \lambda_2 \sqcap_{t-1} + (1-\beta_2)\left(\frac{\partial f(\beta_t)}{\partial \beta_t}\right)^2. \tag{4.33}$$

Here, λ_1 and λ_2 represent the decay rates for the first and second-moment estimates, commonly set to 0.9 and 0.999, respectively. Next, we compute the bias-corrected first and second moment estimates using:

$$\hat{g}_t = \frac{g_t}{(1-\lambda_1^t)}, \tag{4.34}$$

$$\hat{\sqcap}_t = \frac{\sqcap_t}{(1-\lambda_2^t)}. \tag{4.35}$$

Finally, we update the parameter β as follows:

$$\beta_t = \beta_{t-1} - \alpha \frac{\hat{g}_t}{\sqrt{\hat{\sqcap}_t} + \epsilon}. \tag{4.36}$$

Here, $\epsilon > 0$ is a small constant, and α represents the learning rate.

Note: Algorithm 4.3 outlines the details of calculating β of the proposed Wave-RVFL model.

Computational Complexity

The time complexity of the proposed Wave-RVFL model is influenced by two main components: (i) hidden layer generation and (ii) the application of the Adam algorithm for calculating β. According to [17], the complexity of generating the hidden layer is $\mathcal{O}(MLN)$, where M is the number of training samples, L is the number of features, and N represents the number of hidden layer nodes. In line with [71], the Adam algorithm's computational complexity primarily arises from computing the moving averages of the gradients' first and second moments, along with bias correction for these averages. This step requires a complexity of $\mathcal{O}(s)$, where s is the number of samples selected per iteration. Therefore, the total time complexity of the proposed Wave-RVFL model is $\mathcal{O}(Itr(MLN + s))$, where Itr represents the number of iterations used in the Adam algorithm.

4.3.2 Experiments, Results and Discussion

This section presents comprehensive details of the experimental setup, datasets, and compared models. Subsequently, we delve into the experimental results and conduct statistical analyses. We also examine the influence of noise and outliers on the performance of the proposed Wave-RVFL. At last, we conduct sensitivity analyses for various hyperparameters of the proposed Wave-RVFL.

Algorithm 4.3 Wave-RVFL classifier

Input: Let $\{u_i\}_{i=1}^{M}$ be the input training dataset, and T_i be the target output. $\mathcal{C}$, η, and γ are the trade-off parameters.
Output: The Wave-RVFL parameter β.

1: Initialize: α_0 and t.
2:
3: Select s samples $\{u_i\}_{i=1}^{s}$ through uniform random sampling.
4:
5: Calculate $\frac{\partial f(\beta)}{\partial \beta}$ using (4.31).
6: Calculate g_t using (4.32).
7: Calculate $\sqcap_t$ using (4.33).
8: Calculate $\hat{g}_t$ using (4.34).
9: Calculate $\hat{\sqcap}_t$ using (4.35).
10: Update the Wave-RVFL's output layer weight β_t using (4.36).
11: Increment the iteration counter: $t = t + 1$.

Until: $|\beta_t - \beta_{t-1}| < \delta$ or $t = Itr$.
Return: β_t.

4.3.2.1 Compared Models, Datasets and Experimental Setup

We conduct comparisons among the proposed Wave-RVFL; and several benchmarks, including RVFL [1], ELM (also known as RVFL without direct link (RVFLwoDL)) [22], intuitionistic fuzzy RVFL (IF-RVFL) [32], graph embedded ELM with linear discriminant analysis (GEELM-LDA) [173], graph embedded ELM with local Fisher discriminant analysis (GEELM-LFDA) [173], minimum class variance based ELM (MCVELM) [175] and Neuro-fuzzy RVFL (NF-RVFL) whose fuzzy layer centers are generated using random-means [17].

To evaluate the efficacy of the proposed Wave-RVFL models, we employ 23 benchmark datasets from the UCI [7] repository from various domains and sizes. For example, the number of samples in the "fertility" dataset is 100, and the number of samples in the "connect_4" dataset is $67,557$.

Experimental Setup

The experimental hardware configuration includes a personal computer featuring an Intel(R) Xeon(R) Gold 6226R CPU with a clock speed of 2.90 GHz and 128 GB of RAM. The system runs on Windows 11 and utilizes Matlab2023a to run all the experiments. Following the experimental setup of [17], we tune the best hyperparamter setting and testing accuracy by employing grid search and five fold cross validation technique. Furthermore, all the hyperparameters of the baseline models are tune using the experimental setup followed in [17]. For all the models (baseline and proposed), following [17], we tune 6 activation functions: 1 denotes *Sigmoid*, 2 denotes *Sine*, 3 denotes *Tribas*, 4 denotes *Radbas*, 5 denotes *Tansig* and 6 denotes *Relu*. The regularization parameter $\mathcal{C}$ is taken from $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. The number of hidden nodes (N) is selected from a range spanning from 3 to 203, with a step size of 20. Wave loss parameters are selected from the following range $\eta = [0.1 : 0.25 : 2]$ and $\gamma = [-2 : 0.5 : 5]$. The Adam algorithm is initialized with the following parameters: starting weights $\beta_0 = 0.01$, initial learning rate α selected from the set $\{0.0001, 0.001, 0.01\}$, initial first moment $g_0 = 0.01$, initial second moment $u_0 = 0.01$, first-order exponential decay rate $\lambda_1 = 0.9$, second-order exponential decay rate $\lambda_2 = 0.999$, error tolerance $\delta = 10^{-5}$, division constant $\epsilon = 10^{-8}$, maximum number of iteration $Itr = 1000$, and mini-batch size $s = 2^5$ if number of samples in the dataset is less than 500, otherwise 2^8.

4.3.2.2 Experimental Results on UCI Dataset

In this section, we thoroughly analyze and compare the performance of the proposed Wave-RVFL model against baseline models using various statistical metrics and tests, including accuracy, rank, and Friedman test, to provide a robust comparison. Table 4.10 presents the experimental findings regarding accuracy (ACC), standard deviation (Std), and ranks.

Table 4.10: Testing accuracy, standard deviation and average rank of all the baseline models along with the proposed Wave-RVFL model.

Model →	RVFLwoDL [22]	RVFL [1]	GEELM-LDA [173]	GEELM-LFDA [173]	MCVELM [175]	IFRVFL [32]	NF-RVFL [17]	Wave-RVFL†
Dataset ↓	ACC ± Std	ACC ± Std	ACC ± Std	ACC ± Std	ACC ± Std	ACC ± Std	ACC ± Std	ACC ± Std
acute_inflammation	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0
adult	83.8725 ± 0.1637	84.0342 ± 0.1941	83.422 ± 1.7111	83.1784 ± 2.8045	83.9011 ± 0.3574	83.5438 ± 1.7036	83.2992 ± 0.5117	76.3544 ± 0.4048
bank	89.4934 ± 0.3245	89.4051 ± 0.5132	89.6042 ± 0.6524	46.4966 ± 11.7491	89.6703 ± 0.5543	89.1173 ± 0.7047	89.4269 ± 0.6225	88.5202 ± 0.5503
blood	76.9056 ± 13.1203	76.5065 ± 14.5247	67.1732 ± 26.7687	76.2398 ± 14.985	77.3065 ± 13.2094	77.4398 ± 14.3189	77.838 ± 11.8577	76.5065 ± 14.7975
breast_cancer	70.1754 ± 44.6249	70.1754 ± 44.6249	89.8246 ± 22.753	84.5796 ± 22.9315	70.5263 ± 44.4418	71.9298 ± 28.2614	72.3351 ± 21.0696	92.6316 ± 16.4763
breast_cancer_wisc_prog	80.3846 ± 8.7784	81.359 ± 4.3621	62.0897 ± 4.3205	62.8462 ± 26.7954	81.8718 ± 5.2731	78.359 ± 7.4479	83.359 ± 4.4786	79.3462 ± 6.3219
congressional_voting	63.2184 ± 2.2989	63.6782 ± 4.4219	59.7701 ± 7.6676	54.2529 ± 3.671	63.4483 ± 2.3556	58.8506 ± 5.7125	63.908 ± 4.1919	62.5287 ± 3.4097
conn_bench_sonar_mines_rocks	60.5226 ± 5.489	62.079 ± 9.017	80.5343 ± 24.2302	73.6585 ± 36.1107	64.4251 ± 7.4481	54.8316 ± 6.5215	65.8885 ± 16.4039	94.6341 ± 11.9984
connect_4	75.4059 ± 3.8174	75.4518 ± 3.7595	75.4518 ± 0.6034	75.4281 ± 2.1291	75.4459 ± 3.7962	75.3407 ± 1.3324	75.4844 ± 3.9187	76.3911 ± 3.7587
fertility	91 ± 8.2158	91 ± 6.5192	69 ± 32.8634	68 ± 12.0416	91 ± 6.5192	92 ± 6.7082	91 ± 7.4162	90 ± 7.9057
haberman_survival	73.8181 ± 8.5202	73.4902 ± 8.4751	55.5632 ± 4.4362	52.2898 ± 5.5721	73.4902 ± 8.4751	75.1348 ± 6.8822	75.1296 ± 7.1747	76.4738 ± 9.2421
hepatitis	82.5806 ± 10.3528	85.1613 ± 10.6011	85.1613 ± 9.5693	85.8065 ± 7.4264	85.1613 ± 9.5693	85.8065 ± 6.6892	85.8065 ± 13.0236	84.5161 ± 8.3498
ilpd_indian_liver	72.2149 ± 4.2892	71.5311 ± 5.5959	61.7521 ± 8.3776	61.4353 ± 8.5934	72.5508 ± 5.9868	72.7277 ± 5.9611	72.3932 ± 5.6986	73.2508 ± 5.3362
magic	78.5279 ± 15.2244	78.7487 ± 15.4047	77.4289 ± 0.9713	78.3579 ± 2.8589	78.7592 ± 15.7559	77.5289 ± 16.2896	76.3407 ± 7.31	95.1682 ± 10.7894
molec_biol_promoter	74.632 ± 8.0443	72.7706 ± 7.6472	71.8182 ± 20.158	77.4892 ± 15.167	72.684 ± 5.8889	78.3983 ± 7.4359	82.0346 ± 5.3415	92.381 ± 7.0367
musk_1	69.7675 ± 7.941	72.0614 ± 2.3802	72.4496 ± 29.6161	65.9518 ± 18.3889	70.1776 ± 5.1064	71.864 ± 7.1741	75.2149 ± 6.5728	96.6316 ± 7.532
parkinsons	80.5128 ± 19.5781	80.5128 ± 17.7276	84.1026 ± 19.3075	83.5897 ± 16.3782	83.5897 ± 18.1853	78.4615 ± 7.3871	83.0769 ± 12.6398	87.1795 ± 12.4299
planning	71.3814 ± 8.8534	71.3814 ± 8.8534	59.1592 ± 23.3447	61.1261 ± 22.3356	73.048 ± 8.5221	69.8048 ± 9.2549	73.018 ± 8.381	73.6036 ± 11.0985
ringnorm	51.5405 ± 1.191	51.5541 ± 1.4357	51.5541 ± 7.0865	51.5784 ± 1.7619	51.9459 ± 1.4217	51.473 ± 1.2444	51.6081 ± 0.5721	51.8649 ± 1.437
spambase	87.0472 ± 4.8721	88.546 ± 4.5479	88.9582 ± 6.8046	88.546 ± 3.5979	87.4382 ± 5.3044	85.0883 ± 7.8931	88.9582 ± 4.6103	99.3913 ± 1.3611
spect	66.7925 ± 6.3425	68.3019 ± 5.2356	63.3962 ± 7.9604	62.6415 ± 9.8403	67.9245 ± 8.7487	68.3019 ± 9.6577	69.0566 ± 10.0373	66.0377 ± 9.5278
spectf	79.7205 ± 20.7374	79.3431 ± 20.8936	80.0978 ± 20.0899	80.8526 ± 18.722	79.7205 ± 20.7374	79.3431 ± 20.8936	79.7205 ± 20.7374	81.9846 ± 16.8489
statlog_heart	80 ± 3.3127	80.3704 ± 2.8085	73.7037 ± 4.0147	74.8148 ± 1.6563	82.5926 ± 1.0143	81.8519 ± 4.2229	81.4815 ± 3.7037	80.3704 ± 1.0143
Average (ACC ± Std)	76.5006 ± 8.9605	76.8462 ± 8.6758	74.0007 ± 12.3177	71.7026 ± 11.5442	77.2469 ± 8.6379	76.3999 ± 7.9868	78.1034 ± 7.6641	**82.4246 ± 7.2881**
Average Rank	5.2826	4.6304	5.2609	5.7609	3.7391	4.9348	3.2391	**3.1522**

Here, † indicates the proposed model. Boldface denotes the best-performed model.

Table 4.11: Win Tie Loss test for the results tabulated in Table 4.10.

	RVFLwoDL [22]	RVFL [1]	GEELM-LDA [173]	GEELM-LFDA [173]	MCVELM [175]	IFRVFL [32]	NF-RVFL [17]
RVFL [1]	[12, 5, 6]						
GEELM-LDA [173]	[10, 1, 12]	[7, 4, 12]					
GEELM-LFDA [173]	[9, 1, 13]	[7, 2, 14]	[9, 1, 13]				
MCVELM [175]	[18, 3, 2]	[12, 4, 7]	[14, 2, 7]	[15, 2, 6]			
IFRVFL [32]	[10, 1, 12]	[8, 3, 12]	[12, 1, 10]	[13, 2, 8]	[9, 1, 13]		
NF-RVFL [17]	[17, 3, 3]	[19, 2, 2]	[14, 2, 7]	[16, 2, 5]	[12, 3, 8]	[15, 2, 6]	
Wave-RVFL†	[15, 1, 7]	[13, 3, 7]	[19, 1, 3]	[20, 1, 2]	[12, 1, 10]	[15, 1, 7]	[13, 1, 9]

Accuracy and standard deviation: Our analysis based on ACC reveals that the proposed Wave-RVFL model consistently outperforms several baseline models, including RVFLwoDL, RVFL, GEELM-LDA, GEELM-LFDA, MCVELM, IFRVFL, and NF-RVFL, across most datasets. From Table 4.10, it's clear that our proposed Wave-RVFL model achieves the highest average ACC at 82.4246%. In contrast, the average ACC of the RVFLwoDL, RVFL, GEELM-LDA, GEELM-LFDA, MCVELM, IFRVFL, and NF-RVFL models are 76.5006%, 76.8462%, 74.0007%, 71.7026%, 77.2469%, 76.3999%, and 78.1034%, respectively. The proposed Wave-RVFL model demonstrates minimal standard deviation values compared to the baseline models, indicating a high level of prediction certainty. On the breast_cancer dataset, our proposed Wave-RVFL achieves an ACC of 92.6316%, showcasing exceptional performance compared to the baseline models, with an increase of around 22% compared to the RVFL model. On the

conn_bench_sonar_mines_rocks dataset, Wave-RVFL outperforms the baseline models by securing the top position with an increase of 14.10%. Similar observations are recorded for the other datasets as well.

Statistical rank: Table 4.10 reports the average ranks of all models across the datasets. The proposed Wave-RVFL model achieves the lowest average rank of 3.1522, followed by NF-RVFL (3.2391) and MCVELM (3.7391). The average ranks of RVFLwoDL, RVFL, GEELM-LDA, GEELM-LFDA, and IFRVFL are 5.2826, 4.6304, 5.2609, 5.7609, and 4.9348, respectively. The lowest rank obtained by Wave-RVFL indicates its superior performance and stronger generalization capability compared to the other models.

Friedman test: For $q = 8$ models and $P = 23$ datasets, the Friedman statistics are $\chi_F^2 = 26.7289$ and $F_F = 4.3795$. At the 5% significance level, the critical value is $F_F(7, 154) = 2.0695$. Since $4.3795 > 2.0695$, the null hypothesis is rejected, indicating that statistically significant differences exist among the compared models.

Win Tie Loss test: Next, we utilize the pairwise Win Tie Loss (W-T-L) test to assess the performance of the proposed model compared to baseline models. Table 4.11 presents a comparative analysis of the proposed Wave-RVFL model alongside the baseline models. It delineates their performance regarding pairwise wins, ties, and losses across UCI datasets. In Table 4.11, the entry $[x, y, z]$ indicates the frequency with which the model listed in the row wins x times, ties y times and loses z times when compared to the model listed in the corresponding column. In Table 4.11, the W-T-L outcomes of the models are evaluated pairwise. The proposed Wave-RVFL model has achieved 15 wins (against RVFLwoDL), 13 wins (against RVFL), 19 wins (against GEELM-LDA), 20 wins (against GEELM-LFDA), 12 wins (against MCVELM), 15 wins (against IFRVFL), and 13 wins (against NF-RVFL) out of 23 datasets. Thus, this indicates that the proposed Wave-RVFL model achieves superior performance compared to the baseline models.

Table 4.12: Performance of the proposed Wave-RVFL along with the RVFL with varying levels of outliers and noise.

	Experiments with outliers			Experiments with noise		
Dataset	Outliers	RVFL [1]	Wave-RVFL†	Noise	RVFL [1]	Wave-RVFL†
blood	5%	45.685	**73.1615**	5%	44.868	**68.3463**
	10%	50.5647	**71.6886**	10%	71.6716	**76.2398**
	15%	51.2161	**69.0112**	15%	50.7714	**76.2398**
	20%	52.9333	**65.1302**	20%	57.3477	**69.7969**
	Average	50.0998	**69.7479**	Average	56.1647	**72.6557**
magic	5%	54.3428	**63.4437**	5%	**69.0799**	65.7992
	10%	37.7077	**70.2103**	10%	66.6404	**84.837**
	15%	60.3207	**60.4048**	15%	68.0389	**70.0053**
	20%	50.857	**59.2429**	20%	**65.5363**	64.837
	Average	50.8071	**63.3254**	Average	67.3239	**71.3696**
Overall Average		50.4534	**66.5367**		61.7443	**72.0127**
Average Rank		2	**1**		1.75	**1.25**

Boldface in each row denotes the best-performed model.

Considering the above discussion based on accuracy, rank, and statistical tests, we can conclude that the proposed Wave-RVFL model showcases superior and robust performance against the baseline models.

4.3.2.3 Robustness Evaluation of the Proposed Wave-RVFL Model on Datasets with Outliers

To assess the robustness of the proposed Wave-RVFL model against outliers, we introduce label noise into the blood and magic datasets at levels of 5%, 10%, 15%, and 20% to create outliers. This allowed a detailed analysis of the model's performance under adverse conditions, with results presented in Table 4.12.

[1] ***"blood" Dataset***: The proposed Wave-RVFL model consistently outperformed the baseline models across all outlier levels, from 0% to 20%. It achieved the highest average accuracy for the blood dataset at 69.7479%, demonstrating remarkable robustness.

[2] ***"magic" Dataset***: The Wave-RVFL model led the performance table in 4 out of 4 scenarios for the magic dataset. It recorded the highest average accuracy of 63.3254%, showcasing its effectiveness.

[3] ***Overall Performance***: The Wave-RVFL model achieved an overall average accuracy of 63.5367%, approximately 16% higher than the RVFL model. It also attained the lowest overall average rank of 1, indicating its superior performance to the RVFL model.

These findings illustrate that the proposed Wave-RVFL model displays exceptional robustness in challenging conditions.

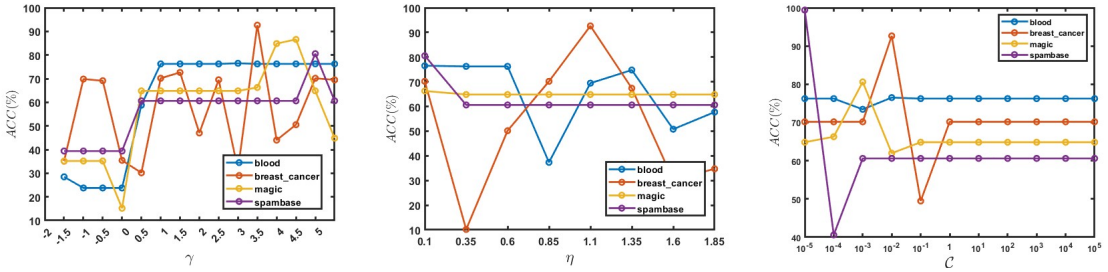
4.3.2.4 Robustness Evaluation of the Proposed Wave-RVFL Model on Datasets with Gaussian Noise

To evaluate the robustness of the proposed Wave-RVFL model against noise, we introduce Gaussian noise into the magic and blood datasets at levels of 5%, 10%, 15%, and 20%. This approach provided a thorough comparison of the model's performance under adverse conditions against the RVFL, as shown in Table 4.12.

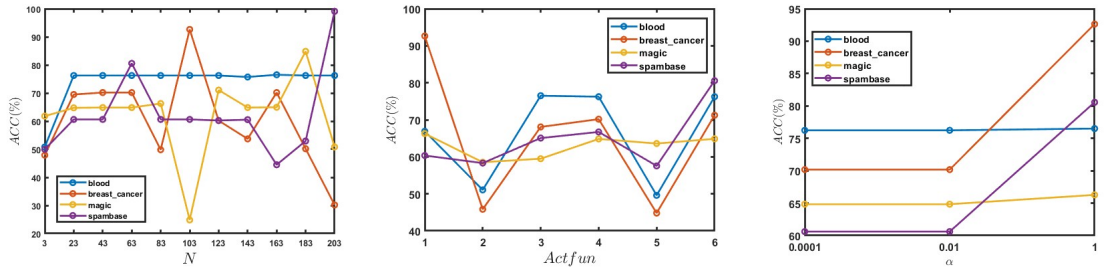
[1] ***"blood" Dataset***: The Wave-RVFL model demonstrated exceptional robustness, consistently outperforming the baseline models across all outlier levels from 0% to 20%. It achieved the highest average accuracy of 72.6557% for the blood dataset.

[2] ***"magic" Dataset***: Similarly, the Wave-RVFL model showcased its effectiveness by recording the highest average accuracy of 71.3696% under varying levels of label noise for the magic dataset.

[3] ***Overall Performance***: The Wave-RVFL model's overall average accuracy is 72.0127%, approximately 10% higher than the RVFL model. Furthermore, it attains



(a) γ v/s ACC (b) η v/s ACC (c) $\mathcal{C}$ v/s ACC



(d) N v/s ACC (e) $Actfun.$ v/s ACC (f) α v/s ACC

Figure 4.11: Effect of hyperparameters on the proposed Wave-RVFL model.

the lowest overall average rank of 1.25, underscoring its superior performance among all tested models.

These observations show that the proposed Wave-RVFL model exhibits notable insensitivity to noise, demonstrating superior robustness compared to the baseline models in challenging scenarios.

4.3.2.5 Sensitivity Analyses

To comprehensively understand the robustness of the proposed Wave-RVFL model, it is essential to analyze its sensitivity to various hyperparameters. Therefore, we conduct sensitivity analyses focusing on the following aspects: (i) γ v/s ACC, (ii) η v/s ACC, (iii) $\mathcal{C}$ v/s ACC, (iv) N v/s ACC, (v) $Actfun$ v/s ACC, and (vi) α v/s ACC. We do experiment with varying ranges of each hyperparameter and assessed their influence on the model's performance using four datasets: blood, breast_cancer, magic, and spambase. Insights from

these analyses and from their corresponding Figure 4.11 are detailed below.

(i) ***Effect of the Wave loss hyperparameter*** γ: Analysis of Figure 4.11(a) indicates that increasing γ generally enhances the model's performance up to a certain point. Optimal results are often observed when γ is between 1 and 4. Thus, we recommend γ to be varying in $[1, 4]$ for best performance.

(ii) ***Effect of the Wave loss hyperparameter*** η: From Figure 4.11(b), it is evident that the model's performance improves as η increases up to approximately 1.35, beyond which performance declines. Therefore, for optimal results, η should be set below 1.35.

(iii) ***Effect of the regularization parameter*** $\mathcal{C}$: Examination of Figure 4.11(c) reveals that performance varies with $\mathcal{C}$ values, peaking around $\mathcal{C} = 1$. Beyond this point, changes in $\mathcal{C}$ do not significantly impact performance. Consequently, we recommend $\mathcal{C}$ within the range of 10^{-5} to 1 for optimal accuracy.

(iv) ***Effect of the hyperparameter*** N ***(number of hidden nodes)***: As observed in Figure 4.11(d), the impact of N on performance is dataset-dependent. Therefore, it is advisable to fine-tune N for each specific scenario.

(v) ***Effect of the hyperparameter*** $Actfun$ ***(activation function)***: Figure 4.11(e) shows that certain activation functions, specifically $Sine$ (2) and $Tansig$ (5), consistently yield lower performance across datasets. Hence, it is prudent to exclude Sine and Tansig during activation function tuning.

(vi) ***Effect of the hyperparameter*** α **(learning rate)**: According to Figure 4.11(f), the model's performance improves as the learning rate α increases. Therefore, setting α to 1 is recommended for efficient performance.

It is important to note that the performance of Wave-RVFL may vary depending on the dataset and application (in line with the No Free Lunch Theorem [182]). Therefore, tuning hyperparameters becomes crucial to improve the overall generalization performance of the proposed Wave-RVFL model.

4.4 Comparison of Wave Loss, Granular-ball Computing, and Graph Embedding

Although the proposed Granular Ball Random Vector Functional Link (GB-RVFL), Graph Embedded GB-RVFL (GE-GB-RVFL), and Wave-RVFL models are all designed to improve the robustness of randomized neural networks, they address different aspects of the learning process and therefore should not be viewed as competing approaches. Instead, these methods operate at complementary stages of the learning pipeline, namely data representation, feature representation, and optimization.

The proposed GB-RVFL improves robustness at the *data representation level*. Instead of learning directly from individual training samples, the original dataset is first transformed into a set of adaptive granular balls that summarize neighboring samples. This granulation process naturally suppresses noisy observations and isolated outliers by representing local regions through their centers and radii. Consequently, the subsequent RVFL model learns from a compact and more reliable representation of the data, leading to improved robustness and significantly reduced computational complexity.

Building upon GB-RVFL, the proposed GE-GB-RVFL introduces graph embedding to preserve the intrinsic geometric structure among the granular ball centers. Unlike GB-RVFL, which primarily focuses on robust data abstraction, GE-GB-RVFL incorporates graph regularization into the optimization process to maintain neighborhood relationships

and improve class separability. Therefore, graph embedding operates at the *feature representation level*, ensuring that the learned output weights preserve both the local manifold structure and discriminative information contained in the granular balls.

Table 4.13: Comparison between Wave Loss, Granular-ball Computing, and Graph Embedding.

Method	**Learning Stage**	**Primary Objective**	**Limitations**
Wave Loss	Optimization	Reduces the influence of noisy samples and outliers through a bounded non-convex loss function.	Requires iterative gradient-based optimization and loses the closed-form analytical solution of RVFL.
Granular-ball Computing	Data Representation	Suppresses noise by representing neighboring samples with adaptive granular balls while reducing computational complexity.	Does not explicitly preserve geometric relationships among granular balls.
Graph Embedding	Feature Representation	Preserves intrinsic manifold structure and improves class separability through graph regularization.	Requires graph construction and increases computational complexity.
Integrated Framework (Future Work)	Representation + Optimization	Combines robust data abstraction, geometry preservation, and robust optimization.	Requires solving a more challenging non-convex graph-regularized optimization problem.

In contrast, the proposed Wave-RVFL addresses robustness from a fundamentally different perspective by modifying the optimization objective itself. Instead of changing the data representation or preserving geometric relationships, Wave-RVFL replaces the conventional squared loss function with the Wave loss function, which is smooth, bounded, and non-convex. Owing to its bounded penalty, the Wave loss reduces the influence of samples with large prediction errors, thereby limiting the effect of noisy observations and outliers

during optimization. Consequently, Wave-RVFL improves robustness at the *optimization level* while preserving the original data representation used by RVFL.

Therefore, although all three approaches improve robustness, they achieve this objective through different mechanisms. Granular-ball computing removes the adverse influence of noise before learning, graph embedding preserves the geometric relationships among the learned representations during training, whereas the Wave loss function reduces the influence of difficult or corrupted samples during optimization. Since these mechanisms are complementary rather than mutually exclusive, they can potentially be integrated within a unified randomized neural network framework.

Specifically, the optimization objective of GE-GB-RVFL currently employs the conventional squared loss while incorporating graph regularization over the granular-ball representation. A natural extension would be to replace the squared loss term with the Wave loss function, resulting in a unified optimization framework that simultaneously exploits (i) robust data abstraction through granular-ball computing, (ii) geometry-preserving representation learning through graph embedding, and (iii) robust optimization through the Wave loss function. Such a framework could potentially provide enhanced robustness against noise and outliers while preserving the intrinsic geometric structure of the data. However, owing to the non-convex nature of the Wave loss function, the resulting optimization problem would require iterative gradient-based optimization instead of the analytical closed-form solution employed by GE-GB-RVFL. Consequently, although such an integration is promising, it introduces additional computational complexity and is therefore left as an interesting direction for future research. Summary is given in Table 4.13.

4.5 Summary

This chapter presents robust and scalable extensions of RVFLs in the presence of noise, outliers, and large-scale data. First, we proposed the granular ball RVFL (GB-RVFL), which utilizes granular balls (GBs) instead of individual training samples for classifier construction. By operating on the centers of GBs rather than the entire training dataset, the proposed GB-RVFL improves scalability and enhances robustness against noise and outliers through coarse-grained data representation. To further preserve the intrinsic geometric structure of the dataset, we introduced the graph embedded GB-RVFL (GE-GB-RVFL), which integrates granular computing with graph embedding.

Second, we proposed the Wave-RVFL model, which replaces the conventional square loss function in RVFL with the Wave loss function. The proposed formulation improves robustness against noise and outliers by preventing excessive penalization of deviations. Furthermore, the optimization problem is solved using the Adam algorithm, eliminating the need for matrix inversion and significantly improving scalability for large datasets.

Extensive experiments are conducted on benchmark datasets from the UCI and KEEL repositories, as well as large-scale NDC datasets and real-world biomedical datasets. The experimental results demonstrate that the proposed models achieve superior performance compared to baseline methods in terms of classification accuracy, robustness to noise, and scalability. Statistical analyses further confirm the effectiveness and reliability of the proposed approaches across datasets of varying sizes and domains.

Chapter 5

Geometry and Class Imbalance Aware Robust Randomized Neural Networks

Outcome of this Chapter

Publication: M. A. Ganaie*, **M. Sajid***, A. K. Malik and M. Tanveer (2024). "Graph Embedded Intuitionistic Fuzzy Random Vector Functional Link Neural Network for Class Imbalance Learning," in ***IEEE Transactions on Neural Networks and Learning Systems***, v. 35, no. 9, pp. 11671–11680, **DOI:** https://doi.org/10.1109/TNNLS.2024.3353531 [**SCI Indexed with Impact Factor.:** 9.7, Q1] * M. Sajid and M. A. Ganaie contributed equally to this work.

Source Code: https://github.com/mtanveer1/GE-IFRVFL-CIL

In the previous chapter, it was demonstrated that Wave-RVFL and GB-RVFL effectively address challenges related to robustness, computational efficiency, and scalability. However, the domain of machine learning is confronted with a crucial research area known as class imbalance learning, which presents considerable hurdles in precise classification of minority classes. This issue can result in biased models where the majority class takes precedence in the training process, leading to the underrepresentation of the minority class. The random vector functional link (RVFL) network is a widely used and effective learning model for classification due to its good generalization performance and efficiency. However, it suffers when dealing with imbalanced datasets. To overcome this limitation, we propose a novel graph embedded intuitionistic fuzzy RVFL for class imbalance learning (GE-IFRVFL-CIL) model incorporating a weighting mechanism to handle imbalanced datasets. The proposed GE-IFRVFL-CIL model offers plethora of benefits: (i) leveraging graph embedding to preserve the inherent topological structure of the datasets, (ii) employing intuitionistic fuzzy theory to handle uncertainty and imprecision in the data, (iii) and the most important, it tackles class imbalance learning.

5.1 Background

The real-world datasets often contain anomalous samples that exhibit characteristics of one class but are labeled as belonging to another; such samples are commonly referred to as outliers. The presence of noisy data, outliers, and class imbalance significantly degrades the predictive performance of machine learning models. In particular, the class imbalance (CI) problem arises when there exists a substantial disparity in the number of samples across different classes within a dataset.

The original RVFL disregards the geometrical relationship of the data while computing the final output parameters [25]. Many improved versions of the RVFL have been pro-

posed as remedies for the aforementioned problem. Li et al. [184] proposed a manifold learning-based variant of RVFL, namely discriminative manifold RVFL (DMRVFL). In order to effectively utilize the intraclass discriminative information and concurrently increase the distances between interclass samples, DMRVFL replaced the inflexible one-hot label matrix with a more flexible soft label matrix. On the one hand, DMRVFL enlarges the distance of the interclass samples. On the other hand, it makes the intraclass samples more compact. Ganaie et al. [174] presented two multiclass classifiers, namely Class-Var-RVFL and Total-Var-RVFL, which utilize both the original and the projected randomized space's dispersion of training data to optimize output layer weights. The former relies on intraclass variance minimization, while the latter employs total variance minimization. Recently, graph embedded intuitionistic fuzzy weighted RVFL (GE-IFWRVFL) [35] was proposed by incorporating subspace learning (SL) criteria within the graph embedding (GE) framework with RVFL. Although GE-IFWRVFL brings forth many benefits, such as it preserves the geometrical property of the dataset and IF assists the model in handling outliers and noisy data samples; however, GE-IFWRVFL fails to deal with the imbalanced datasets.

In reality, patterns from the minority class can be more significant than patterns from other classes. Conventional RVFL treats each sample uniformly irrespective of its belongingness to minority or majority classes, and as a result, RVFL is biased in favor of the majority class samples. Thus, CI issues hinder the performance of the RVFL model. The improved fuzziness-based RVFL (IF-RVFL) [37] demonstrated its generalization performance in real-life liver disease imbalanced datasets. Class-specific weighted RVFL (CSWRVFL) [38] handles the dilemma of imbalanced datasets and carried out tests to identify the power quality disturbance. Although RVFL has made significant progress wherein IFRVFL [32] deals with noise and outliers and CSWRVFL [38] and IF-RVFL [37] retains the topological relationship of the data, there has been relatively little improvement in dealing with imbal-

ance datasets. At the time of conducting this study, to the best of our knowledge, no existing model within the RVFL family simultaneously addressed the challenges of noise and outliers, preserved the inherent geometric structure of the data, and effectively handled class imbalance in a unified framework. To bridge this gap, we propose a novel graph embedded intuitionistic fuzzy random vector functional link network for class imbalance learning (GE-IFRVFL-CIL). The GE framework, IF membership functions and CI weighting enable the proposed GE-IFRVFL-CIL model to address the imbalance issue along with noise, outliers, and geometrical structure preservation of the dataset.

This chapter's primary highlights are as follows:

[1] We propose a generic framework that can adapt to different criteria used to handle imbalance scenarios. To demonstrate the efficacy, the proposed GE-IFRVFL-CIL model incorporates a weighting scheme [6] to address the CI issue. The weighting scheme takes care of the minority class by assigning unit weightage to its samples, and the weights of the majority class are lowered by the ratio of the number of positive class samples to the number of negative class samples.

[2] To compute the output weights, the proposed GE-IFRVFL-CIL model integrates SL criteria, utilizing both intrinsic and penalty SL within the GE framework. Under the GE framework, linear discriminant analysis (LDA) [192] and local Fisher discriminant analysis (LFDA) [193] are employed, yielding two alternative combinations of models. The GE, along with the utilization of a regularization term, seeks to maintain graph structural information in the projected space.

[3] The IF membership technique is employed in the proposed GE-IFRVFL-CIL model to deal with noisy and outlier samples in the datasets. The degree of membership and non-membership are assigned to each sample in the IF concept to determine if it is a

pure sample, noise, or outlier, and data samples are weighted accordingly.

[4] We demonstrate the efficacy of the proposed GE-IFRVFL-CIL models over the KEEL imbalanced datasets (with and without Gaussian noise) from various domains, and diagnosis the Alzheimer's disease using the ADNI dataset. Experiments confirmed that the proposed GE-IFRVFL-CIL model outperforms numerous state-of-the-art models for CIL.

5.2 The Proposed Graph Embedded Intuitionistic Fuzzy RVFL for Class Imbalance Learning (GE-IFRVFL-CIL)

This section provides a detailed description of the proposed GE-IFRVFL-CIL model.

5.2.1 Proposed Method

We first define the generic mathematical framework of the proposed model to handle the class imbalance issue. The proposed GE-IFRVFL-CIL model addresses the class imbalance issue by incorporating a weighting scheme based on the class imbalance ratio. The optimization process for calculating network output weights integrates subspace learning (SL) criteria, leveraging both intrinsic and penalty SL within the GE framework. To preserve the geometrical structure, LDA [192] and LFDA [193] are incorporated under the GE framework, along with a GE regularization parameter. Additionally, IF numbers are assigned to each training sample for handling noise and outliers. The optimization problem of the

proposed GE-IFRVFL-CIL model is defined as follows:

$$\begin{aligned} \underset{\beta}{\text{minimize}} \;\; & \frac{1}{2}\|\beta\|_2^2 + \frac{\theta}{2}\left\|U_{uni}^{\frac{1}{2}}\beta\right\|_2^2 + \frac{\mathcal{C}}{2}(d_+)\left\|\mathcal{S}_+^{\frac{1}{2}}\xi_+\right\|_2^2 + \frac{\mathcal{C}}{2}(d_-)\left\|\mathcal{S}_-^{\frac{1}{2}}\xi_-\right\|_2^2, \\ s.t. \;\; & Z_+\beta = T_+ - \xi_+ \text{ and } Z_-\beta = T_- - \xi_-. \end{aligned} \tag{5.1}$$

Here, β is the weight matrix connecting the hidden layer and the input layer to the output layer, θ is a graph regularization parameter, U_{uni} is the graph embedding matrix defined in Section 2.9, $\mathcal{S}_-$ ($\mathcal{S}_+$) is the diagonal matrix with IF scores along its diagonal for the negative (positive) class samples. ξ_- (ξ_+) is the error of the negative (positive) class samples. Here, d_- (d_+) is the weighting scheme of the negative (positive) class defined in Subsection 5.2.2. $\mathcal{C} \in \mathbb{R}^+$ is the regularization parameter to penalize the error variables. Z_- (Z_+) denotes the non-linear as well as linear projection of the negative (positive) class samples, defined as:

$$Z_- = \begin{bmatrix} U_- & H_{neg} \end{bmatrix} \text{ and } Z_+ = \begin{bmatrix} U_+ & H_{pos} \end{bmatrix}, \tag{5.2}$$

where U_- (U_+) is the negative (positive) class' input matrix and H_{neg} (H_{pos}) is the negative (positive) class' hidden layer matrix obtained by transforming U_- (U_+) with the help of randomly initialized weights and biases followed by the non-linear activation function ψ. T_- (T_+) is the target matrix for the negative (positive) class samples.

The objective function in (5.1) comprises four key components:

1. Structural Risk Minimization (SRM): The first term aims to minimize the norm of the parameter β, adhering to the principle of SRM.

2. Graph Embedding Regularization: The second term allocates GE weights to β and minimizes the resultant norm, promoting a structured relationship between samples.

3. Positive Class Enhancement: The third term is designed for the positive class and assigns IF weights to the positive class error. This component considers the CI by incorporating the

d_+ (positive CI weights) to each sample, with the weights being higher for minority class samples.

4. Negative Class Enhancement: The fourth term is analogous to the third but pertains to the negative class. It assigns IF weights to the negative class error, incorporating the d_- (negative CI weights), which favor minority class samples.

Hence, the introduction of CI weights (d_+ and d_-) effectively balances the influence of both majority and minority class samples during training, leading to improved learning compared to existing baseline models.

The Lagrangian of (5.1) is given as follows:

$$L = \frac{1}{2}\|\beta\|_2^2 + \frac{\theta}{2}\left\|U_{uni}^{\frac{1}{2}}\beta\right\|_2^2 + \frac{\mathcal{C}}{2}(d_+)\left\|\mathcal{S}_+^{\frac{1}{2}}\xi_+\right\|_2^2 + \frac{\mathcal{C}}{2}(d_-)\left\|\mathcal{S}_-^{\frac{1}{2}}\xi_-\right\|_2^2 - \alpha_+^T(Z_+\beta - T_+ + \xi_+) - \alpha_-^T(Z_-\beta - T_- + \xi_-), \tag{5.3}$$

where α_+ and α_- are the Lagrange multipliers. Let

$$l_+ = \mathcal{C} \times d_+ \text{ and } l_- = \mathcal{C} \times d_-. \tag{5.4}$$

Rewriting (5.3), we have

$$L = \frac{1}{2}\|\beta\|_2^2 + \frac{\theta}{2}\left\|U_{uni}^{\frac{1}{2}}\beta\right\|_2^2 + \frac{l_+}{2}\left\|\mathcal{S}_+^{\frac{1}{2}}\xi_+\right\|_2^2 + \frac{l_-}{2}\left\|\mathcal{S}_-^{\frac{1}{2}}\xi_-\right\|_2^2 - \begin{bmatrix}\alpha_+^T & \alpha_-^T\end{bmatrix}\left(\begin{bmatrix}Z_+\\ Z_-\end{bmatrix}\beta - \begin{bmatrix}T_+\\ T_-\end{bmatrix} + \begin{bmatrix}\xi_+\\ \xi_-\end{bmatrix}\right). \tag{5.5}$$

By applying K.K.T. conditions to (5.5), we get

$$\beta + \theta U_{uni}\beta - (Z_+^T\alpha_+ + Z_-^T\alpha_-) = 0, \tag{5.6}$$

$$l_+\mathcal{S}_+\xi_+ - \alpha_+ = 0, \tag{5.7}$$

$$l_-\mathcal{S}_-\xi_- - \alpha_- = 0, \tag{5.8}$$

$$Z_+\beta - T_+ + \xi_+ = 0, \tag{5.9}$$

$$Z_-\beta - T_- + \xi_- = 0. \tag{5.10}$$

Rewriting the equations (5.6), (5.7) and (5.8), we have

$$(I + \theta U_{uni})\beta = (Z_+^T\alpha_+ + Z_-^T\alpha_-), \tag{5.11}$$

$$l_+\mathcal{S}_+\xi_+ = \alpha_+, \tag{5.12}$$

$$l_-\mathcal{S}_-\xi_- = \alpha_-, \tag{5.13}$$

where I is an identity matrix of conformal dimensions. Using (5.12) and (5.13) in (5.11), we get

$$(I + \theta U_{uni})\beta = l_+Z_+^T\mathcal{S}_+\xi_+ + l_-Z_-^T\mathcal{S}_-\xi_-, \tag{5.14}$$

$$(I + \theta U_{uni})\beta = l_+Z_+^T\mathcal{S}_+(T_+ - Z_+\beta) + \ l_-Z_-^T\mathcal{S}_-(T_- - Z_-\beta), \tag{5.15}$$

$$(I + \theta U_{uni} + l_+Z_+^T\mathcal{S}_+Z_+ + l_-Z_-^T\mathcal{S}_-Z_-)\beta = l_+Z_+^T\mathcal{S}_+T_+ + l_-Z_-^T\mathcal{S}_-T_-. \tag{5.16}$$

Multiply $(\frac{1}{l_+} + \frac{1}{l_-})$ to (5.16), we have

$$\begin{aligned}&\left(\frac{1}{l_+} + \frac{1}{l_-}\right)\left(I + \theta U_{uni} + l_+Z_+^T\mathcal{S}_+Z_+ + l_-Z_-^T\mathcal{S}_-Z_-\right)\beta \\ &= \left(\frac{1}{l_+} + \frac{1}{l_-}\right)\left(l_+Z_+^T\mathcal{S}_+T_+ + l_-Z_-^T\mathcal{S}_-T_-\right),\end{aligned} \tag{5.17}$$

$$\left(\left(\frac{1}{l_+}+\frac{1}{l_-}\right)\left(I+\theta U_{uni}\right)+Z_+^T\mathcal{S}_+Z_++\frac{l_-}{l_+}Z_-^T\mathcal{S}_-Z_-+\frac{l_+}{l_-}Z_+^T\mathcal{S}_+Z_++Z_-^T\mathcal{S}_-Z_-\right)\beta$$
$$=Z_+^T\mathcal{S}_+T_++\frac{l_-}{l_+}Z_-^T\mathcal{S}_-T_-+\frac{l_+}{l_-}Z_+^T\mathcal{S}_+T_++Z_-^T\mathcal{S}_-T_-, \tag{5.18}$$

$$\beta=\left(\left(\frac{1}{l_+}+\frac{1}{l_-}\right)\left(I+\theta U_{uni}\right)+Z_+^T\mathcal{S}_+Z_++\frac{l_-}{l_+}Z_-^T\mathcal{S}_-Z_-+\frac{l_+}{l_-}Z_+^T\mathcal{S}_+Z_++Z_-^T\mathcal{S}_-Z_-\right)^{-1}$$
$$\left(Z_+^T\mathcal{S}_+T_++\frac{l_-}{l_+}Z_-^T\mathcal{S}_-T_-+\frac{l_+}{l_-}Z_+^T\mathcal{S}_+T_++Z_-^T\mathcal{S}_-T_-\right), \tag{5.19}$$

$$\beta=\left(\left(\frac{1}{l_+}+\frac{1}{l_-}\right)\left(I+\theta U_{uni}\right)+(1+\frac{l_+}{l_-})Z_+^T\mathcal{S}_+Z_+\ +(1+\frac{l_-}{l_+})Z_-^T\mathcal{S}_-Z_-\right)^{-1}$$
$$\left((1+\frac{l_+}{l_-})Z_+^T\mathcal{S}_+T_+\ +(1+\frac{l_-}{l_+})Z_-^T\mathcal{S}_-T_-\right), \tag{5.20}$$

$$\beta=\left(\left(\frac{1}{l_+}+\frac{1}{l_-}\right)\left(I+\theta U_{uni}\right)+[Z_+^T,\ Z_-^T]\begin{bmatrix}(1+\frac{l_+}{l_-})\mathcal{S}_+ & \mathbf{0}\\ \mathbf{0} & (1+\frac{l_-}{l_+})\mathcal{S}_-\end{bmatrix}\begin{bmatrix}Z_+\\ Z_-\end{bmatrix}\right)^{-1}$$
$$\begin{bmatrix}Z_+^T & Z_-^T\end{bmatrix}\begin{bmatrix}(1+\frac{l_+}{l_-})\mathcal{S}_+ & \mathbf{0}\\ \mathbf{0} & (1+\frac{l_-}{l_+})\mathcal{S}_-\end{bmatrix}\begin{bmatrix}T_+\\ T_-\end{bmatrix}. \tag{5.21}$$

Using equation (5.21), we find the output layer's weight matrix.

Consequent Sections discuss LDA and LFDA under the GE framework and the class imbalance weighting scheme.

5.2.2 Class Imbalance Weighting Scheme [6]

Let l_n (l_p) denote the number of samples of the negative (positive) class.

$$d_+ = 1, \text{ if } u_i \text{ is in the positive class,} \tag{5.22}$$

$$d_- = \frac{l_p}{l_n}, \text{ if } u_i \text{ is in the negative class.} \tag{5.23}$$

In this weighting method, the sample of the minority class is given unit weights, and the weight of the majority class is reduced by a factor equal to the proportion of positive class samples to negative class samples.

In this way, the proposed GE-IFRVFL-CIL model reduces the influence of majority class samples during training by correspondingly lowering their weights, effectively balancing sample weights between positive and negative classes. This CI weighting strategy empowers the minority class in training while maintaining the weights between both classes to some extent. This leads to improved parameter learning in the proposed GE-IFRVFL-CIL compared to baseline models.

5.2.3 LDA and LFDA under the GE Framework [5]

Both the intrinsic as well as penalty graphs are based on the concatenated matrix Z, *i.e.,* $\mathcal{U}^{int} = \{U, \Theta^{int}\}$ and $\mathcal{U}^{pen} = \{U, \Theta^{pen}\}$., respectively. Therefore, $U_{int} = Z^T \mathcal{L} Z$, $U_{pen} = Z^T \mathcal{U} Z$ and $U_{uni} = U_{pen}^{-1} U_{int}$. Weights for intrinsic and penalty graphs for the LDA and LFDA are given as follows:

- **Linear discriminant analysis (LDA):** The intrinsic and penalty graph weights for

the LDA model are as follows:

$$\Theta_{ij}^{int} = \begin{cases} \frac{1}{l_{t_i}}, & \text{if } t_i = t_j, \\ 0, & \text{otherwise.} \end{cases} \tag{5.24}$$

$$\Theta_{ij}^{pen} = \begin{cases} \frac{1}{M} - \frac{1}{l_{t_i}}, & \text{if } t_i = t_j, \\ \frac{1}{M}, & \text{otherwise.} \end{cases} \tag{5.25}$$

Here, l_{t_i} denotes the number of samples in the class t_i.

- **Local Fisher discriminant analysis (LFDA):** LFDA model assigns intrinsic and penalty graph weights as follows:

$$\Theta_{ij}^{int} = \begin{cases} \frac{\eta_{ij}}{l_{t_i}}, & \text{if } t_i = t_j, \\ 0, & \text{otherwise.} \end{cases} \tag{5.26}$$

$$\Theta_{ij}^{pen} = \begin{cases} \eta_{ij}\left(\frac{1}{M} - \frac{1}{l_{t_i}}\right), & \text{if } t_i = t_j, \\ \frac{1}{M}, & \text{otherwise.} \end{cases} \tag{5.27}$$

Here, $\eta_{ij} = exp(-\frac{\|z_j - z_i\|^2}{2\sigma^2})$, where $z_j,\ z_i \in Z$ and σ is the scaling parameter. The similarity between z_j and z_i in the matrix Z is measured by η_{ij}.

To effectively contend with the challenge of imbalanced datasets, we used a weighting scheme in conjunction with two distinct graph embedding techniques, namely LDA and LFDA. From now onwards, GE-IFRVFL-CIL with LDA structure is referred to as GE-IFRVFL-CIL-1, and GE-IFRVFL-CIL with LFDA structure is referred to as GE-IFRVFL-CIL-2.

5.3 Computational Complexity of the Proposed GE-IFRVFL-CIL Models

In the solution provided for GE-IFRVFL-CIL in (5.21), and following the methodology in [173], the computation of the graph embedding matrix U_{uni} considers both intrinsic and penalty graph structures. The time complexity associated with this process is $\mathcal{O}((L+h_l)^3+(L+h_l)^2M)$. As per [34], the intuitionistic fuzzy score matrix for each sample has time complexity $\mathcal{O}(M)$ and the imbalance ratio assignment requires $\mathcal{O}(M)$ complexity. Following the standard procedure for determining the time complexity of matrix inversion and multiplication of matrix, the time complexity of a matrix inversion and the four matrix multiplication in (5.21) requires a time complexity of $\mathcal{O}((L+h_l)^3+(L+h_l)^2M)$. Thus, the resultant time complexity of the proposed GE-IFRVFL-CIL model is equal to $\mathcal{O}((L+h_l)^3+(L+h_l)^2M)+\mathcal{O}(M)+\mathcal{O}(M)+\mathcal{O}((L+h_l)^3+(L+h_l)^2M) \approx \mathcal{O}((L+h_l)^3+(L+h_l)^2M)$.

5.4 Experiments and Results

The experiments are executed on a computing system possessing MATLAB R2017b software, an Intel(R) Xeon(R) CPU E5-2697 v4 processor operating at 2.30 GHz with 128-GB Random Access Memory (RAM), and a Windows-10 operating platform. To generate IF weights in IFKRR [194], IFRVFL [32], GE-IFWRVFL [35] and IFTWSVM [34] models, the Gaussian Kernel function is employed and is defined as: $K(u_1,u_2)=exp(-\frac{\|u_1-u_2\|^2}{\mu^2})$, where μ is the kernel parameter.

5.4.1 Experimental Procedure and Hyperparameter Settings

The dataset is split into a 70% portion for model training and a 30% portion for testing. We employed the grid search technique to optimize hyperparameters through five-fold cross-validation. This entails the random partitioning of the training dataset into five distinct, non-overlapping subsets. In each iteration, one subset is set aside for testing while the remaining four are utilized for training. The hyperparameters corresponding to different models are chosen using the grid search technique from the following range: the regularization parameter $\mathcal{C} = [10^{-5}, 10^{-4}, \cdots, 10^{5}]$, the number of hidden nodes $h_l = 3 : 20 : 203$, the Kernel parameter $\mu = [2^{-5}, 2^{-4}, \cdots, 2^{5}]$, the graph regularization parameter $\theta = [10^{-5}, 10^{-4}, \cdots, 10^{5}]$. $\mathcal{C}_1 = [10^{-5}, 10^{-4}, \cdots, 10^{5}]$ and $\mathcal{C}_2 = [10^{-5}, 10^{-4}, \cdots, 10^{5}]$ are the range of two regularization parameters range used for IFTWSVM. The final accuracy of each model is reported by evaluating its performance under the best hyperparameter settings on the testing datasets.

5.4.2 Evaluation on KEEL Benchmark Datasets

We demonstrate the efficacy of our proposed models in class imbalance learning by leveraging 28 standard benchmark datasets sourced from the KEEL imbalanced dataset repository [8]. These datasets encompass diverse domains and imbalance ratios (IR). Specifically, we strategically select 14 small datasets with sample sizes below 500 and 14 medium-sized datasets with sample sizes exceeding 500. For further details, including statistical insights and dataset-specific imbalance ratios, please refer to Table 5.5.

We compare the proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 models with various machine learning models, namely intuitionistic fuzzy twin support vector machine (IFTWSVM) [34], RVFL [1], intuitionistic fuzzy kernel ridge regression (IFKRR) [194], ELM [22], IFRVFL [32] and GE-IFWRVFL [35]. The area under the curve (AUC) from

Table 5.1: Statistical description of the KEEL datasets.

Dataset	#Samples	#Minority Class Samples (+1)	#Majority Class Samples (-1)	#Features	IR
abalone9-18	731	42	689	7	0.061
aus	690	307	383	14	0.8016
checkerboard_Data	690	307	383	14	0.8016
crossplane150	150	75	75	2	1
ecoli-0-1-4-6_vs_5	280	20	260	6	0.0769
ecoli-0-1_vs_5	240	20	220	6	0.0909
ecoli-0-3-4-6_vs_5	205	20	185	7	0.1081
ecoli-0-6-7_vs_5	220	20	200	6	0.1
ecoli0137vs26	311	54	257	7	0.2101
glass2	214	17	197	9	0.0863
glass5	214	9	205	9	0.0439
haber	306	81	225	3	0.36
heart-stat	270	150	120	13	1.25
led7digit-0-2-4-5-6-7-8-9_vs_1	443	37	406	7	0.0911
monk2	601	206	395	7	0.5215
new-thyroid1	215	35	180	5	0.1944
pima	768	268	500	8	0.536
shuttle-6_vs_2-3	230	10	220	9	0.0455
shuttle-c0-vs-c4	1829	123	1706	9	0.0721
transfusion	748	178	570	4	0.3123
vehicle1	846	217	629	18	0.345
vowel	988	90	898	10	0.1002
wpbc	194	46	148	33	0.3108
yeast-0-3-5-9_vs_7-8	506	50	456	8	0.1096
yeast-0-5-6-7-9_vs_4	528	51	477	8	0.1069
yeast-2_vs_4	2968	858	2110	8	0.4066
yeast3	1484	163	1321	8	0.1234
yeast5	1484	44	1440	8	0.0306

Table 5.2 shows the performance of the models in classifying data. We follow five metrics, namely average AUC, ranking scheme, Friedman test, Nemenyi post hoc test, and win-tie-loss sign test [153] based on analysis and statistical tests for an overall comparison of the models listed in Table 5.2.

Average AUC: Using table 5.2, the average AUC of the existing models, represented as (AUC, model name), are (80.76%, IFTWSVM), (70.41%, IFKRR), (79.71%, ELM), (80.31%, RVFL), (82.97%, IFRVFL) and (81.79%, GE-IFWRVFL). Whereas the AUC values of proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 models are 84.54% and 84.97%, better than that of the existing baseline models. The results show that the proposed models, GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2, have the first and second spots in terms of average AUC and the RVFL family models', *i.e.,* IFRVFL, GE-IFWRVFL, and RVFL models took third, fourth and fifth spots, respectively, followed by ELM, IFTWSVM,

and IFKRR. The proposed GE-IFRVFL-CIL-2 and GE-IFRVFL-CIL-1 models showed

Table 5.2: The classification accuracies of the proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 models along with the existing models, *i.e.,* IFTWSVM, IFKRR, ELM, RVFL, IFRVFL and GE-IFWRVFL on KEEL [8] imbalanced benchmark datasets.

Dataset	IFTWSVM [34] (AUC,C_1) (C_3, μ)	IFKRR [194] (AUC, C) (μ)	ELM [22] (AUC, h_l) ($\mathcal{C}$)	RVFL [1] (AUC, h_l) ($\mathcal{C}$)	IFRVFL [32] (AUC, h_l) ($\mathcal{C}$,μ)	GE-IFWRVFL [35] (AUC, h_l) ($\mathcal{C}$,μ,θ)	GE-IFRVFL-CIL-1† (AUC, h_l) ($\mathcal{C}$,μ,θ)	GE-IFRVFL-CIL-2† (AUC, h_l) ($\mathcal{C}$,μ,θ)
abalone9-18	(0.7502, 0.001) (10, 0.25)	(0.7022, 10) 0.125	(0.6614, 83) 100	(0.7035, 183) 100	(0.7683, 3) (100000, 0.25)	(0.8272, 23) (0.1, 0.25, 0.0001)	(**0.8319**, 23) (10000, 0.125, 0.001)	(0.8272, 63) (100000, 0.03125, 10)
aus	(0.8482, 1000) (10, 4)	(0.8023, 0.00001) 0.125	(0.8484, 203) 0.01	(0.8492, 43) 0.001	(0.8526, 43) (0.001, 32)	(0.8449, 43) (0.001, 8, 0.00001)	(0.8449, 43) (10, 1, 0.00001)	(**0.8545**, 43) (0.001, 16, 0.00001)
checkerboard_Data	(0.8482, 1000) (10, 4)	(0.8023, 0.00001) 0.125	(0.8484, 203) 0.01	(0.8492, 43) 0.001	(0.8526, 43) (0.001, 32)	(0.8449, 43) (0.001, 8, 0.00001)	(0.8449, 43) (10, 1, 0.00001)	(**0.8545**, 43) (0.001, 16, 0.00001)
crossplane150	(**1**, 0.00001) (0.00001, 0.03125)	(0.5873, 10000) 0.03125	(0.9893, 43) 100000	(0.9643, 63) 100	(0.9821, 63) (1000, 4)	(0.9821, 63) (0.1, 4, 0.1)	(0.9821, 63) (0.1, 4, 0.1)	(**1**, 63) (0.1, 8, 1)
ecoli-0-1_vs_5	(**0.8868**, 100) (0.01, 0.5)	(0.8259, 0.01) 0.25	(0.8303, 23) 1000	(0.8259, 63) 0.1	(0.8333, 63) (0.1, 8)	(0.8184, 143) (1000, 8, 0.00001)	(0.8259, 23) (1, 1, 1)	(0.8259, 23) (0.00001, 0.25, 0.1)
ecoli-0-1-4-6_vs_5	(0.9321, 100000) (0.01, 0.5)	(0.9938, 0.00001) 0.0625	(0.9963, 143) 100	(**1**, 63) 1000	(0.9753, 63) (10, 0.125)	(**1**, 63) (0.00001, 1, 0.001)	(**1**, 63) (0.01, 2, 0.001)	(**1**, 163) (0.001, 2, 0.01)
ecoli-0-3-4-6_vs_5	(0.8083, 10) (0.001, 1)	(0.825, 0.00001) 0.125	(0.795, 23) 0.1	(**0.8333**, 203) 0.01	(**0.8333**, 63) (10, 2)	(0.825, 43) (0.00001, 2, 0.001)	(**0.8333**, 43) (0.01, 32, 0.1)	(0.825, 43) (0.001, 4, 0.1)
ecoli-0-6-7_vs_5	(0.7336, 0.0001) (0.1, 0.5)	(0.7172, 0.1) 0.125	(0.8169, 23) 10	(0.75, 163) 100	(0.7923, 63) (0.1, 2)	(0.8087, 63) (0.00001, 2, 0.001)	(**0.8675**, 63) (100000, 0.25, 0.001)	(0.8347, 63) (10000, 0.25, 0.1)
ecoli0137vs26	(**0.9872**, 100000) (1000, 4)	(0.9321, 1) 0.125	(0.9529, 43) 0.01	(0.9706, 143) 1	(0.9449, 123) (0.01, 1)	(0.9706, 203) (10, 1, 0.0001)	(0.9578, 63) (10000, 0.03125, 0.00001)	(0.9706, 183) (0.0001, 2, 0.01)
glass2	(0.8065, 0.00001) (0.001, 32)	(0.5, 100000) 8	(0.7121, 83) 100	(0.8105, 83) 10000	(0.8145, 3) (10000, 0.5)	(0.5565, 203) (1, 0.03125, 0.0001)	(**0.8387**, 143) (0.001, 4, 0.0001)	(0.7258, 43) (10000, 0.5, 10)
glass5	(0.4922, 0.01) (0.001, 4)	(0.7031, 0.01) 0.25	(0.7453, 143) 1	(0.7188, 83) 1000	(0.7188, 83) (100, 8)	(0.7266, 83) (100000, 0.25, 0.00001)	(0.7813, 123) (0.01, 16, 1)	(**0.9766**, 123) (100000, 0.03125, 1)
haber	(0.5588, 1000) (0.1, 0.125)	(0.5891, 1000) 8	(0.5822, 143) 100	(0.5253, 43) 10	(0.6097, 63) (1000, 0.5)	(**0.6138**, 3) (10000, 0.0625, 0.01)	(0.6085, 3) (100000, 0.0625, 0.001)	(0.5938, 3) (100000, 0.125, 0.1)
heart-stat	(0.8398, 10) (10, 16)	(0.7866, 0.01) 0.5	(0.8653, 183) 0.0001	(0.8574, 123) 0.0001	(0.8647, 123) (100000, 0.125)	(0.8611, 3) (0.001, 4, 0.00001)	(0.8505, 3) (0.00001, 4, 0.00001)	(**0.886**, 103) (10000, 0.03125, 0.00001)
led7digit-0-2-4-5-6-7-8-9_vs_1	(0.9016, 10000) (100, 2)	(0.5, 0.00001) 0.03125	(0.9073, 63) 100	(**0.9419**, 203) 0.01	(0.9258, 143) (0.1, 4)	(0.9258, 3) (0.1, 2, 0.0001)	(0.9339, 23) (0.01, 0.03125, 0.00001)	(0.9258, 103) (0.001, 2, 0.01)
monk2	(**0.8706**, 100) (0.00001, 8)	(0.5, 1) 0.125	(0.7993, 203) 10	(0.7961, 123) 10000	(0.5593, 163) (100000, 1)	(0.7918, 123) (100000, 2, 1)	(0.8059, 123) (10000, 1, 10000)	(0.7778, 123) (10000, 1, 10000)
new-thyroid1	(0.9444, 0.0001) (0.0001, 4)	(0.9825, 10) 0.0625	(0.9947, 43) 10	(**1**, 23) 10	(0.9825, 23) (10, 0.03125)	(**1**, 23) (100, 0.5, 0.001)	(**1**, 23) (0.01, 4, 0.001)	(**1**, 23) (0.001, 4, 0.1)
pima	(0.7582, 0.001) (1, 2)	(0.6886, 1) 0.03125	(0.73, 83) 0.001	(0.7452, 23) 100	(0.7827, 3) (1000, 0.5)	(**0.8048**, 23) (1, 1, 0.001)	(0.8035, 3) (100, 0.5, 0.00001)	(0.787, 23) (100000, 0.25, 10)
shuttle-6_vs_2-3	(0.75, 0.0001) (0.0001, 2)	(0.9394, 0.00001) 0.03125	(0.9985, 23) 10	(**1**, 3) 0.1	(0.9924, 3) (0.1, 2)	(**1**, 3) (0.1, 8, 0.00001)	(**1**, 3) (0.00001, 0.03125, 0.00001)	(**1**, 3) (0.00001, 0.03125, 10)
shuttle-c0-vs-c4	(**1**, 0.0001) (0.00001, 16)	(**1**, 0.00001) 2	(0.9892, 23) 0.1	(0.9865, 43) 0.1	(0.9865, 23) (0.0001, 0.5)	(0.9865, 3) (0.1, 0.25, 0.00001)	(0.9865, 3) (0.001, 4, 0.00001)	(**1**, 3) (0.01, 2, 100)
transfusion	(0.6146, 0.001) (0.1, 8)	(0.5, 0.00001) 0.03125	(0.606, 123) 0.1	(0.6257, 23) 100	(0.6555, 163) (100000, 0.25)	(0.6382, 163) (1, 0.125, 0.0001)	(0.6245, 63) (0.1, 8, 0.0001)	(**0.6672**, 23) (0.1, 32, 0.01)
vehicle1	(0.7859, 0.001) (0.1, 16)	(0.7366, 1) 0.5	(0.7626, 163) 100	(0.7683, 83) 10	(0.7954, 203) (100000, 0.03125)	(**0.8367**, 163) (0.1, 1, 0.0001)	(0.8339, 163) (10, 1, 0.0001)	(0.7997, 83) (100000, 0.5, 0.1)
vowel	(**0.9482**, 0.00001) (0.001, 1)	(0.8279, 0.00001) 0.0625	(0.7582, 103) 100	(0.8704, 143) 100	(0.8631, 163) (100, 2)	(0.8333, 143) (1, 2, 0.1)	(0.8889, 143) (10, 1, 0.001)	(0.8704, 143) (1000, 0.5, 10000)
wpbc	(0.564, 0.01) (0.1, 32)	(0.5195, 0.1) 1	(0.5811, 143) 0.1	(0.6716, 3) 10	(**0.7356**, 203) (0.00001, 0.5)	(0.6002, 163) (1000, 1, 0.00001)	(0.6252, 163) (0.1, 4, 0.0001)	(0.6558, 3) (1000, 2, 10)
yeast-0-3-5-9_vs_7-8	(0.5574, 0.01) (1000, 16)	(0.5926, 0.1) 0.25	(0.6126, 163) 100000	(0.4926, 203) 10000	(0.5963, 163) (1000, 0.5)	(0.5444, 143) (0.01, 2, 0.001)	(**0.7**, 43) (0.00001, 1, 0.0001)	(0.5981, 3) (0.1, 1, 0.001)
yeast-0-5-6-7-9_vs_4	(0.7137, 100) (10, 16)	(0.6619, 10) 0.125	(0.6517, 123) 10	(0.6625, 163) 1	(0.7277, 203) (10000, 0.25)	(0.7108, 43) (10, 1, 0.01)	(0.7067, 3) (100, 0.5, 0.00001)	(**0.7775**, 43) (10, 0.5, 0.01)
yeast-2_vs_4	(**0.9507**, 0.01) (0.1, 16)	(0.5, 100000) 0.125	(0.8172, 103) 1	(0.7822, 63) 100	(0.915, 43) (1000, 1)	(0.8109, 23) (100000, 0.5, 0.1)	(0.8581, 203) (0.001, 8, 0.001)	(0.9326, 63) (100000, 0.03125, 100)
yeast3	(0.8175, 10000) (10000, 32)	(0.5, 0.00001) 0.0625	(0.8124, 203) 10	(0.8192, 123) 10	(0.9031, 163) (10, 0.03125)	(0.912, 183) (0.01, 0.5, 0.00001)	(0.9034, 163) (0.0001, 4, 0.00001)	(**0.9298**, 203) (100, 0.03125, 0.001)
yeast5	(0.9453, 10000) (10000, 16)	(0.5, 100000) 0.03125	(0.6542, 143) 1	(0.6655, 163) 100	(**0.9681**, 203) (1, 0.03125)	(0.8251, 183) (100, 1, 0.00001)	(0.9335, 23) (0.001, 4, 0.00001)	(0.8942, 43) (1000, 0.5, 1)
Average AUC	0.8076	0.7041	0.7971	0.8031	0.8297	0.8179	<u>0.8454</u>	**0.8497**
Average Rank	4.8036	7.0714	5.2857	4.625	3.875	4.2679	<u>3.3393</u>	**2.7321**

The boldface in each row signifies the performance of the best model. The underline in the last two rows signifies the second-best model in terms of average AUC and rank. † represents the proposed models.

around $14-15\%$ greater AUC than that of IFKRR. Thus, the overall results demonstrate the superiority of the proposed algorithms.

Ranking scheme: Table results show the average ranks of all models across the datasets. The average ranks (average rank, model) are (3.3393, GE-IFRVFL-CIL-1), (4.2679, GE-IFWRVFL), (2.7321, GE-IFRVFL-CIL-2), (4.8036, IFTWSVM), (7.0714, IFKRR), (5.2857, ELM), (4.625, RVFL), and (3.875, IFRVFL). The proposed GE-IFRVFL-CIL-2

obtains the lowest average rank of 2.7321, securing the first position among all models, while GE-IFRVFL-CIL-1 achieves an average rank of 3.3393 and secures the third position. These lower average ranks indicate the superior performance of the proposed models compared to the existing approaches.

Friedman test: For $d = 8$ models and $K = 28$ datasets, the obtained values are $\chi_F^2 = 57.5946$ and $F_F = 11.2355$. At the 5% significance level, the critical value is $F_F(7, 189) = 2.0583$. Since $11.2355 > 2.0583$, the null hypothesis is rejected, indicating that significant differences exist among the models.

Nemenyi post hoc test: The Nemenyi test yields a critical difference of $C.D. = 1.98$. The rank differences between GE-IFRVFL-CIL-2 and IFTWSVM, IFKRR, and ELM are 2.0715, 4.3393, and 2.5536, respectively, all exceeding the $C.D.$ value. Similarly, the rank difference between GE-IFRVFL-CIL-1 and IFKRR is 3.7321, which also exceeds the $C.D.$ value. These results indicate that GE-IFRVFL-CIL-2 is statistically superior to IFTWSVM, IFKRR, and ELM, while GE-IFRVFL-CIL-1 significantly outperforms IFKRR. Although the proposed models are not statistically superior to all other models, their lower average ranks demonstrate their better overall performance.

The results further show that the proposed GE-IFRVFL-CIL models achieve improved generalization performance on imbalanced datasets. The integration of the graph embedding framework, weighting technique, and intuitionistic fuzzy theory contributes to the enhanced performance of the proposed models compared to the baseline approaches.

Table 5.3: Performance of classification models on different levels of noise.

Datasets	Noise	IFTWSVM [34]	IFKRR [194]	ELM [22]	RVFL [1]	IFRVFL [32]	GE-IFWRVFL [35]	GE-IFRVFL-CIL-1†	GE-IFRVFL-CIL-2†
Average AUC	0%	0.8896	0.7747	0.8002	0.7862	0.8862	0.8534	<u>0.8949</u>	**0.9064**
	5%	0.8835	0.8022	0.7985	0.7866	0.8833	0.8576	<u>0.8986</u>	**0.9038**
	10%	0.8864	0.8009	0.7989	0.7913	0.8895	0.8691	<u>0.8948</u>	**0.9073**
	20%	0.8907	0.7967	0.7994	0.8004	0.89	0.8785	<u>0.8939</u>	**0.9116**
	30%	0.8926	0.7917	0.7961	0.7983	0.8907	0.8813	<u>0.8997</u>	**0.9134**
Overall Average AUC		0.785	0.708	0.7442	0.7496	0.7955	0.7899	<u>0.8098</u>	**0.8176**
Overall Average Rank		4.04	6.78	5.97	5.4	4.1	4.06	<u>3.01</u>	**2.64**

The boldface and underline in each row signify the performance of the best and second-best models, respectively. † represent the proposed models.

Table 5.4: Performance of proposed as well as existing models on ADNI Data.

Subjects	IFTWSVM [34] (AUC, Seny.) (Spey., Pren.)	IFKRR [194] (AUC, Seny.) (Spey., Pren.)	ELM [22] (AUC, Seny.) (Spey., Pren.)	IFRVFL [32] (AUC, Seny.) (Spey., Pren.)	RVFL [1] (AUC, Seny.) (Spey., Pren.)	GE-IFWRVFL [35] (AUC, Seny.) (Spey., Pren.)	GE-IFRVFL-CIL-1† (AUC, Seny.) (Spey., Pren.)	GE-IFRVFL-CIL-2† (AUC, Seny.) (Spey., Pren.)
CN_vs_AD	(0.8801, 0.9245)	(0.7737, 0.6981)	(0.8444, 0.834)	(0.898, 0.9057)	(0.8903, 0.8491)	(**0.9049**, 0.9057)	(0.8912, 0.9057)	(0.8655, 0.8679)
	(0.8356, 0.8033)	(0.8493, 0.7708)	(0.8548, 0.8083)	(0.8904, 0.8571)	(0.9315, 0.9)	(0.9041, 0.8727)	(0.8767, 0.8421)	(0.863, 0.8214)
CN_vs_MCI	(0.6623, 0.5902)	(0.6329, 0.8361)	(0.6318, 0.5246)	(0.6763, 0.6885)	(0.6295, 0.5246)	(0.681, 0.7213)	(0.6986, 0.6393)	(**0.7122**, 0.7213)
	(0.7344, 0.5143)	(0.4297, 0.4113)	(0.7391, 0.4894)	(0.6641, 0.4941)	(0.7344, 0.4848)	(0.6406, 0.4889)	(0.7578, 0.5571)	(0.7031, 0.5366)
MCI_vs_AD	(0.6212, 0.4923)	(0.5787, 0.3538)	(0.5858, 0.3662)	(**0.6813**, 0.8)	(0.6621, 0.5385)	(0.6805, 0.7538)	(**0.6813**, 0.8)	(0.6681, 0.6308)
	(0.75, 0.5333)	(0.8036, 0.5111)	(0.8054, 0.5263)	(0.5625, 0.5149)	(0.7857, 0.5932)	(0.6071, 0.5269)	(0.5625, 0.5149)	(0.7054, 0.5541)
Average AUC	0.7212	0.6618	0.6873	0.7519	0.7273	<u>0.7555</u>	**0.757**	0.7486

Here, AUC, Seny., Spey. and Pren. denote the area under the curve, sensitivity, specificity and precision, respectively. The boldface and underline in each row signify the performance of the best and second-best models, respectively, in terms of AUC value. † represent the proposed models.

5.4.3 Evaluation on KEEL Datasets with Gaussian Noise

To assess the robustness of the proposed GE-IFRVL-CIL-1 and GE-IFRVFL-CIL-2 models against noise, we contaminate samples with Gaussian noise at levels of 5%, 10%, 20%, and 30% to perturb the features of datasets. These datasets exhibit diverse characteristics in terms of nature and imbalance ratios (IR).

A detailed discussion of the selection of these datasets is expounded as follows: To demonstrate the robustness of the proposed GE-IFRVL-CIL models, we selected a diverse set of 10 datasets characterised by distinct natures and imbalance ratios (IR). Table 5.5 provides comprehensive statistical details pertaining to these datasets. Notably, within this selection, 5 datasets exhibit sample sizes below 500, while the remaining 5 have sample sizes more than 500. To ensure fairness in model evaluations, we chose 5 datasets where neither the proposed GE-IFRVFL-CIL-1 nor the GE-IFRVFL-CIL-2 models attain the highest performance at a noise level of 0%, resulting in a loss for the proposed models (refer to Table 5.6). In addition, we include 5 datasets where either the proposed GE-IFRVFL-CIL-1 or the GE-IFRVFL-CIL-2 models outperform other models at the 0% noise level, indicating a win for the proposed models. To conduct a comprehensive analysis of model performance under varying conditions, we introduce Gaussian noise at levels of 5%, 10%, 20%, and 30% to perturb the features of these datasets.

The detailed experimental results are shown in Table 5.7, and the average AUC of each

Table 5.5: Statistical description of the KEEL datasets.

Dataset	#Samples	#Minority Class Samples (+1)	#Majority Class Samples (-1)	#Features	IR
abalone9-18	731	42	689	7	0.061
aus	690	307	383	14	0.8016
checkerboard_Data	690	307	383	14	0.8016
crossplane150	150	75	75	2	1
ecoli-0-1-4-6_vs_5	280	20	260	6	0.0769
ecoli-0-1_vs_5	240	20	220	6	0.0909
ecoli-0-3-4-6_vs_5	205	20	185	7	0.1081
ecoli-0-6-7_vs_5	220	20	200	6	0.1
ecoli0137vs26	311	54	257	7	0.2101
glass2	214	17	197	9	0.0863
glass5	214	9	205	9	0.0439
haber	306	81	225	3	0.36
heart-stat	270	150	120	13	1.25
led7digit-0-2-4-5-6-7-8-9_vs_1	443	37	406	7	0.0911
monk2	601	206	395	7	0.5215
new-thyroid1	215	35	180	5	0.1944
pima	768	268	500	8	0.536
shuttle-6_vs_2-3	230	10	220	9	0.0455
shuttle-c0-vs-c4	1829	123	1706	9	0.0721
transfusion	748	178	570	4	0.3123
vehicle1	846	217	629	18	0.345
vowel	988	90	898	10	0.1002
wpbc	194	46	148	33	0.3108
yeast-0-3-5-9_vs_7-8	506	50	456	8	0.1096
yeast-0-5-6-7-9_vs_4	528	51	477	8	0.1069
yeast-2_vs_4	2968	858	2110	8	0.4066
yeast3	1484	163	1321	8	0.1234
yeast5	1484	44	1440	8	0.0306

Table 5.6: The description of the datasets used for Gaussian noise experiments.

Dataset	**#Samples**	**#Features**	**Win/Loss**	**IR**
ecoli-0-3-4-6_vs_5	205	7	Loss	0.1081
ecoli-0-6-7_vs_5	220	6	Win	0.1
glass5	214	9	Win	0.0439
haber	306	3	Loss	0.3597
led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	Loss	0.0911
vehicle1	846	18	Loss	0.345
yeast-0-3-5-9_vs_7-8	506	8	Win	0.1096
yeast-0-5-6-7-9_vs_4	528	8	Win	0.1069
yeast-2_vs_4	2968	8	Loss	0.4066
yeast3	1484	8	Win	0.1234

The fourth column indicates that if neither the proposed GE-IFRVFL-CIL-1 nor the GE-IFRVFL-CIL-2 models attain the highest performance at a noise level of 0%, resulting in a loss for the proposed models. In addition, if either the proposed GE-IFRVFL-CIL-1 or the GE-IFRVFL-CIL-2 models outperform other models at the 0% noise level, indicating a win for the proposed models.

Table 5.7: Classification performance of the proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 along with the baseline models w.r.t. different levels of Gaussian Noise.

Dataset	Noise	IFTWSVM [34]	IFKRR [194]	ELM [22]	RVFL [1]	IFRVFL [32]	GE-IFWRVFL [35]	GE-IFRVFL-CIL-1†	GE-IFRVFL-CIL-2†
yeast-0-5-6-7-9_vs_4	0%	0.7137	0.6619	0.6517	0.6625	0.7277	0.7108	0.7067	**0.7775**
	5%	**0.781**	0.6394	0.6741	0.6849	0.7544	0.7367	0.7402	0.7487
	10%	0.7705	0.7102	0.6663	0.7108	0.7382	0.7073	0.762	**0.797**
	20%	0.7411	0.6794	0.6656	0.6625	0.7347	0.6954	0.6983	**0.783**
	30%	0.7425	0.7053	0.5947	0.6695	0.7369	0.6919	0.7067	**0.7676**
ecoli-0-3-4-6_vs_5	0%	0.8083	0.825	0.795	**0.8333**	**0.8333**	0.825	**0.8333**	0.825
	5%	**0.8333**	0.825	0.7883	**0.8333**	0.8	**0.8333**	0.825	0.825
	10%	0.8083	0.8167	0.8217	**0.8333**	0.7917	**0.8333**	0.825	**0.8333**
	20%	0.825	0.7917	0.8283	0.8167	**0.8333**	0.8167	0.8167	**0.8333**
	30%	0.8083	0.8083	**0.83**	0.825	0.8	0.825	0.825	0.825
ecoli-0-6-7_vs_5	0%	0.7336	0.7172	0.8169	0.75	0.7923	0.8087	**0.8675**	0.8347
	5%	0.8087	0.7008	0.7434	0.5342	0.776	0.6093	**0.8169**	0.8087
	10%	0.7923	0.709	0.8085	**0.8333**	0.776	0.7842	0.8005	0.8005
	20%	0.7842	0.709	0.7536	**0.8333**	0.7842	0.8251	0.8087	0.8087
	30%	0.7842	0.7172	0.7801	**0.8333**	0.7842	0.8251	0.8251	0.7418
glass5	0%	0.4922	0.7031	0.7453	0.7188	0.7188	0.7266	0.7813	**0.9766**
	5%	0.9766	0.6875	0.7422	0.7344	**1**	0.9688	0.9531	**1**
	10%	0.7344	0.6875	0.8406	0.9844	**1**	0.9844	0.7578	0.8203
	20%	0.7422	0.6875	0.8406	**1**	0.9766	**1**	**1**	**1**
	30%	0.9609	0.7344	0.9359	**1**	0.9766	0.9922	**1**	0.9688
haber	0%	0.5588	0.5891	0.5822	0.5253	0.6097	**0.6138**	0.6085	0.5938
	5%	0.5991	0.4544	0.5337	0.5938	0.6138	0.5885	**0.6338**	0.6118
	10%	**0.6138**	0.5585	0.5999	0.5453	0.5779	0.5885	**0.6138**	**0.6138**
	20%	0.5865	0.5074	0.5699	0.5526	**0.6285**	0.5885	0.6265	0.6044
	30%	0.6191	0.5076	0.5251	0.5106	0.6138	0.6212	0.6697	**0.675**
led7digit-0-2-4-5-6-7-8-9_vs_1	0%	0.9016	0.5	0.9073	**0.9419**	0.9258	0.9258	0.9339	0.9258
	5%	0.9258	0.8371	0.921	0.9097	**0.9419**	0.9177	0.9177	**0.9419**
	10%	0.9298	0.8371	**0.9419**	**0.9419**	0.9218	0.8976	0.9298	0.9177
	20%	0.9298	0.9137	**0.9419**	**0.9419**	0.9218	0.9056	0.9298	**0.9419**
	30%	0.9298	0.9137	0.8955	0.9379	0.9056	0.9097	0.9218	**0.9419**
vehicle1	0%	0.7859	0.7366	0.7626	0.7683	0.7954	**0.8367**	0.8339	0.7997
	5%	0.7805	0.7298	0.7483	0.7574	0.7996	**0.8215**	0.8161	0.7833
	10%	0.7818	0.7325	0.7341	0.741	0.794	0.7901	**0.8025**	0.8012
	20%	0.7941	0.7079	0.6971	0.6725	0.7598	**0.8148**	0.7997	0.771
	30%	**0.7832**	0.6955	0.675	0.7054	0.7681	**0.7832**	0.7654	0.7819
yeast-2_vs_4	0%	**0.9507**	0.5	0.8172	0.7822	0.915	0.8109	0.8581	0.9326
	5%	0.8546	0.8144	0.7944	0.7394	0.8219	0.743	**0.9361**	0.8682
	10%	0.8431	0.8431	0.7944	0.7002	0.8838	0.7752	**0.9044**	0.8647
	20%	0.8722	0.8466	0.8315	0.8214	0.8838	0.8501	0.836	**0.8934**
	30%	0.9074	0.8325	0.7801	0.8179	0.9014	0.8501	0.9115	**0.9789**
yeast-0-3-5-9_vs_7-8	0%	0.5574	0.5926	0.6126	0.4926	0.5963	0.5444	**0.7**	0.5981
	5%	0.5815	0.5685	0.5696	0.5593	0.5315	0.5741	**0.7111**	0.65
	10%	0.6315	0.5907	0.5448	0.5833	0.5815	**0.7537**	0.65	0.6944
	20%	**0.7204**	0.5815	0.5526	0.5852	0.6111	0.6556	0.687	0.6667
	30%	0.7037	0.5815	0.5715	0.6	0.6796	0.6296	0.6407	**0.7241**
yeast3	0%	0.8175	0.5	0.8124	0.8192	0.9031	0.912	0.9034	**0.9298**
	5%	**0.9375**	0.862	0.7881	0.8101	0.8756	0.8915	0.9097	0.8814
	10%	**0.9326**	0.8607	0.8037	0.8115	0.8886	0.8902	0.9005	0.9093
	20%	0.8866	0.849	0.7935	0.7794	0.8927	**0.909**	0.8994	0.8931
	30%	0.894	0.8386	0.7866	0.7807	0.8961	0.9018	0.89	**0.9124**
Average AUC	0%	0.8896	0.7747	0.8002	0.7862	0.8862	0.8534	0.8949	**0.9064**
	5%	0.8835	0.8022	0.7985	0.7866	0.8833	0.8576	0.8986	**0.9038**
	10%	0.8864	0.8009	0.7989	0.7913	0.8895	0.8691	0.8948	**0.9073**
	20%	0.8907	0.7967	0.7994	0.8004	0.89	0.8785	0.8939	**0.9116**
	30%	0.8926	0.7917	0.7961	0.7983	0.8907	0.8813	0.8997	**0.9134**
Overall Average AUC		0.785	0.708	0.7442	0.7496	0.7955	0.7899	0.8098	**0.8176**
Overall Average Rank		4.04	6.78	5.97	5.4	4.1	4.06	3.01	**2.64**

The boldface in each row signifies the performance of the best model. The underline in the last two rows signifies the second-best model in terms of average AUC and rank. † represents the proposed models.

model *w.r.t.* different level of noise is presented in Table 5.3. Upon scrutinizing Table 5.3, it is evident that the proposed GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 models consistently outperformed the baseline models across various noise levels, as indicated by their superior average AUC. Notably, the GE-IRVFL-CIL-2 and GE-IFRVFL-CIL-1 emerged as the top and second-top performers, respectively, at each noise level. With overall average accuracies of 81.76 and 80.98, the GE-IFRVFL-CIL-2 and GE-IFRVFL-CIL-1 models se-

cured positions among the top two performers. The IFKRR model demonstrated an overall average AUC of 70.8, revealing a substantial approximately 10% performance gap when compared to the proposed GE-IFRVFL-CIL models. This stark contrast demonstrates the superior noise-handling capabilities of the proposed models.

Further, the average ranks affirm the excellence of the GE-IFRVFL-2 and GE-IFRVFL-1 models, standing at 2.64 and 3.01, respectively. These ranks represent the lowest and second lowest among all compared models. This statistical evidence substantiates the robustness of the proposed GE-IFRVFL-CIL models in the presence of noise, consistently surpassing baseline models in a statistically significant manner.

5.4.4 Evaluation on ADNI Dataset

Alzheimer's disease (AD) is a brain dysfunction illness that gradually impairs people's memory and cognitive abilities. In order to train the proposed models GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2, we use scans from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, which is accessible at $adni.loni.usc.edu$. The ADNI project was started in 2003 with the intention of examining neuroimaging techniques such as magnetic resonance imaging (MRI), positron emission tomography (PET), and other diagnostic tests of AD at the stage of mild cognitive impairment (MCI). The feature extraction pipeline is the same as the methodology outlined in [195]. The dataset consists of three cases: control normal (CN) versus moderate cognitive impairment (MCI) (CN_vs_MCI), MCI versus AD (MCI_vs_AD), and CN versus AD (CN_vs_AD).

The AUC values of the proposed GE-IFRVFL-CIL models as well as existing models for AD diagnosis are shown in Table 5.4. We analyze that with an average AUC of 75.70%, the proposed GE-IFRVFL-CIL-1 is the best classifier. The AUC of the remaining models, IFTWSVM, IFKRR, ELM, RVFL, IFRVFL GE-IFWRVFL, and GE-IFRVFL-

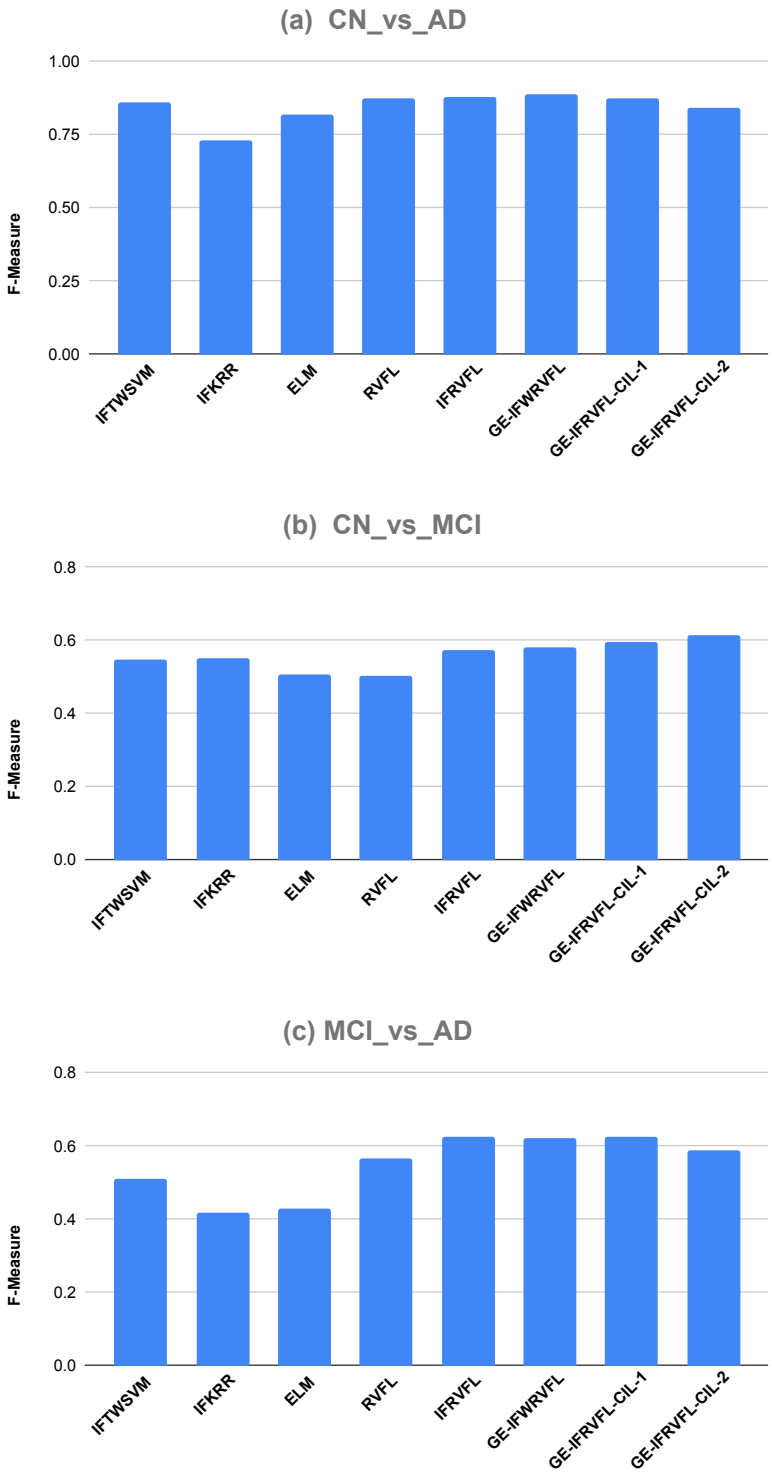


Figure 5.1: Performance comparison of classification models on ADNI data using F-Measure analysis.

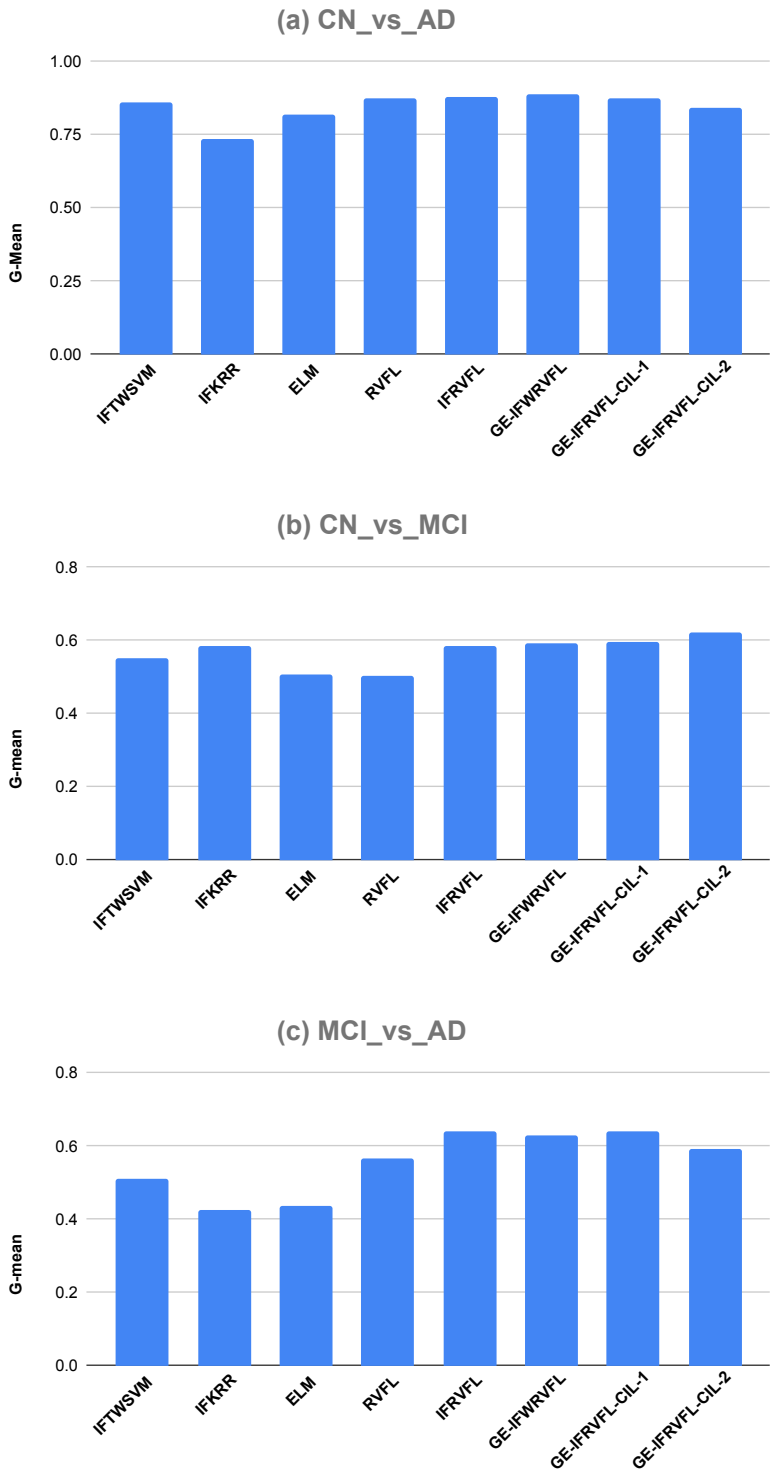


Figure 5.2: Performance comparison of classification models on ADNI data using G-Mean analysis.

CIL-2, are 72.12%, 66.18%, 68.73%, 72.73%, 75.19%, 75.55% and 74.86%, respectively. In comparison to the AUC of IFKRR, the proposed models (GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2) have an average AUC that is around 9.52% and 8.68% higher, respectively. With an AUC of 71.22% in the CN_vs_MCI case, the proposed GE-IFRVFL-CIL-2 came out on top followed by the proposed GE-IFRVFL-CIL-1, which has an AUC of 69.86%. With 68.13% AUC value, the proposed GE-IFRVFL-CIL-1 model and the existing IFRVFL model are the most accurate classifiers for the MCI_vs_AD comparison. Therefore, the proposed GE-IFRVFL-CIL-1 model has an overall winning performance, whereas the proposed GE-IFRVFL-CIL-2 wins in the CN_vs_MCI subject and comes under the top four models in terms of overall performance. Of all models, IFKRR has the lowest average AUC, followed by ELM. The top five models (in terms of average AUC) across all the compared models are from the RVFL family, demonstrating the superior generalization capabilities and dominance of the RVFL-based models.

From Figure 5.1 (a) CN_vs_AD, the F-measure analysis shows that the proposed GE-IFRVFL-CIL-1 model has the competitive performance *w.r.t.* baseline models. From Figure 5.1 (b) CN_vs_MCI, the F-measure of the proposed GE-IFRVFL-CIL-2 is the highest, whereas the F-measure of the proposed GE-IFRVFL-CIL-1 is in the second place. From Figure 5.1 (c) MCI_vs_AD, the F-measure of the proposed GE-IFRVFL-CIL-1 and IFRVFL model secured the first position. Thus, F-measure analysis also attests to the superiority of the proposed models.

Based on Figure 5.2 (a) CN_vs_AD, the F-measure analysis indicates that the performance of the proposed GE-IFRVFL-CIL-1 model is almost on par with that of the baseline models, demonstrating its competitive nature. From Figure 5.2 (b) CN_vs_MCI, we observe that the G-mean of the proposed GE-IFRVFL-CIL-2 and GE-IFRVFL-CIL-1 models has the highest and second highest value, respectively. The proposed GE-IFRVFL-CIL-1 and

IFRVFL models achieved the highest G-mean in the MCI_vs_AD comparison, as shown in Figure 5.2 (c). The statistical tests provide further evidence that the proposed models outperformed all other models in the AD diagnosis and achieved the highest rankings in both G-mean and F-measure analyses.

5.5 Summary

This chapter addressed the class imbalance problem in RVFL models by proposing two models, GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2, which integrate graph embedding, intuitionistic fuzzy theory, and a class weighting mechanism. These components enable the proposed models to effectively handle class imbalance, noisy samples, and outliers while preserving the intrinsic geometric structure of the data.

The performance of the proposed models was evaluated on 28 KEEL imbalanced benchmark datasets. The results based on average AUC and statistical tests demonstrate that GE-IFRVFL-CIL-1 and GE-IFRVFL-CIL-2 outperform the baseline classifiers. Under Gaussian noise contamination, GE-IFRVFL-CIL-2 and GE-IFRVFL-CIL-1 achieve the best and second-best performances, respectively, indicating their robustness.

To further demonstrate their applicability, the proposed models were applied to the ADNI dataset for Alzheimer's disease diagnosis. GE-IFRVFL-CIL-1 achieves the best performance in the challenging MCI vs AD classification task, while GE-IFRVFL-CIL-2 performs best for the CN vs MCI classification task.

Chapter 6

RVFL-X: A Novel Complex Randomized Network and Methods for Complex Transformation of Real-Valued Tabular Datasets

Outcome of this Chapter

Publication: **M. Sajid**, A. Quadir, M. Akhtar and M. Tanveer (2025). "RVFL-X: A Novel Randomized Network Based on Complex Transformed Real-Valued Tabular Datasets," in ***International Joint Conference on Neural Networks (IJCNN)***, pp. 1-8, **DOI:** https://doi.org/10.1109/IJCNN64981.2025.11227691 [**Core Rank B Conference**]

Source Code: https://github.com/MdSajid1044/RVFL-X

The previous chapters have successfully addressed several critical challenges, including interpretability, class imbalance, robustness, computational efficiency, and scalability. However, recent advancements in neural networks, supported by foundational theoretical insights, highlight the superior representational capacity of complex-valued models. Despite these advantages, their integration into RdNNs remains limited, primarily due to the absence of effective mechanisms for transforming real-valued tabular data into complex-valued representations. To overcome this limitation, we first propose two novel mechanisms for generating complex-valued representations from real-valued datasets: a natural transformation approach and an autoencoder-driven method. Building upon these representations, we introduce RVFL-X, a complex-valued extension of the RVFL network. RVFL-X seamlessly incorporates complex transformations into real-valued data while preserving the simplicity and computational efficiency of the original RVFL architecture. By utilizing complex-valued inputs, weights, and activation functions, the proposed model effectively processes complex representations while producing real-valued outputs. Extensive experiments conducted on 80 real-valued UCI datasets demonstrate that RVFL-X consistently outperforms the standard RVFL as well as state-of-the-art (SOTA) RdNN variants, thereby highlighting its enhanced robustness and expressive power across diverse application domains.

6.1 Background

The performance of RdNN-based models is significantly influenced by the generation of hidden layer features, which play a critical role in the propagation of information to the output layers. In the case of the RVFL network, hidden layer features are generated through random projection. However, this random generation often fails to capture some of the nonlinear features inherent in the data, potentially limiting the model's expressiveness.

Prior studies have demonstrated that augmenting RVFL variants with feature enhancement techniques [28], [158], [196] substantially improves their performance by capturing intricate patterns and relationships in the data. Despite these advancements, the exploration of feature enhancement remains relatively nascent in the context of RVFL-based models.

Recent studies in neural networks, coupled with foundational theoretical analyses, suggest that complex numbers offer a richer representational capacity [47], [48], [197]. They enable robust memory retrieval mechanisms while exhibiting noise-resistant properties. These benefits, however, have been underutilized in randomized neural networks, primarily due to the lack of mechanisms to transform real tabular datasets into complex-valued representations. To address this gap, we propose the RVFL-X model: a novel extension of RVFL that integrates complex transformations into real-valued datasets while maintaining the simplicity and efficiency of the original architecture.

The use of complex parameters introduces several advantages across computational, biological, and signal processing domains. From a computational perspective, Danihelka et al. [47] demonstrated the efficiency and numerical stability of holographic reduced representations [49], which employ complex numbers for associative memory retrieval. Similarly, biologically inspired architectures have leveraged complex weights to enhance the versatility of representations. For example, Reichert and Serre [50] introduced a biologically plausible deep network using complex-valued neurons to model richer and more adaptable representations. These studies emphasize the potential of complex formulations to encode the firing rate and relative timing of neurons.

Despite these theoretical and practical advantages, most existing applications of complex-valued neural networks are restricted to datasets inherently expressed in complex forms, such as those arising from signal processing tasks. The RVFL-X model addresses this limitation by proposing a novel mechanism to transform real tabular datasets into a

complex-valued representation. This transformation allows the RVFL-X to retain the architectural simplicity of its real-valued counterpart while exploiting the rich representational capacity of complex-valued features.

To the best of our knowledge, this is the first attempt to systematically integrate complex-valued transformations into real-valued RVFL architectures. The RVFL-X maintains the core properties of the original RVFL model with minimal architectural modifications, enabling the effective capture of nuanced patterns and relationships in real-world datasets. This novel approach paves the way for exploring new avenues in RdNNs and unlocking their full potential for a wide range of applications.

The primary highlights of this chapter are as follows:

[1] A novel mechanism is introduced to transform real tabular datasets into complex-valued representations.

[2] Two transformation techniques are proposed: an intuitive and natural approach; and an autoencoder-based method.

[3] The complex variant of RVFL (RVFL-X) is developed, which converts real-valued tabular data into complex values, employs complex weights and complex activation functions and generates real-valued outputs.

[4] Comprehensive comparisons of RVFL-X with 10 SOTA models are conducted across diverse datasets, including binary and multiclass classification, as well as small and large-scale data.

6.2 Proposed Work

Complex-valued neural networks offer significant advantages, however their application has primarily been limited to datasets inherently expressed in complex forms. There is a gap in extending these transformations to real-valued tabular data, limiting the broader potential of neural networks in various domains. This creates an opportunity to develop a more versatile approach that leverages complex-valued representations. We propose RVFL-X, a complex-valued variant of RVFL that transforms real-valued tabular datasets into complex representations. The proposed model preserves the simplicity of the original RVFL while utilizing complex weights and activation functions to capture more intricate patterns and relationships in the data. We first fix the notations for this chapter and then discuss the proposed methods.

6.2.1 Notations

Consider a training dataset represented as $\mathcal{Z} = \{(u_j, t_j) \mid j \in \{1, 2, \ldots, M\}\}$, where $u_j \in \mathbb{R}^{1\times L}$ corresponds to the input features, and $t_j \in \mathbb{R}^{1\times C}$ is the associated target vector for C classes. Here, M denotes the total number of training samples, and L indicates the number of attributes. The transpose of a matrix is denoted by $(\cdot)^T$. The matrices representing the inputs and outputs are given by $U = [u_1^T, u_2^T, \ldots, u_k^T]^T$and $T = [t_1^T, t_2^T, \ldots, t_k^T]^T$, respectively.

6.2.2 Conversion of Real-valued Data to Complex-valued Data

This section provides a detailed explanation of the proposed process used to convert real-valued data into complex-valued data. In the subsequent subsection, we apply this methodology to define our proposed RVFL-X model using the complex-converted data. To transform the real-valued tabular dataset into a complex-valued one, we propose two distinct

methods: first, an intuitive and natural approach (Section 6.2.2.1), and second, based on an autoencoder-based method (Section 6.2.2.2).

6.2.2.1 Natural Method

For the numerical dataset U, we define the complex dataset U^X as follows:

$$U^X = U + iS, \tag{6.1}$$

where $S = \mathbf{0}$ is the zero matrix having the same dimension as of U. We call it the natural method because, in the Field of complex numbers, every real number is a complex number, and the natural way to write a real number u into a complex number is $u + i0$, where 0 is scalar zero.

6.2.2.2 Autoencoder-Based Method

In this method, the complex dataset U^X is constructed from a numerical dataset U as:

$$U^X = U + iS, \tag{6.2}$$

where S is a matrix of the same dimensions as U, obtained using an autoencoder applied to U. The autoencoder, consisting of an encoder and a decoder, transforms U into a latent representation S, preserving key structural properties. This latent representation serves as the imaginary part of U^X, enriching the dataset with additional information for improved modeling. This approach leverages the autoencoder's capacity to extract and preserve essential features, ensuring the complex dataset U^X is both informative and structurally robust.

The encoder maps the input matrix U to the latent representation $\hat{S}$ through random weights:

$$\hat{S} = \xi(UW), \tag{6.3}$$

where $W \in \mathbb{R}^{L\times L}$ is the weight matrix initialized randomly (e.g., drawn from uniform distributions), and ξ is the normalization function. The decoder reconstructs U from S using

$$\hat{U} = \xi(SV), \tag{6.4}$$

where $V \in \mathbb{R}^{L\times L}$ is the decoder weight matrix. This normalization ensures consistency in scaling during both encoding and decoding while preserving the data's structural properties.

To learn the latent representation S, the reconstruction error between U and $\hat{U}$ is minimized through an ℓ_2-regularized optimization problem, preventing overfitting. The optimization problem is formulated as:

$$V^* = \arg\min_V \frac{\mathcal{C}}{2}\|U - \hat{S}V\|^2 + \frac{1}{2}\|V\|^2, \tag{6.5}$$

where $\mathcal{C}$ is a regularization parameter. The closed-form solution for V^* is given by:

$$V^* = \begin{cases} \hat{S}^T\left(\hat{S}\hat{S}^T + \frac{1}{\mathcal{C}}I\right)^{-1}U, & \text{if } M < L, \\ \left(\hat{S}^T\hat{S} + \frac{1}{\mathcal{C}}I\right)^{-1}\hat{S}^TU, & \text{if } M \geq L, \end{cases} \tag{6.6}$$

where I represents the identity matrix.

Once V^* is computed, the latent representation S is obtained from the encoder as:

$$S = \xi\left(U\left(\varpi V^* + (1-\varpi)V^{*T}\right)\right), \tag{6.7}$$

where ϖ takes only two values, i.e., 0 or 1, thus $\varpi \in \{0, 1\}$. ϖ is tuned during training. If $\varpi = 0$, V^* does not contribute to the formation of the imaginary part S, whereas if $\varpi = 1$, V^{*T} is excluded. This latent representation effectively captures critical features of the input data in a transformed space, enhancing robustness and generalization through regularization. Consequently, this representation defines the imaginary part of the data.

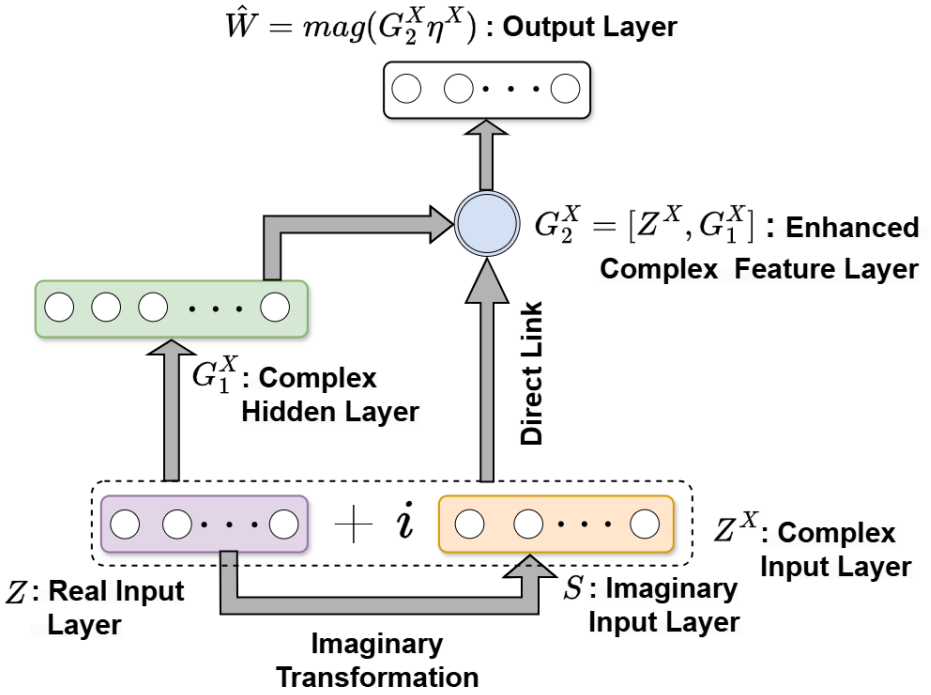


Figure 6.1: Architecture of the proposed RVFL-X network.

6.2.3 RVFL-X

The architecture of the proposed RVFL-X network is depicted in Figure 6.1. For the given input matrix U, we convert it to the complex version U^X and the method of calculating it is given in Section 6.2.2. Let the complex weight matrix, connecting the input matrix U^X to the hidden layer and complex bias matrix of the hidden layer be denoted as F_w^X and F_b^X, respectively, and defined as follows:

$$F_w^X = F_w^{real} + iF_w^{imag} \text{ and } F_b^X = F_b^{real} + iF_b^{imag}, \tag{6.8}$$

where $F_w^{real} \in \mathbb{R}^{L \times N_h}$ and $F_w^{imag} \in \mathbb{R}^{L \times N_h}$ are generated randomly from the uniform distribution of $[-1, 1]$ space and $F_b^{real} \in \mathbb{R}^{L \times N_h}$ and $F_b^{imag} \in \mathbb{R}^{L \times N_h}$ are generated randomly from the uniform distribution of $[0, 1]$. All the columns of the F_b^{real} are identical, and similarly for F_b^{imag}.

To regularize the model from overfitting, we make some of the entries of the F_w^X and F_b^X to be zero and denoted as ${}_\alpha\hat{F}_w^X$ and ${}_\alpha\hat{F}_b^X$, respectively. The number of entries set to be zero is regulated the hyperparameter α, where it denotes the percentage of the entries set to be zero which can be tuned during the model learning process.

The complex hidden layer is generated as follows:

$$G_1^X = \sigma^X(U^X \cdot_X {}_\alpha\hat{F}_w^X +_X {}_\alpha\hat{F}_b^X) \in \mathbb{R}^{M \times N_h}, \tag{6.9}$$

where $\cdot_X$ and $+_X$ are complex multiplication and addition, respectively. σ^X is modified complex activation function for the real activation function σ and defined as follows:

$$\sigma^X(z_x) = \sigma(Real(z_x)) + i\sigma(Imag(z_x)), \tag{6.10}$$

where $Real(z_x)$ and $Imag(z_x)$ are the real and imaginary part of the complex number z_x, respectively.

Next, we form the enhanced feature matrix as follows:

$$G_2^X = [U^X, G_1^X] \tag{6.11}$$

the predicted output $\hat{T^X}$ of the proposed model can be calculated as:

$$\hat{T} = mag(G_2^X \eta^X), \tag{6.12}$$

where η^X is the complex weight matrix connecting the enhanced feature layer (concatenation of input and hidden layers) to the output layer. $mag(z_x)$ is the magnitude of the complex number z_x and defined as $\sqrt{Real(z_x)^2 + Imag(z_x)^2}$.

The proposed optimization problem can be formulated as:

$$\eta^X_{\min} = \arg\min_{\eta^X} \frac{\mathcal{C}}{2}\|G_2^X \eta^X - T\|_F^2 + \frac{1}{2}\|\eta^X\|_F^2, \tag{6.13}$$

where $\mathcal{C} > 0$ is a regularization parameter and same as used in (6.5). After solving the optimization problem (6.13) for η^X, we get:

$$\eta^X_{\min} = \begin{cases} {G_2^X}^T \left(G_2^X {G_2^X}^T + \frac{1}{\mathcal{C}} I \right)^{-1} T, & \text{if } M < (L + N_h), \\ \left({G_2^X}^T G_2^X + \frac{1}{\mathcal{C}} I \right)^{-1} {G_2^X}^T T, & \text{if } M \geq (L + N_h). \end{cases} \tag{6.14}$$

As described in Section 6.2.2, we proposed two methods for generating the imaginary part of the data. Accordingly, the RVFL-X model based on the method in Section 6.2.2.1 will be referred to as RVFL-X-N, while the RVFL-X model based on the method in Section 6.2.2.2 will be denoted as RVFL-X-Auto henceforth.

6.3 Discussion on the Proposed Complex-valued Representation

Complex-valued neural networks have demonstrated remarkable success in applications where the input data are naturally represented in the complex domain, including radar signal processing, wireless communications, magnetic resonance imaging, electroencephalography (EEG), and synthetic aperture radar imaging. However, the vast majority of real-world machine learning problems involve real-valued tabular datasets, for which a natural complex-valued representation does not exist. Consequently, an important question arises: *can real-valued tabular data benefit from complex-valued learning, and if so, how should such a transformation be constructed?* This section discusses the motivation behind the proposed RVFL-X framework, compares it with existing approaches for generating complex-

valued representations, clarifies why the proposed framework differs fundamentally from simply increasing the model width, and explains the rationale for employing an autoencoder to construct the imaginary component.

6.3.1 Motivation for Complex-valued Representation of Real-valued Tabular Data

The primary objective of the proposed RVFL-X framework is to extend the advantages of complex-valued neural networks beyond naturally complex-valued signals to general real-valued tabular datasets. Unlike conventional real-valued neural networks, complex-valued neural networks process information in the complex field, where each feature consists of coupled real and imaginary components. Such representations enable complex-valued arithmetic, complex-valued weight matrices, and complex activation functions to jointly model interactions between these two components, thereby providing a richer representational space than purely real-valued learning.

It is important to emphasize that the proposed transformation does not replace the original data representation. Instead, the original real-valued attributes are preserved as the real component of the complex representation, while an additional complementary representation is introduced as the imaginary component. Consequently, no information contained in the original dataset is discarded. Rather, the complex-valued representation enriches the feature space by augmenting each feature with additional latent information that can be jointly exploited by the complex-valued RVFL architecture. This richer representation enables the network to capture more intricate nonlinear relationships among the attributes while preserving the computational simplicity of the original RVFL framework.

6.3.2 Difference Between Complex-valued Representation and Increasing Model Width

A natural question is whether the improvement obtained by RVFL-X merely results from increasing the capacity of the network, similar to employing a wider real-valued RVFL with additional hidden neurons. Although both approaches increase the expressive capability of the model, they do so through fundamentally different mechanisms.

Increasing the width of a real-valued neural network enlarges the number of hidden neurons, thereby increasing the number of learned nonlinear feature mappings while maintaining the same real-valued representation space. In contrast, the proposed RVFL-X preserves the original network architecture and hidden-layer structure but transforms the representation space itself from the real field to the complex field. Consequently, the improvement does not arise solely from increasing the number of hidden neurons, but from enabling each feature to simultaneously encode complementary real and imaginary information that interacts through complex-valued arithmetic.

Furthermore, complex-valued multiplication inherently couples the real and imaginary components during feature transformation, producing interactions that cannot generally be reproduced by merely adding more real-valued neurons of identical architecture. Therefore, the proposed complex-valued representation increases the expressive capability of the network through richer feature interactions rather than solely increasing the network width. Since the experiments in this chapter employ the same RVFL architecture and comparable hidden-layer configurations, the observed performance improvements can be attributed primarily to the enhanced representational capability introduced by complex-valued learning instead of simply increasing model capacity.

6.3.3 Comparison with Existing Complex-valued Transformation Methods

Several approaches have been proposed in the literature to enable complex-valued neural networks to process real-valued data. The simplest strategy constructs complex-valued inputs by assigning zero values to the imaginary component, thereby representing each real-valued feature as $u + i0$. Although this approach allows existing complex-valued neural networks to process real-valued datasets without modification, it does not introduce any additional information beyond the original data representation.

Another category of approaches derives the imaginary component using predefined mathematical transformations such as the Hilbert transform, Fourier transform, analytic signal representation, random complex projections, or other signal-processing techniques. These methods have achieved considerable success in applications involving temporal, frequency-domain, or naturally oscillatory signals because they preserve meaningful phase information. However, such transformations are generally less appropriate for heterogeneous tabular datasets whose attributes often possess different semantic meanings and do not exhibit inherent temporal or spectral structures.

In contrast, the proposed RVFL-X introduces two transformation strategies specifically designed for real-valued tabular data. The first strategy, referred to as the natural method, serves as a direct baseline by assigning zero values to the imaginary component, thereby preserving the conventional interpretation of complex numbers without modifying the original information. The second strategy employs an autoencoder to generate the imaginary component in a data-driven manner. Unlike predefined mathematical transformations, the proposed approach adapts the complex-valued representation to the statistical characteristics of the dataset itself, making it applicable to general tabular learning problems without requiring domain-specific assumptions regarding signal structure or frequency content.

6.3.4 Rationale for Employing an Autoencoder to Generate the Imaginary Component

The principal motivation for employing an autoencoder is to construct an informative imaginary component that captures complementary structural information rather than merely duplicating the original features. Autoencoders are specifically designed to learn compact latent representations by reconstructing the input data while preserving their most informative characteristics. Consequently, the learned latent representation naturally captures nonlinear dependencies and correlations among the original attributes.

Within the proposed RVFL-X framework, the original observations constitute the real component of the complex-valued representation, whereas the latent representation learned by the autoencoder forms the imaginary component. As a result, each complex-valued feature simultaneously contains the original measurements together with an additional representation describing the underlying structural relationships among the attributes. This complementary representation enables the subsequent complex-valued RVFL to jointly exploit both observed information and latent feature interactions during learning.

Compared with manually designed transformations, the proposed autoencoder-based strategy offers several advantages. First, it is entirely data-driven and therefore automatically adapts to datasets having different statistical characteristics. Second, because the latent representation is learned through reconstruction, the generated imaginary component preserves essential structural information while suppressing redundant variations. Third, the optimization incorporates ℓ_2 regularization, which improves the robustness and generalization capability of the learned latent representation. Consequently, the proposed autoencoder-based transformation is expected to produce a more informative complex-valued representation than fixed mathematical transformations that do not adapt to the underlying data distribution.

6.4 Experiments, Results and Discussion

This section presents comprehensive details of the experimental setup, datasets, and compared models. Subsequently, we delve into the experimental results and conduct statistical analyses. At last, we conduct sensitivity analyses for various hyperparameters of the proposed RVFL-X and ablation study.

6.4.1 Compared Models, Datasets and Experimental Setup

Compared Models: We conduct comparisons among the proposed RVFL-X; and several benchmarks, including RVFL [1], RVFLwoDL (RVFL without direct link, also known as ELM) [22], intuitionistic fuzzy RVFL (IFRVFL) [32], graph embedded ELM with linear discriminant analysis (GEELM-LDA) [173], graph embedded ELM with local Fisher discriminant analysis (GEELM-LFDA) [173], minimum class variance based ELM (MCVELM) [175] and Neuro-fuzzy RVFL-R (NF-RVFL-R), Neuro-fuzzy RVFL-K (NF-RVFL-K), and Neuro-fuzzy RVFL-C (NF-RVFL-C) whose fuzzy layer centers are generated using random-means, K-means and fuzzy C-means [17].

Datasets: To assess the effectiveness of the proposed RVFL-X models, we utilize 80 benchmark datasets from the UCI repository [7], spanning various domains and sizes. These datasets are categorized into four groups: C-1 category: binary and small, C-2 category: binary and large, C-3 category: multiclass and small, and C-4 category: multiclass and large. Small datasets have sample sizes ranging from 100 to 1,000, while large datasets range from 1,001 to 100,000.

Experimental Setup: The experimental hardware setup includes a personal computer equipped with an Intel(R) Xeon(R) Gold 6226R CPU, operating at a clock speed of 2.90 GHz, and 128 GB of RAM. The system runs Windows 11 and utilizes Matlab 2023a to execute all experiments. Following the experimental protocol in [17], we determine the

Table 6.1: Accuracy and rank across 21 C-1 category datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	IFRVFL [32]	GEELM-LDA [173]	GEELM-LFDA [173]	Total-var-RVFL [174]	MCVELM [175]	NF-RVFL-R [17]	NF-RVFL-K [17]	NF-RVFL-C [17]	RVFL-X-N†	RVFL-X-Auto†
acute_inflammation	100	100	100	100	100	100	100	100	100	100	100	100
acute_nephritis	100	100	100	95.8333	95.8333	100	100	100	100	100	100	100
breast_cancer_wisc	88.5653	87.9897	89.8499	86.5591	84.2724	88.704	88.8448	88.2785	88.4183	87.705	90.2785	89.9897
chess_krvkp	72.0313	71.936	72.6262	76.7814	69.2351	73.7833	73.7828	82.6045	81.7908	85.2322	83.0741	84.5757
congressional_voting	63.6782	63.2184	58.8506	59.7701	54.2529	63.908	63.4483	63.908	63.908	64.1379	63.6782	64.8276
credit_approval	85.2174	85.3623	86.5217	85.5072	85.6522	85.5072	85.5072	85.942	85.942	85.7971	85.5072	86.087
cylinder_bands	66.4154	65.8081	63.4875	67.5405	67.5385	67.7727	66.0156	68.9606	69.724	70.1294	70.1256	70.3103
echocardiogram	84.6724	83.9031	80.7692	73.2194	68.661	85.4701	85.4986	85.4416	85.4416	85.4416	85.4416	86.2108
haberman_survival	73.4902	73.8181	75.1348	55.5632	52.2898	74.4738	73.4902	75.1296	75.1296	75.4574	74.146	75.4574
ilpd_indian_liver	71.5311	72.2149	72.7277	61.7521	61.4353	72.5626	72.5508	72.3932	72.3917	72.2178	72.3814	73.4188
mammographic	79.9196	79.2978	79.7107	75.7572	73.7932	80.0237	80.2332	79.2957	79.504	79.1899	80.2321	79.1942
monks_1	83.7934	82.1654	84.6895	73.5811	71.5476	84.1538	84.8697	81.4527	82.3584	84.342	93.5151	90.8156
monks_2	80.0138	81.6804	82.6722	60.7273	57.23	85.8485	82.8499	85.8485	87.3774	87.3774	89.0041	87.5028
monks_3	91.1564	90.7912	90.0704	82.8419	74.5602	91.6986	91.8821	91.8436	91.6486	92.1744	92.9615	93.1433
pima	72.0075	72.2698	73.8282	66.401	54.0625	72.92	72.9166	73.6983	72.9174	72.6594	73.824	74.3545
pittsburg_bridges_T_OR_D	87.1905	87.1905	89.1905	82.1429	80.1905	89.1429	90.1429	90.1905	88.2381	90.1905	90.1429	90.2381
planning	71.3814	71.3814	69.8048	59.1592	61.1261	72.4925	73.048	73.018	71.9369	71.9369	72.4925	73.033
spect	68.3019	66.7925	68.3019	63.3962	62.6415	69.0566	67.9245	69.0566	69.434	68.6792	68.3019	70.1887
statlog_heart	80.3704	80	81.8519	73.7037	74.8148	81.4815	82.5926	81.4815	82.2222	81.8519	81.8519	82.2222
tic_tac_toe	88.8264	88.9278	81.4316	94.6875	79.2059	91.8521	90.0769	65.3125	68.5684	79.3216	94.5681	84.6564
vertebral_column_2clases	71.2903	70.6452	85.4839	94.8387	92.2581	72.5806	74.5161	79.0323	81.6129	80.3226	73.5484	83.871
Average Accuracy	79.993	79.7806	80.3335	75.703	72.4096	81.1158	80.9615	80.6137	80.884	81.6269	82.6226	**82.8618**
Average Rank	8.2381	8.9048	6.2381	9.3571	10.5714	5.5	5.5	5.6905	5.5714	5.2619	4.5	**2.6667**

The bold values in each row indicate the best-performing model w.r.t. to the row metric, while the underlined values highlight the second-best-performing model. † denotes the proposed models.

Table 6.2: Accuracy, rank and standard deviation (Std. Dev.) across 14 C-2 category datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	IFRVFL [32]	GEELM-LDA [173]	GEELM-LFDA [173]	Total-var-RVFL [174]	MCVELM [175]	NF-RVFL-R [17]	NF-RVFL-K [17]	NF-RVFL-C [17]	RVFL-X-N†	RVFL-X-Auto†
adult	84.0342	83.8725	83.5438	83.422	83.1784	84.0178	83.9011	83.2992	83.5879	84.6218	84.6976	84.6259
bank	89.4051	89.4934	89.1173	89.6042	46.4966	89.6705	89.6703	89.4269	89.4492	89.9139	89.8475	89.8032
breast_cancer_wisc_diag	93.8503	92.2683	89.2719	93.1455	86.988	94.1997	93.4995	94.3751	94.3782	95.6063	95.4324	95.08
breast_cancer_wisc_prog	81.359	80.3846	78.359	62.0897	62.8462	82.359	81.8718	83.359	81.8846	80.9103	81.8718	82.8462
conn_bench_sonar_mines_rocks	62.079	60.5226	54.8316	80.5343	73.6585	64.4251	64.4251	65.8885	66.2602	63.4611	63.4959	74.007
connect_4	75.4518	75.4059	75.3407	76.7316	75.4281	75.4992	75.4459	75.4844	75.5022	76.2156	77.2266	77.0017
hill_valley	82.2603	79.7014	80.2772	76.9231	80.3846	83.6592	80.0289	82.8429	82.6732	83.0874	83.0028	85.9742
ionosphere	88.6358	86.9175	84.3581	83.2274	82.664	89.4809	88.326	88.3461	89.7626	88.9095	91.7505	90.8974
magic	78.7487	78.5279	77.5289	77.4289	78.3579	78.7171	78.7592	76.3407	76.3302	78.4017	79.3218	82.224
mushroom	96.3931	96.0735	93.9684	96.3071	98.7561	97.2919	96.7382	92.6761	95.9132	99.3351	97.7596	97.6118
oocytes_merluccius_nucleus_4d	82.2884	81.5045	79.1554	81.9957	82.7273	82.3883	82.1942	81.6021	81.7006	82.1923	83.5643	83.4605
oocytes_trisopterus_nucleus_2f	78.9419	77.5182	75.2219	78.0694	79.8889	79.6013	79.6025	79.3863	79.3779	79.9345	80.8119	82.4566
ringnorm	51.5541	51.5405	51.473	51.5541	51.5784	52.0405	51.9459	51.6081	51.6486	51.6892	51.7973	52.0135
spambase	88.546	87.0472	85.0883	88.9582	88.546	88.6116	87.4382	88.9582	88.9583	89.3931	89.2423	89.3075
Average Accuracy	80.9677	80.0556	78.3954	79.9994	76.5356	81.5687	80.9891	80.971	81.2448	81.6908	82.1302	**83.3793**
Average Rank	7.2857	9.5714	10.8571	8.2143	8.1071	4.6071	6.5	7.3214	6.5	4.1429	2.8214	**2.0714**
Average Std. Dvn.	4.4417	4.7219	5.2407	5.3081	10.0387	4.7577	4.4265	4.8611	4.6659	4.9101	**4.3711**	4.4173

The bold values in each row indicate the best-performing model w.r.t. to the row metric, while the underlined values highlight the second-best-performing model. † denotes the proposed models.

optimal hyperparameters and testing accuracy using grid search along with a five-fold cross-validation technique. Additionally, the hyperparameters for all baseline models are tuned according to the approach outlined in [17]. For the baseline models, we tune six activation functions: (1) Sigmoid, (2) Sine, (3) Tribas, (4) Radbas, (5) Tansig, and (6) ReLU, as specified in [17]. For the proposed RVFL-X model, real-valued activation functions are transformed into complex activation functions using the method described in Section 6.2.3. The regularization parameter $\mathcal{C}$ is chosen from the range $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. The number of hidden nodes (N) is selected from the range of 3 to 203, with a step size of 20. For the proposed model, the percentage parameter (α) is tuned from the set $\{0, 0.1, \ldots, 0.5\}$, where $\alpha = t$ represents the $t \times 100$ percentage that sets the parameters to zero, as discussed in Section 6.2.3.

6.4.2 Evaluation on C-1 Category Datasets

Table 6.1 summarizes the experimental results, showcasing the accuracy for each dataset, average accuracy and rank of the proposed RVFL-X models compared to baseline models across 21 C-1 category (binary and small) datasets. Among them, the proposed RVFL-X-Auto achieves the highest average accuracy of 82.8618%, followed by the proposed RVFL-X-N at 82.6226%, outperforming other SOTA models.

In the ranking approach, models are ranked according to their performance on each dataset, where lower ranks correspond to better-performing models. The average rank of each model is then computed across all datasets. The last row of Table 6.1 reports these average ranks. The proposed RVFL-X-Auto and RVFL-X-N models obtain average ranks of 2.6667 and 4.5, respectively, indicating better performance compared with the baseline models.

The Friedman test [154] is employed to determine whether statistically significant differences exist among the models based on their average ranks. In this study, twelve models were evaluated across twenty-one datasets. The Friedman statistic is ($\chi_F^2 = 92.653$). Using the improved test proposed by Iman and Davenport [155], the corresponding statistic is ($F_F = 13.3943$). According to the F-distribution table, the critical value ($F_F(11, 220)$) at the (5%) significance level is (1.8324). Since the obtained value (13.3943) is greater than the critical value (1.8324), the null hypothesis is rejected, confirming that significant differences exist among the evaluated models.

Furthermore, the Nemenyi post hoc test [153] is used to analyze pairwise statistical differences between models. Two models are considered significantly different if the difference between their average ranks exceeds the critical difference ($C.D.$). At the significance level ($\alpha = 0.05$), the computed $C.D.$ value is (3.6363). Based on the ranks reported in Table 6.1, both proposed RVFL-X models are statistically superior to RVFL, RVFLwoDL,

GEELM-LDA, and GEELM-LFDA. However, the test does not establish statistically significant differences between the proposed models and several other baseline methods. Nevertheless, the lowest average ranks and the highest accuracy achieved by the proposed models compared with existing state-of-the-art models demonstrate their superior generalization capability.

6.4.3 Evaluation on C-2 Category Datasets

Table 6.2 presents the experimental results, including the accuracy for each dataset, the average accuracy, the standard deviation, and the rank of the proposed RVFL-X models compared to baseline models across 14 C-2 category (binary and large) datasets.

From the results, the RVFL-X-Auto model demonstrates the highest accuracy (83.3793%), the lowest standard deviation (2.0714), and the second-lowest rank (4.4173). This highlights the model's ability to effectively transform real-valued datasets and process them through the complex architecture of the RVFL network. Notably, the RVFL-X-Auto and RVFL-X-N models achieve approximately 3% higher accuracy than the RVFL and RVFLwoDL networks, further confirming the superior generalization performance of the proposed RVFL-X models. Similarly, the RVFL-X-N model emerges as the second-best in accuracy while achieving the best rank, demonstrating its competitive performance. The low standard deviations of both models reflect the high certainty and stability of their predictions.

Applying the Friedman test yields $F_F = 15.3871$. Referring to the F-distribution table, with $F_F(11, 143) = 1.8562$ at a 5% significance level, we reject the null hypothesis, confirming significant differences among the models. Furthermore, the Nemenyi post hoc test calculates a critical difference ($C.D.$) of 4.4535, indicating that both proposed RVFL-X models are statistically superior to RVFL, RVFLwoDL, IF-RVFL, GEELM-LDA, GEELM-

Table 6.3: Average accuracy and rank across 23 C-3 category datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	Total-var-RVFL [174]	MCVELM [175]	NF-RVFL-R [17]	NF-RVFL-K [17]	NF-RVFL-C [17]	RVFL-X-N†	RVFL-X-Auto†
annealing	89.6381	88.1918	89.1937	88.6381	86.6313	87.0782	90.8591	91.0832	91.4165
audiology_std	69.3205	65.7564	50.3718	51.8846	71.3846	70.3718	67.8077	71.359	70.3462
balance_scale	98.4	98.4	98.56	98.56	98.4	98.24	98.24	98.72	98.56
car	72.6265	72.1057	72.6234	72.2758	70.0139	70.8337	71.0626	73.2048	72.1642
contrac	40.458	40.1183	41.3386	41.0688	41.4934	42.3673	41.5416	41.6089	45.6813
dermatology	97.5379	97.0011	97.2714	97.5417	97.5454	97.8193	97.5379	97.8119	97.5417
ecoli	60.9175	61.2116	51.0667	51.0667	60.619	60.6277	61.5145	60.9175	62.7085
energy_y1	88.6716	88.2777	89.7106	89.0604	87.889	88.5434	89.0595	91.7927	90.7555
energy_y2	90.4932	90.4932	91.14	90.7538	89.3269	89.1928	89.8404	91.9243	92.0567
glass	37.6855	39.0808	34.4297	37.6855	42.3477	42.8239	40.4873	39.546	44.1971
hayes_roth	61.875	60.625	62.5	61.875	61.875	61.875	65	65	65
heart_cleveland	59.7268	59.377	61.0328	59.3607	61.0383	61.3388	60.0437	61.0383	61.6721
heart_va	40	40	40.5	40	41	42.5	41.5	41	41.5
iris	74.6667	74	74.6667	77.3333	73.3333	74.6667	74.6667	79.3333	85.3333
led_display	72.6	72.5	73.5	72.8	72.7	72.2	73.4	73.6	73.5
lymphography	86.4138	85.0345	86.4138	86.4368	85.7701	87.1034	88.5057	88.4368	87.7931
nursery	70.3935	70.1775	63.0093	62.8627	66.7824	66.3889	71.9444	78.1481	77.1373
pittsburg_bridges_REL_L	62.1429	63.1429	64.1429	67.0952	67.2381	69.0476	62.1905	65.1905	68.0952
seeds	89.0476	87.1429	90	89.0476	90.4762	90	90.4762	90	90.4762
soybean	87.6954	86.9579	88.4264	88.275	87.3991	88.1387	90.4723	90.3231	90.1803
teaching	69.3763	69.3763	70.6882	72.7097	70.7312	70.043	70.7097	72.0215	72.6882
vertebral_column_3clases	65.1613	65.1613	84.1935	84.1935	67.4194	69.0323	65.4839	65.8065	68.3871
yeast	57.0077	57.4117	57.4133	57.1444	55.6593	56.0005	56.8064	57.8171	57.5475
Average Accuracy	71.385	70.9367	70.9649	71.203	71.6119	72.0101	72.137	<u>73.2906</u>	**74.119**
Average Rank	6.3478	7.2826	5.2391	5.4783	5.8478	5.1957	4.6522	<u>2.6739</u>	**2.2826**

The bold values in each row indicate the best-performing model w.r.t. to the row metric, while the underlined values highlight the second-best-performing model. † denotes the proposed models.

LFDA, and NF-RVFL-R.

Thus the highest accuracy, lowest standard deviation, and superior rankings of the RVFL-X-Auto and RVFL-X-N models demonstrate their exceptional generalization performance. This can be attributed to the proposed RVFL-X models' ability to exploit the rich representational capacity of complex-valued features, which enhances their predictive capabilities compared to existing baseline models.

6.4.4 Evaluation on C-3 Category Datasets

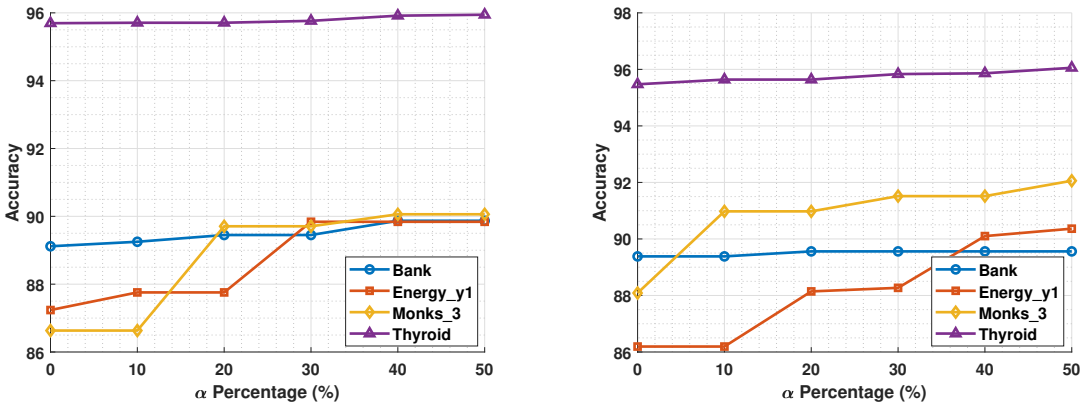
Table 6.3 summarises the experimental results, presenting the accuracy for each dataset, the average accuracy and the rank of the proposed RVFL-X models compared to baseline models across 23 C-3 category (multiclass and small) datasets.

The RVFL-X-Auto and RVFL-X-N models achieve the highest and second-highest accuracies, 74.119% and 73.2906%, respectively, surpassing all baseline models. Notably, the proposed models outperform the NF-RVFL models (a recent SOTA approach leveraging an IF-THEN human-like methodology) by approximately 1%. This highlights the remarkable effectiveness of the proposed models. Their simplicity, combined with the ability to

Table 6.4: Average accuracy, rank and standard deviation (Std. Dev.) across 22 C-4 category datasets.

Dataset ↓ | Model →	**RVFL [1]**	**RVFLwoDL [22]**	**Total-var-RVFL [174]**	**MCVELM [175]**	**NF-RVFL-R [17]**	**NF-RVFL-K [17]**	**NF-RVFL-C [17]**	**RVFL-X-N†**	**RVFL-X-Auto†**
abalone	63.4665	63.4419	63.754	63.6578	63.8253	64.1365	64.0887	63.8735	64.1126
cardiotocography_10clases	65.9924	65.2894	66.1351	66.2307	62.9851	62.8425	69.2395	71.6383	71.3568
cardiotocography_3clases	86.1733	85.5622	86.0808	85.9382	84.6213	84.2003	85.4674	86.3151	86.2216
conn_bench_vowel_deterding	95.8586	95.1515	96.1616	95.7576	87.9798	88.7879	92.2222	96.1616	96.3636
image_segmentation	87.6623	87.0996	88.3117	87.4459	83.2035	83.5931	83.6797	89.4372	89.2641
letter	80.505	79.695	80.67	79.86	61.14	63.205	81.605	82.21	82.33
low_res_spect	87.7605	86.6285	73.9852	74.1721	87.3849	87.5666	87.7605	88.8873	88.8873
oocytes_merluccius_states_2f	91.2846	90.8948	91.8737	91.483	91.3845	91.1894	92.067	92.3625	92.4605
oocytes_trisopterus_states_5b	87.9385	85.853	89.3647	87.8286	87.7205	87.2816	87.7175	89.2584	88.9221
optical	95.4626	94.3594	95.4626	94.8754	93.3808	94.0214	97.0107	97.4733	97.4021
page_blocks	95.4137	95.4137	95.4136	95.3952	94.6645	94.7012	94.7558	95.5963	95.5416
pendigits	98.5353	98.4716	98.6354	98.5262	92.7402	94.6233	97.5345	98.6262	98.6718
semeion	87.0673	82.4233	87.2562	83.0492	88.0726	88.5732	87.3187	90.2072	89.8925
statlog_image	95.1948	94.8918	95.6277	95.2814	91.5584	91.5584	93.0736	96.0606	96.1905
statlog_landsat	82.0513	81.4141	82.2999	81.7249	80.4196	81.3209	81.2898	82.6107	82.6884
statlog_shuttle	98.6483	98.6241	98.681	98.6793	98.6345	98.6638	98.6724	98.7224	98.7328
statlog_vehicle	82.0327	81.5593	82.0313	81.3227	79.1974	79.5475	82.2666	83.2175	83.3317
thyroid	95.625	95.4722	95.7917	95.5833	94.6667	94.7639	96.0278	96.0417	96.0278
wall_following	75.6057	75.239	75.734	75.1845	71.518	72.1223	79.583	81.1777	80.2244
waveform	86.54	85.98	86.56	86.26	86.88	86.68	87.22	87.26	87.24
waveform_noise	86.28	85.02	86.42	85.42	86.28	86.3	86.58	86.46	86.46
wine_quality_white	52.3699	52.5332	52.472	52.4103	52.4106	52.6349	53.4925	53.1047	53.4311
Average Accuracy	85.3395	84.5917	84.9419	84.3676	82.7576	83.1052	85.3942	**86.6683**	86.6252
Average Rank	5.2727	7.0227	4.2727	6	7.5909	6.8409	4.5	1.7955	**1.7045**
Average Std. Dvn.	2.8274	3.0834	4.0949	4.1645	3.0201	3.3523	3.1336	2.9423	**2.8087**

The bold values in each row indicate the best-performing model w.r.t. to the row metric, while the underlined values highlight the second-best-performing model. † denotes the proposed models.



(a) RVFL-X-N (b) RVFL-X-Auto

Figure 6.2: Sensitivity analyses of the proposed RVFL-X-N and RVFL-X-Auto models w.r.t. α.

efficiently handle complex-valued features, demonstrates both their robustness and adaptability.

In terms of rank, RVFL-X-Auto and RVFL-X-N secure the lowest and second-lowest ranks, 2.2826 and 2.6739, respectively, further validating their superior performance. The Friedman test ($F_F = 11.8338 > F_F(11, 143) = 1.9913$ at $\alpha = 0.05$) confirms significant differences among all models on these datasets. Additionally, the Nemenyi post hoc test ($C.D. = 2.5051$) reveals that the proposed RVFL-X models are statistically superior to all baseline models except NF-RVFL-C. However, the proposed models achieve lower ranks than NF-RVFL-C, emphasizing their strong generalization capability and consistent performance across diverse datasets.

6.4.5 Evaluation on C-4 Category Datasets

Table 6.4 presents the experimental results, including the accuracy for each dataset, the average accuracy, the standard deviation, and the rank of the proposed RVFL-X models compared to baseline models across 22 C-4 category (multiclass and large) datasets.

The proposed RVFL-X-N model achieves the highest accuracy of 86.6683%, followed closely by RVFL-X-Auto with 86.6252%. In terms of standard deviation and rank, RVFL-X-Auto demonstrates the lowest average standard deviation and rank among all models. The Friedman test ($F_F = 34.2013 > F_F(11, 143) = 1.9939$ at $\alpha = 0.05$) and the Nemenyi post hoc test ($C.D. = 2.5614$) further confirm that the proposed models are statistically superior to all baseline models, except Total-Var-RVFL. Nevertheless, the lowest ranks of the proposed models highlight their superior generalization and adaptability.

Notably, the RVFL-X models consistently outperform RVFLwoDL (ELM) in terms of rank and standard deviation, emphasizing the critical role of the direct link in enhancing model performance and stability.

Table 6.5: Ablation Study of the RVFL-X-N Model

Proposed Variant ↓ | Dataset →	bank	energy_y1	monks_3	thyroid
RVFL-X-N	**89.8475**	**91.7927**	**92.9615**	**96.0417**
RVFL-X-N$_{\alpha=0}$	89.1175	87.2371	86.6323	95.6944
RVFL-X-N-woDL	89.6042	89.9711	89.8886	95.9028
RVFL-X-N-woDL$_{\alpha=0}$	89.2502	86.0682	87.3530	95.4722

Table 6.6: Ablation Study of the RVFL-X-Auto Model

Proposed Variant↓ | Dataset →	bank	energy_y1	monks_3	thyroid
RVFL-X-Auto	**89.8032**	**90.7555**	**93.1433**	**96.0278**
RVFL-X-Auto$_{\alpha=0}$	89.3829	86.1930	88.0868	95.4722
RVFL-X-Auto-woDL	89.4050	90.4940	91.3333	95.8750
RVFL-X-Auto-woDL$_{\alpha=0}$	89.2723	86.4536	88.4472	95.5000

6.4.6 Sensitivity Analysis

Sensitivity analyses of the proposed RVFL-X-N and RVFL-X-Auto models are conducted w.r.t. the percentage parameter α, which controls the proportion of zeros in the hidden layer weights and bias matrices for regularization. The results, shown in Fig. 6.2 for four datasets across all categories, illustrate the impact of α values ranging from 0% to 50%.

As α increases, the accuracy of both models improves, demonstrating the effectiveness of regularization through sparsity in the weight and bias matrices. Notably, accuracy stabilizes beyond a certain point, indicating that increasing α beyond this threshold does not yield additional benefits. For instance, when $\alpha = 0.5$, the models achieve near-optimal performance, highlighting this value as a practical choice for maximizing accuracy in the proposed RVFL-X models.

6.4.7 Ablation Study

An ablation study is conducted to evaluate the importance of different components in the proposed RVFL-X models. The results are summarized in Table 6.5 and Table 6.6 for RVFL-X-N and RVFL-X-Auto, respectively. In this study, RVFL-X-N with $\alpha = 0$ (indicating no sparsity) is denoted as RVFL-X-N$_{\alpha=0}$, RVFL-X-N without the direct link is referred to as RVFL-X-N-woDL, and RVFL-X-N without the direct link and with $\alpha = 0$ is labeled as RVFL-X-N-woDL$_{\alpha=0}$. A similar nomenclature is used for RVFL-X-Auto variants.

The results demonstrate that the complete versions of RVFL-X-N and RVFL-X-Auto, which include sparsity controlled by the parameter α and incorporate the direct link, consistently outperform their respective ablated variants. This highlights the critical role of sparsity in weights and biases, as well as the inclusion of the direct link, in achieving superior performance in the proposed models.

6.5 Summary

This chapter introduces two innovative mechanisms for transforming real-valued datasets into complex-valued representations: one based on natural transformations and the other leveraging an autoencoder-driven methodology. Using these transformations, we propose RVFL-X, an enhanced extension of the Random Vector Functional Link (RVFL) network, which incorporates complex weights, activation functions, and architectures while producing real-valued outputs.

A rigorous evaluation across 80 UCI datasets, categorized into four groups, demonstrates the superior performance of RVFL-X against 10 SOTA baseline models. Detailed analyses, including accuracy, standard deviation, ranking metrics, and statistical validations

such as the Friedman and Nemenyi post hoc tests, affirm the strong generalization and adaptability of RVFL-X. In every evaluation scenario, RVFL-X achieves the highest average accuracy, the lowest standard deviation, and the most favorable rankings, consistently outperforming competing methods. Furthermore, sensitivity analysis and ablation studies highlight the critical role of specific hyperparameters and architectural components in the model's success.

Chapter 7

Expressive and Interpretable Ensemble Deep Randomized Neural Networks

Outcome of this Chapter

Publication: **M. Sajid**, M. Tanveer and P. N. Suganthan (2025). "Ensemble Deep Random Vector Functional Link Neural Network Based on Fuzzy Inference System," in ***IEEE Transactions on Fuzzy Systems***, v. 33, no. 1, pp. 479-490, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3411614 **[SCI Indexed with Impact Factor.:** 10.2, Q1]

Source Code: https://github.com/mtanveer1/edRVFL-FIS

The previous chapters primarily focused on shallow randomized neural network (RdNN) models. While these models are computationally efficient and exhibit strong generalization in several settings, they often lack the hierarchical feature learning capability required to capture complex data distributions. To address this limitation, ensemble deep random vector functional link (edRVFL) networks have been introduced, which enhance stability and learning capacity by integrating deep architectures with ensemble strategies. Despite these advantages, edRVFL relies on random projections to generate hidden features, which may not adequately capture important nonlinear structures present in the data. Thus, we introduce an edRVFL model integrated with a fuzzy inference system, referred to as edRVFL-FIS, to enhance feature representation and interpretability. In this approach, fuzzified features generated through a fuzzy layer are combined with the original input features to construct an enriched feature space, while supervised defuzzified features further improve the learning process. Different clustering strategies are employed to determine fuzzy rule centers, resulting in three variants: edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C. Extensive experimental evaluations on benchmark datasets demonstrate that the proposed models significantly enhance expressiveness, and interpretability compared to existing approaches.

7.1 Background

Despite several advantages of RVFLs, the conventional RVFL model relies on a shallow architecture with a single hidden layer, which limits its ability to capture complex nonlinear patterns present in real-world data. To overcome this limitation, two major research directions have been explored: deep architectures and ensemble learning [2, 133]. Deep RVFL (dRVFL) networks extend the RVFL framework by stacking multiple hidden layers, enabling richer feature representation and improved modeling capacity. Ensemble deep

RVFL (edRVFL) networks further enhance stability and predictive performance by leveraging implicit ensemble learning, where each hidden layer acts as an individual classifier whose outputs are combined to produce the final prediction.

Over the years, several variants of dRVFL and edRVFL have been developed to improve their applicability across different domains. These models have demonstrated promising performance in applications such as time-series forecasting, medical diagnosis, signal processing, and image recognition [63, 67, 198]. Although deep architectures improve representational power, the randomly generated hidden features may fail to capture important latent patterns in the data, thereby limiting the expressive capability of the network. To enhance the expressive power and interpretability of deep randomized networks, fuzzy logic provides a promising framework. Fuzzy inference systems (FIS) [199] model uncertainty through linguistic rules and membership functions, enabling human-interpretable reasoning based on IF–THEN rules. By integrating fuzzy representations with neural networks, it becomes possible to enhance feature representation while simultaneously improving interpretability. Motivated by these advantages, this chapter introduces an ensemble deep RVFL model integrated with a fuzzy inference system (edRVFL-FIS). In this architecture, a fuzzy layer generates fuzzified features that are combined with the original input features to form an enhanced feature space. These enriched representations are then propagated through the deep RVFL structure, enabling the model to capture complex relationships more effectively while maintaining interpretability through fuzzy rule-based reasoning.

Overall, the models presented in this chapter collectively address two key challenges in randomized neural networks: expressiveness and interpretability through fuzzy feature enhancement and rule-based reasoning.

The key highlights of this chapter are summarized as follows:

[1] We propose the edRVFL-FIS model, which unifies a fuzzy inference system (FIS)

with the ensemble deep RVFL framework to enhance feature representation. In this model, hidden layer features are generated from an enhanced feature space formed by concatenating the original input features with fuzzified features.

[2] Each base learner of the proposed edRVFL-FIS model utilizes two complementary feature enhancement components: (i) unsupervised fuzzy features generated through the fuzzification process and (ii) supervised defuzzified features, enabling improved expressive capability and interpretability.

[3] To further enhance feature diversity, we develop three variants of the proposed model; edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C; based on R-means, K-means, and Fuzzy C-means clustering methods for generating fuzzy rule centers.

[4] Extensive experiments conducted on benchmark datasets demonstrate that the proposed models provide improved feature expressiveness compared to existing randomized neural network models.

7.2 The Proposed Ensemble Deep RVFL Based on a Fuzzy Inference System (edRVFL-FIS)

In this section, we fuse the fuzzy inference system, (*i.e.,* TSK) with the edRVFL structure and propose a hybrid edRVFL-FIS model that has a rich feature representation. The architecture of the proposed edRVFL-FIS is represented in Figure 7.1. In the edRVFL-FIS model, input samples undergo a fuzzification process and form an unsupervised fuzzy layer. Further, the generation of a defuzzified layer takes place via the defuzzification process. The subsequent hidden layer is formed by the input, fuzzified, and preceding hidden layers (except for the first hidden layer), utilizing random projection. Each hidden layer within the

model operates as an autonomous base model (RVFL). We refer to the hidden layer features, original input features, and defuzzified features collectively as enhanced fuzzified features. The sample with fuzzified enhanced features for the lth base model serves as an input and propagates to the output layer of the model. Each base model within the edRVFL-FIS framework behaves as RVFL models that incorporate distinct input features. This architecture of the proposed edRVFL-FIS allows for a structured and modular processing of data, where each hidden layer operates as its own self-contained model within the larger network architecture. Moreover, the ensemble structure leverages diverse supervised defuzzified features for each base model, enhancing richness in feature extraction during output prediction.

The subsequent subsections discuss the method of fuzzification and defuzzification, which will be incorporated in the proposed edRVFL-FIS model for feature enhancement.

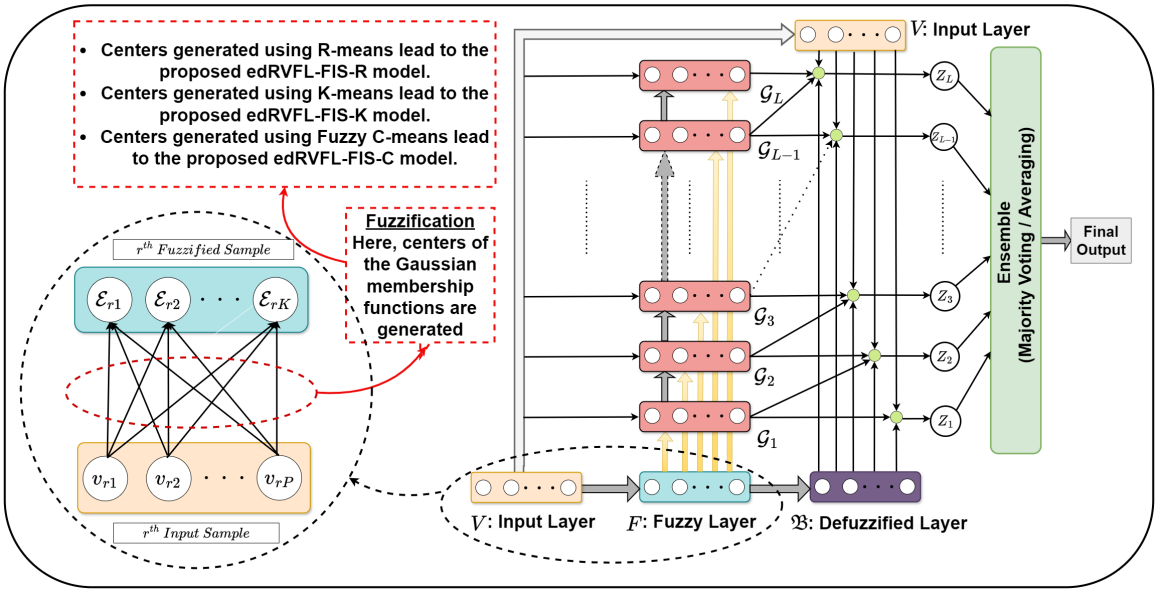


Figure 7.1: The architecture of the edRVFL-FIS model.

7.2.1 Proposed Method: Overall Framework of edRVFL-FIS

Suppose K denotes the total number of fuzzy rules in the fuzzy layer. Let $F \in \mathbb{R}^{M\times K}$ be the fuzzy layer's matrix and $\mathfrak{B} \in \mathbb{R}^{M\times C}$ be the defuzzified matrix corresponding to all

the training samples.

Hidden layer formations and their corresponding outputs:

1st hidden layer and output: The training input matrix U and the fuzzified matrix F traverse the first hidden layer, $\mathcal{G}_1$, through random projection followed by non-linear transformation by employing the activation function ψ. Therefore

$$\mathcal{G}_1 = \psi([U \oplus F]X_1 + \mathbf{1}^t \otimes \zeta_1) \in \mathbb{R}^{M \times g}, \tag{7.1}$$

where $X_1 \in \mathbb{R}^{(L+K)\times g}$ and $\zeta_1 \in \mathbb{R}^{1\times g}$ are the randomly generated weight matrix and bias vector, respectively, for the 1st hidden layer. The first hidden layer generates the first base model for the ensembling. The fuzzified enhanced features (formed by the input, hidden, and defuzzified features) serve as input to the first output layer. Therefore, the first layer's output matrix $T_1 \in \mathbb{R}^{M\times C}$ is defined as:

$$\hat{T_1} = [U \oplus \mathcal{G}_1]\mathcal{Q}_1 + \mathfrak{B} = \mathcal{D}_1\mathcal{Q}_1 + \mathfrak{B}, \tag{7.2}$$

where $\mathcal{Q}_1 \in \mathbb{R}^{(L+g)\times C}$ is the unknown weight matrix connecting the input layer and the 1st hidden layer to the output layer and $\mathcal{D}_1 = [U \oplus \mathcal{G}_1] \in \mathbb{R}^{M\times(L+g)}$.

l^{th} hidden layer and output for $l = 2, 3, \ldots, \mathbf{L}$: The training input matrix U, the previous hidden layer's matrix $\mathcal{G}_{l-1}$ and the fuzzified samples matrix F traverse the lth hidden layer, $\mathcal{G}_l$. Therefore,

$$\mathcal{G}_l = \psi([U \oplus \mathcal{G}_{l-1} \oplus F]X_l + \mathbf{1}^t \otimes \zeta_l) \in \mathbb{R}^{M\times g}, \tag{7.3}$$

where $X_l \in \mathbb{R}^{(L+g+K)\times C}$ and $\zeta_l \in \mathbb{R}^{1\times g}$ are the randomly generated hidden layer weight matrix and bias vector, respectively, for the lth hidden layer. The lth hidden layer behaves

as a base model (RVFL) and the corresponding output matrix $T_l \in \mathbb{R}^{M\times C}$ is defined as:

$$\hat{T}_l = [U \oplus \mathcal{G}_{l-1} \oplus \mathcal{G}_l]\mathcal{Q}_l + \mathfrak{B} = \mathcal{D}_l\mathcal{Q}_l + \mathfrak{B}, \tag{7.4}$$

where $\mathcal{Q}_l \in \mathbb{R}^{(L+2g)\times C}$ is the unknown weight matrix connecting the input layer and the lth hidden layer to the output layer and $\mathcal{D}_l = [U \oplus \mathcal{G}_{l-1} \oplus \mathcal{G}_l] \in \mathbb{R}^{M\times(L+2g)}$.

The anticipated output matrices T_l's (for $l = 1, 2, \ldots, \mathbf{L}$) are combined through majority voting or averaging to find the final predicted output. However, how to find the fuzzy layer's matrix F, the defuzzified matrix $\mathfrak{B}$, and the weights matrix $\mathcal{Q}_l$ connecting the base models (hidden layers) to the output layer remains to answer. We discuss answers to the aforementioned questions in the subsequent subsections.

Remark: The weights of the hidden layers X_l are initialized using different strategies based on the activation function. Specifically, Kaiming initialization [200] is used if the activation function belongs to the ReLU family, Xavier initialization [201] is applied if the activation function belongs to the Sigmoid family, and uniform distribution over the interval $[-1, 1]$ is employed for activation functions that do not belong to either the ReLU or Sigmoid family. The biases ξ_l for each layer are initialized using a uniform distribution over the interval $[0, 1]$.

7.2.1.1 Fuzzification Process for the Proposed Model

The proposed edRVFL-FIS model uses fuzzified matrix F into the generation of the hidden layer process. Therefore, each and every training sample is fuzzified using the TSK FIS. For this, let $\mathcal{F}_{kp}$ be the fuzzy set with the membership function Θ_{kp} of pth component

of kth fuzzy rule of the fuzzy layer, where $p = 1, 2, \ldots, L$, $k = 1, 2, \ldots, K$.

$$\text{If } u_{r1} \text{ is } \mathcal{F}_{k1} \text{ and } u_{r2} \text{ is } \mathcal{F}_{k2} \ \ldots \text{ and } u_{rL} \text{ is } \mathcal{F}_{kL}$$

then, by considering the first-order TSK fuzzy system, we get

$$y_{rk} = \xi_k(u_{r1}, u_{r2}, \ldots, u_{rL}) = \sum_{p=1}^{L} \beta_{kp} u_{rp}, \ \ k = 1, 2, \ldots, K.$$

Here β_{kp} are the randomly generated coefficients.

The weighted fire strength of the kth fuzzy rule in the fuzzy layer is given as:

$$\omega_{rk} = \frac{\prod_{p=1}^{L} \Theta_{kp}(u_{rp})}{\sum_{k=1}^{K} \prod_{p=1}^{L} \Theta_{kp}(u_{rp})}. \tag{7.5}$$

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

$$\Theta_{kp}\left(u_{rp}\right) = e^{-\left(\frac{u_{rp}-c_{kp}}{\sigma_{kp}}\right)^2}, \ \text{ for } \ p = 1, 2, \ldots, L, \tag{7.6}$$

where $c_k = (c_{k1}, c_{k2}, \ldots, c_{kL})$ is the center and $\sigma_k = (\sigma_{k1}, \sigma_{k2}, \ldots, \sigma_{kL})$ is the standard deviation in the kth fuzzy rule. The number of fuzzy rules in the fuzzy layer is equal to the number of centers.

In the proposed edRVFL-FIS model, we generate the centers c_k in the fuzzy layer using three different clustering methods, which are:

[1] Randomly initialized centers (say R-means).

[2] Using K-means clustering.

[3] Using Fuzzy C-means clustering.

Remark: Three distinct variants of edRVFL-FIS arise from the application of three different center initialization approaches, thereby enhancing the diversity within our proposed model. The variations of the proposed edRVFL-FIS initialized with R-means, K-means, and Fuzzy C-means clustering techniques are denoted as edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C, respectively.

Now, the unsupervised fuzzified vector of the rth training sample u_r is given as:

$$\mathcal{E}_r = (\omega_{r1} y_{r1}, \omega_{r2} y_{r2}, \ldots, \omega_{rK} y_{rK}) \,,\ r = 1, 2, \ldots, M. \tag{7.7}$$

The fuzzified matrix F corresponding to the input matrix U is given as:

$$F = \left[\mathcal{E}_1^t, \mathcal{E}_2^t, \ldots, \mathcal{E}_M^t\right]^t \in \mathbb{R}^{M \times K}. \tag{7.8}$$

7.2.1.2 Defuzzification Process for the edRVFL-FIS

Here, we articulate the defuzzification process for the fuzzified vectors. This process enables the defuzzified output from the fuzzy layer of our proposed edRVFL-FIS model to contribute to the anticipation of the final output layers alongside the original features (via direct links) and the hidden layers. Since the target matrix $T = (t_1^t, t_2^t, \ldots, t_M^t)^t \in \mathbb{R}^{M \times C}$ has C classes, therefore the defuzzified output must have C features. The defuzzified output

$\mathfrak{B}_r \in \mathbb{R}^{1\times C}$ of the sample u_r is defined as follows:

$$\begin{aligned}
\mathfrak{B}_r &= \left(\sum_{k=1}^{K}\omega_{rk}\lambda_{k1}y_{rk},\ldots,\sum_{k=1}^{K}\omega_{rk}\lambda_{kC}y_{rk}\right)\\
&= \left(\sum_{k=1}^{K}\omega_{rk}\lambda_{k1}\sum_{p=1}^{L}\beta_{kp}u_{rp},\ldots,\sum_{k=1}^{K}\omega_{rk}\lambda_{kC}\sum_{p=1}^{L}\beta_{kp}u_{rp}\right)\\
&= \Bigg(\omega_{r1}\sum_{p=1}^{L}\beta_{1p}u_{rp}\lambda_{11}+\cdots+\omega_{rK}\sum_{p=1}^{L}\beta_{Kp}u_{rp}\lambda_{K1},\\
&\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
&\qquad \omega_{r1}\sum_{p=1}^{L}\beta_{1p}u_{rp}\lambda_{1C}+\cdots+\omega_{rK}\sum_{p=1}^{L}\beta_{Kp}u_{rp}\lambda_{KC}\Bigg)\\
&= \left(\omega_{r1}\sum_{p=1}^{L}\beta_{1p}u_{rp},\ldots,\omega_{rK}\sum_{p=1}^{L}\beta_{Kp}u_{rp}\right)\Lambda, \qquad (7.9)
\end{aligned}$$

where

$$\Lambda = \begin{bmatrix}\lambda_{11} & \cdots & \lambda_{1C}\\ \vdots & \vdots & \vdots\\ \lambda_{K1} & \cdots & \lambda_{KC}\end{bmatrix},$$

and λ_{kc} is the parameter used in the THEN part of the fuzzy inference system used in the proposed edRVFL-FIS model. The parameters λ_{kc} are unknown and are estimated along with the output layer weights of the edRVFL-FIS model in subsection 7.2.1.3. Therefore, for the input matrix U, the defuzzified matrix is given as:

$$\mathfrak{B} = \left[\mathfrak{B}_1^t, \mathfrak{B}_2^t, \ldots, \mathfrak{B}_M^t\right]^t \in \mathbb{R}^{M\times C}. \qquad (7.10)$$

$$\Rightarrow \mathfrak{B} = \begin{bmatrix}\left(\omega_{11}\sum_{p=1}^{L}u_{1p}\beta_{1p},\cdots,\omega_{1K}\sum_{p=1}^{L}u_{1p}\beta_{Kp}\right)\Lambda\\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots\\ \left(\omega_{M1}\sum_{p=1}^{L}u_{Mp}\beta_{1p},\cdots,\omega_{MK}\sum_{p=1}^{L}u_{Mp}\beta_{Kp}\right)\Lambda\end{bmatrix}$$

$$\Rightarrow \mathfrak{B} = \left(\Omega \odot \begin{bmatrix} \sum_{p=1}^{L} u_{1p}\beta_{1p} & \cdots & \sum_{p=1}^{L} u_{1p}\beta_{kp} \\ \vdots & \vdots & \vdots \\ \sum_{p=1}^{L} u_{Mp}\beta_{1p} & \cdots & \sum_{p=1}^{L} u_{Mp}\beta_{kp} \end{bmatrix} \right) \Lambda$$

$$\Rightarrow \mathfrak{B} = \left[\Omega \odot \left(U\beta^t\right)\right] \Lambda. \tag{7.11}$$

Here $\odot$ represents the componentwise matrix multiplication.

$$\beta = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1L} \\ \vdots & \vdots & \vdots \\ \beta_{K1} & \cdots & \beta_{KL} \end{bmatrix} \text{ and } \quad \Omega = \begin{bmatrix} \omega_{11} & \cdots & \omega_{1K} \\ \vdots & \vdots & \vdots \\ \omega_{M1} & \cdots & \omega_{MK} \end{bmatrix}. \tag{7.12}$$

7.2.1.3 Output Layers Parameters for Each Base Model of the Proposed Model

To get the output of the base models, we put the value of $\mathfrak{B}$ from Eq. (7.11) in (7.2) and (7.4), we get

$$\begin{aligned} \hat{T}_l &= \mathcal{D}_l \mathcal{Q}_l + \left[\Omega \odot \left(U\beta^t\right)\right] \Lambda \\ &= \left[\mathcal{D}_l \ \oplus \ \Omega \odot \left(U\beta^t\right)\right] \begin{bmatrix} \mathcal{Q}_l \\ \Lambda \end{bmatrix} \\ &= \mathcal{J}_l \mathcal{Q}_l^{\Lambda}, \end{aligned} \tag{7.13}$$

$$\text{where } \mathcal{J}_l = \left[\mathcal{D}_l \ \oplus \ \Omega \odot \left(U\beta^t\right)\right] \in \mathbb{R}^{M \times N}, \tag{7.14}$$

$$\mathcal{Q}_l^{\Lambda} = \begin{bmatrix} \mathcal{Q}_l \\ \Lambda \end{bmatrix} \in \mathbb{R}^{N \times C}. \tag{7.15}$$

Here, $N = L + g + K$ if $l = 1$ and $N = L + 2g + K$ if $l > 1$.

To get the output of each base model of the edRVFL-FIS, $\mathcal{Q}_l^{\Lambda}$ is the matrix of unknowns that need to be found. The resultant optimization problem of Eq. (7.13) is given as:

$$(\mathcal{Q}_l^{\Lambda})_{\min} = \underset{\mathcal{Q}_l^{\Lambda}}{\operatorname{argmin}} \frac{\mathcal{C}}{2}\|\mathcal{J}_l\mathcal{Q}_l^{\Lambda} - T\|_2^2 + \frac{1}{2}\|\mathcal{Q}_l^{\Lambda}\|_2^2. \tag{7.16}$$

The weight matrix $\mathcal{Q}_l^{\Lambda}$ is computed as follows:

$$(\mathcal{Q}_l^{\Lambda})_{\min} = \begin{cases} \left(\mathcal{J}_l^t\mathcal{J}_l + \frac{1}{\mathcal{C}}I\right)^{-1}\mathcal{J}_l^t T, & N \leq M, \\ \mathcal{J}_l^t\left(\mathcal{J}_l\mathcal{J}_l^t + \frac{1}{\mathcal{C}}I\right)^{-1} T, & M < N, \end{cases} \tag{7.17}$$

where I is an identity matrix of conformal dimension.

7.2.1.4 Final Output: Test Condition

For an unknown sample u, the output of the lth $(l = 1, 2, \ldots, \mathbf{L})$ base model is

$$T_l(u) = [u \oplus \mathcal{G}_l(u)]\mathcal{Q}_l + \left[\Omega \odot \left(u\beta^t\right)\right]\Lambda. \tag{7.18}$$

Here, $\mathcal{G}_l(u)$ is the hidden layer output corresponding to the sample u. The base model outputs, $T_l(u)$, are combined using majority voting to get the final output as follows:

$$\text{Final Output}(u) = \underset{j}{\operatorname{argmax}} \sum_{l=1}^{\mathbf{L}} \mathbb{I}(T_l(u) = j), \tag{7.19}$$

where $\mathbb{I}(T_l(u) = j)$ is the indicator function that evaluates to 1 if $T_l(u) = j$, and 0 otherwise.

7.2.1.5 Further Takeaways from the Defuzzification Matrix of the Proposed Model

Let $\#Row(\mathcal{Q}_l)$ denote the number of rows in the matrix $\mathcal{Q}_l$, $\mathbf{0}$ be a matrix of dimension $\#Row(\mathcal{Q}_l)$ with all the entries equal zero and I_K as an identity matrix of dimension K. Since,

$$\mathcal{Q}_l^{\Lambda} = \begin{bmatrix} \mathcal{Q}_l \\ \Lambda \end{bmatrix}, \tag{7.20}$$

multiplying (from left) both sides by the matrix $[\mathbf{0}, I_K]$, we get

$$\begin{aligned} [\mathbf{0}, I_K]\mathcal{Q}_l^{\Lambda} &= [\mathbf{0}, I_K] \begin{bmatrix} \mathcal{Q}_l \\ \Lambda \end{bmatrix} \\ \Rightarrow \Lambda &= [\mathbf{0}, I_K]\mathcal{Q}_l^{\Lambda}. \end{aligned} \tag{7.21}$$

Eq. (7.21) implies that the matrix Λ holds distinct values for each base model (hidden layer l) within the ensemble, emphasizing the uniqueness of each model's contribution. The ensemble structure strategically harnesses the diverse supervised defuzzified matrices $\mathfrak{B}$ across multiple base models, capitalizing on the inherent diversity embedded within the proposed edRVFL-FIS model. Consequently, the proposed edRVFL-FIS demonstrates rich feature extraction capability achieved by leveraging original (through direct link), hidden layer, and defuzzified features.

Algorithm 7.4 The proposed edRVFL-FIS models.

1: **Input:** Dataset of M samples, L features and C classes.
2: **Input:** Input matrix U with dimensions $M \times L$.
3: **Input:** Output matrix T with dimensions $M \times C$.
4: **Parameters:** $\mathcal{C}, \mathbf{L}, g$ and K represent the regularization parameter, number of hidden layers, number of hidden nodes and number of fuzzy nodes, respectively.
5: **Input Sample:** Let u_r be the rth training sample, where $r = 1, 2, \ldots, M$.
6: **Generate:** $\beta \in \mathbb{R}^{K \times L}$ randomly.
7: **Calculate:** Centers of the kth fuzzy rule, *i.e.,* $c_k = (c_{k1}, c_{k2}, \ldots, c_{kL})$ for $k = 1, 2, \ldots, K$ of the fuzzy layer using R-means, K-means, or fuzzy C-means clustering technique.
8: **Calculate:** ω_{rk} using Eq. (7.5).
9: **Calculate:** $\mathcal{E}_r$ using Eq. (7.7).
10: **Calculate:** F using Eq. (7.8).
11: **Construct:** $\mathfrak{B}$ using Eq. (7.11).
12: **Construct:** β and Ω using Eq. (7.12).
13: **Calculate:** $\mathcal{G}_1$ using Eq. (7.1).
14: **Construct:** T_1 using Eq. (7.2).
15: **Calculate:** $\mathcal{G}_l$ using Eq. (7.3).
16: **Construct:** T_l using Eq. (7.4).
17: **Construct:** $\mathcal{J}_l$ using Eq. (7.14).
18: **Construct:** $\mathcal{Q}_l^{\Lambda}$ using Eq. (7.15).
19: **Find:** The output weight parameters for each base model $(\mathcal{Q}_l^{\Lambda})_{\min} = \begin{bmatrix} \mathcal{Q}_l \\ \Lambda \end{bmatrix}_{\min}$ using Eq. (7.17) for $l = 1, 2, \ldots, \mathbf{L}$.
20: **Validate:** For an unknown sample u, calculate the output of the lth base model using Eq. (7.18).
21: **Output:** Final output of u is obtained using Eq. (7.19).

Source code link: https://github.com/mtanveer1/edRVFL-FIS

7.2.2 Time and Space Complexity of the Proposed edRVFL-FIS Models

The complexity of the proposed edRVFL-FIS models primarily depends on two components: (a) clustering technique and (b) requirement of matrix inversion in (7.17). The number of hidden layers **L** for the edRVFL-FIS has taken ≤ 10 (see experimental setup), which is very small compared to other variables; therefore, we generally do not give much attention to **L** in the computational complexity.

Time/Temporal Complexity: Following [172], the time complexity of K-means and Fuzzy C-means are $\mathcal{O}((MLK)iter)$ and $\mathcal{O}((MLK^2)iter)$, respectively, where "$iter$" represents the number of iterations. The time complexity of R-means is $\mathcal{O}(K)$. Following [2], time complexity in computing inverse in (7.17) is $\mathcal{O}(\mathcal{J}_{Time})$, where $\mathcal{O}(\mathcal{J}_{Time}) = \mathcal{O}(M^3)$ if $M \leq N$ or $\mathcal{O}(\mathcal{J}_{Time}) = \mathcal{O}(N^3)$ if $N < M$. Therefore, total time complexity for edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C are $\mathcal{O}(\mathcal{J}_{Time}) + \mathcal{O}(K)$, $\mathcal{O}(\mathcal{J}_{Time}) + \mathcal{O}((MLK)iter)$ and $\mathcal{O}(\mathcal{J}_{Time}) + \mathcal{O}((MLK^2)iter)$, respectively.

Space/Spatial Complexity: Following [202], the space complexity of Fuzzy C-means is $\mathcal{O}((L + K)M)$ and the space complexity of K-means and R-means is $\mathcal{O}((M + K)L)$. Following [2], space complexity in computing inverse in (7.17) is $\mathcal{O}(\mathcal{J}_{Space})$, where $\mathcal{O}(\mathcal{J}_{Space}) = \mathcal{O}(M^2)$ if $M \leq N$ or $\mathcal{O}(\mathcal{J}_{Space}) = \mathcal{O}(N^2)$ if $N < M$. Therefore, total space complexity for edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C are $\mathcal{O}(\mathcal{J}_{Space}) + \mathcal{O}((M + K)L)$, $\mathcal{O}(\mathcal{J}_{Space}) + \mathcal{O}((M + K)L)$ and $\mathcal{O}(\mathcal{J}_{Space}) + \mathcal{O}((L + K)M)$, respectively.

7.3 Experiments and Results

The section presents comprehensive details on the datasets and the compared models. For the hyperparameter tuning, we use Bayesian optimization [203] and the detailed experimental setup is discussed in the subsequent subsection. Subsequently, we delve into the experimental results and conduct statistical analyses on both UCI and NDC datasets. Finally, we examine the influence of fuzzy rules, hidden layers and robustness of the proposed edRVFL-FIS models.

7.3.1 Experimental Setup

Experimental software: The experimental procedures were conducted on a computing system with Python 3.11 featuring an Intel(R) Xeon(R) Platinum 8260 CPU @ 2.30GHz, 24 cores, 48 logical processors, and 256 GB RAM. The system operated on a Windows 10 platform.

Experimental procedure for UCI datasets: We use the Bayesian optimization for hyperparameter selection for the UCI datasets. The UCI datasets considered has already the predefined 4-fold training, validation and testing splittings[1]. The training and validation partitions are used for hyperparameter selection, and the testing partition is used to get the testing accuracy and testing standard deviation (referred to as Accuracy and Standard deviation in this chapter). For the Bayesian optimization max_evals (no. of iterations) $= 100$.

Experimental procedure for NDC datasets: The NDC datasets have already the predefined training and testing partition [204]. Grid search with a 4-fold cross-validation technique is used to optimize the model's hyperparameters for the NDC datasets. In order to accomplish this, the training dataset is divided into four distinct, non-overlapping subsets, or "folds". One subset was set to be designated for validation, and the other four were used for

[1] https://github.com/lingping-fuzzy/UCI-data-correct-split

training in every phase. For every fold, validation accuracy was separately calculated using different hyperparameter sets. Corresponding to the best average validation accuracy, the bets hyperparameters are selected. The models' testing accuracy is determined by the testing accuracy in the testing set corresponding to the best hyperparameters. For the Bayesian optimization max_evals (no. of iterations) $= 100$.

Hyperparameter selection: The hyperparameters are tuned using the Bayesian optimization technique. The regularization parameter, $\mathcal{C}$, for each model are chosen from the interval $(2^{-12}, 2^{6})$. Two activation functions, namely, Relu and Sigmoid, are tuned for each model.

For the RVFL and ELM-based models (for shallow as well as deep), the hidden layers nodes g are tuned within the range $128 : 1 : 4096$. Following [205], we take the graph regularization parameter for the edEGERVFL equal to $\mathcal{C}$. For the dRVFL, edRVFL, edEGERVFL and proposed edRVFL-FIS models, the hidden layers are tuned within the range of $1 : 1 : 10$. For the H-ELM, the hidden layers are tuned within the range of $1 : 1 : 3$

Following [2], we used a two-stage tuning approach for the dRVFL, edRVFL, edEGERVFL, and edRVFL-FIS models. In the first stage, we tune the regularization parameter, hidden layers nodes, and the number of hidden layers - and get the best regularization parameter ($\mathcal{C}^*$), hidden layers nodes (g^*), and the number of hidden layers (l^*), respectively. During the second stage, we fine-tune the regularization parameter in the neighbourhood of $\mathcal{C}^*$, i.e., in the set $(\mathcal{C}^*/2, 2\mathcal{C}^*)$, the hidden layer nodes are fine-tuned in the set $g^* - 50 : 1 : g^* + 50$ and number of hidden layers are fine-tuned within the range $l^* - 1 : 1 : l^* + 1$ if $1 < l^* < 10$; within the range $9 : 1 : 10$ if $l^* = 10$; within the range $1 : 1 : 2$ if $l^* = 1$.

Furthermore, for the proposed edRVFL-FIS models, the optimal number of fuzzy nodes in the first stage (denoted as K^*) is determined through tuning and then fine-tuned in the second stage as follows.

Let δ represent $\left\lfloor \frac{\text{number of samples in each fold}}{2} \right\rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor operation.

[1] **Category-1 datasets** ($100 \leq$ #samples ≤ 1000):

- In the first stage, K is tuned within the range $25 : 1 : \delta$.
- In the second stage, K is fine-tuned within the range $K^* - 20 : 1 : K^* + 20$.

[2] **Category-2 datasets** ($1000 <$ #samples $\leq 10,000$):

- In the first stage, K is tuned within the range $100 : 1 : \delta$.
- In the second stage, K is fine-tuned within the range $K^* - 50 : 1 : K^* + 50$.

[3] **NDC datasets** ($10,000 <$ #samples ≤ 1 million):

- In the first stage, K is tuned within the range $128 : 1 : 4096$.
- In the second stage, K is fine-tuned within the range $K^* - 50 : 1 : K^* + 50$.

The selection of feature groups for BLS is made from the range $N_2 = 1 : 1 : 21$, whereas the selection of feature nodes within each feature group is made from the range $N_1 = 128 : 1 : 200$. The Fuzzy BLS model selects the number of fuzzy groups from the range $N_{fg} = 1 : 2 : 21$ and the number of fuzzy nodes in each fuzzy group from the range $N_{fn} = 128 : 1 : 200$. The number of enhancement nodes for BLS and Fuzzy BLS is selected from the range $N_3 = 600 : 1 : 701$.

7.3.2 Datasets

To evaluate the efficacy of the proposed edRVFL-FIS models, we employ datasets from UCI [7] and NDC [204]. We use the predefined training, validation and testing splitting for UCI datasets[2]. We categorize UCI datasets into two groups: Category-1 with sample sizes

[2] https://github.com/lingping-fuzzy/UCI-data-correct-split

ranging from 100 to 1000, and Category-2 with sample sizes between 1000 and 10000. These datasets span diverse domains and sizes, with detailed statistics provided in Tables 7.1 and 7.2. The NDC dataset, generated using David Musicant's NDC Data Generator [204], spans from 10 thousand to 1 million samples, consistently featuring 32 features.

Table 7.1: Details of the Category-1 datasets used in experiments sourced from the UCI [7] repository.

Dataset	Samples	Features	Classes
balance_scale	625	5	3
breast_cancer_wisc	699	10	2
dermatology	366	35	6
echocardiogram	131	11	2
ecoli	336	8	8
energy_y1	768	9	3
energy_y2	768	9	3
flags	194	29	8
heart_switzerland	123	13	5
iris	150	5	3
led_display	1000	8	10
lymphography	148	19	4
mammographic	961	6	2
pittsburg_bridges_REL_L	103	8	3
pittsburg_bridges_T_OR_D	102	8	2
statlog_heart	270	14	2
statlog_vehicle	846	19	4
tic_tac_toe	958	10	2
zoo	101	17	7

Table 7.2: Details of the Category-2 datasets used in experiments sourced from the UCI [7] repository.

Dataset	Samples	Features	Classes
car	1728	7	4
cardiotocography_10clases	2126	22	10
cardiotocography_3clases	2126	22	3
chess_krvkp	3196	37	2
contrac	1473	10	3
mushroom	8124	22	2
titanic	2201	4	2
wine_quality_red	1599	12	6
yeast	1484	9	10

7.3.3 Compared Models

We compare the proposed edRVFL-FIS models with eight baseline and state-of-the-art (SOTA) RdNN-based models. The enumerated models under examination include:

[1] RVFL: Standard shallow RVFL model [1].

[2] RVFLwoDL (ELM): RVFL without direct link (also known as "extreme learning machine") [22].

[3] BLS: Broad learning system [3].

[4] H-ELM: Hierarchical ELM model [206].

[5] dRVFL: Deep RVFL model [2].

[6] edRVFL: Ensemble deep RVFL model [2].

[7] Fuzzy BLS: Neuro-Fuzzy-based BLS model [56].

[8] edEGERVFL: Ensemble deep extended graph embedded RVFL [205].

[9] edRVFL-FIS-R: The proposed ensemble deep RVFL based on FIS with R-means clustering.

[10] edRVFL-FIS-K: The proposed ensemble deep RVFL based on FIS with K-means clustering.

[11] edRVFL-FIS-C: The proposed ensemble deep RVFL based on FIS with fuzzy C-means clustering.

Table 7.3: Classification accuracies of the proposed edRVFL-FIS models and the baseline models on Category-1 UCI datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	BLS [3]	H-ELM [206]	dRVFL [2]	edRVFL [2]	Fuzzy BLS [56]	edEGERVFL [205]	edRVFL-FIS-R†	edRVFL-FIS-K†	edRVFL-FIS-C†
balance-scale	**98.2**	98	96.2	96.2	96.2	96.2	98	98	96.2	96.2	96.2
breast-cancer-wisc	96.49	95.86	96.59	95.86	97	97.14	95.59	97	96.57	97.29	**97.43**
dermatology	96.16	97.31	97.41	97.03	97.64	98.31	97.41	95.95	97.97	98.3	**98.37**
echocardiogram	84.05	83.85	84.16	83.97	84.29	84.09	83.16	84.85	84.84	84.85	**85.61**
ecoli	85.21	85.01	85.25	85.23	87.5	87.5	85.35	87.75	87.75	87.2	**87.9**
energy-y1	93.86	92.75	93.77	93.86	92.97	93.1	93.86	93.87	94.3	94.25	**94.4**
energy-y2	90.09	89.89	90.56	90.09	89.58	90.76	91.23	90.85	91.23	91.23	**91.45**
flags	48.56	47.64	48.58	47.64	53.85	48.58	53.85	48.85	49.36	54.15	**54.87**
heart-switzerland	42.2	41	43	43	43	43	43	41	**45**	43	**45**
iris	97.29	94.27	97.36	97.32	95.95	96.62	95.44	**97.49**	96.62	97.3	**97.49**
led-display	73.1	72.3	73.1	73.1	72.6	**73.6**	73.1	**73.6**	**73.6**	**73.6**	**73.6**
lymphography	85.69	85.51	88.05	87.17	88.51	88.81	87.07	88.16	88.51	**89.13**	89.06
mammographic	80.45	79	81.69	79	82.6	81.69	79	83.69	82.25	83.75	**84.15**
pittsburg-bridges-REL-L	62.3	64.05	65.47	65.17	65.71	65.48	65.33	65.64	65.81	**65.87**	**65.87**
pittsburg-bridges-T-OR-D	86.1	85.9	86.56	86.36	80.95	86.7	84.12	85.71	86.52	86.65	**88.05**
statlog-heart	86.01	82.58	84.03	86.01	86.19	87.39	87.28	87.39	86.57	87.23	**87.87**
statlog-vehicle	82.73	82.29	82.75	82.75	82.23	83.12	82.97	**84.03**	80.92	83.12	83.12
tic-tac-toe	96.15	97.95	96.97	96.35	98.64	98.74	98.64	98.95	**99.06**	**99.06**	**99.06**
zoo	96.82	97.81	97.89	96.82	96.43	97.62	97.69	97.89	97.97	**98.12**	**98.12**
Average Accuracy	83.23	82.79	83.65	83.31	83.78	84.13	83.79	84.25	84.27	84.75	**85.14**
Average Rank	8.21	9.39	6.71	8.11	7.05	5.21	7.03	4.66	4.58	3.26	**1.79**

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

Table 7.4: Wilcoxon-signed-rank test of the proposed edRVFL-FIS models *w.r.t.* baseline models on Category-1 UCI datasets. † represents the proposed models.

Proposed →	edRVFL-FIS-R†		edRVFL-FIS-K†		edRVFL-FIS-C†	
Baseline ↓	p-value	Null hypothesis	p-value	Null hypothesis	p-value	Null hypothesis
RVFL [1]	0.0035	Rejected	0.00112	Rejected	0.00044	Rejected
RVFLwoDL [22]	0.00054	Rejected	0.00026	Rejected	0.00018	Rejected
BLS [3]	0.01778	Rejected	0.00046	Rejected	0.0002	Rejected
H-ELM [206]	0.00158	Rejected	0.00036	Rejected	0.0002	Rejected
dRVFL [2]	0.0139	Rejected	0.00038	Rejected	0.0002	Rejected
edRVFL [2]	0.42372	Not Rejected	0.00988	Rejected	0.00044	Rejected
Fuzzy BLS [56]	0.02642	Rejected	0.00194	Rejected	0.0005	Rejected
edEGERVFL [205]	0.9442	Not Rejected	0.13104	Not Rejected	0.00596	Rejected

Table 7.5: Classification accuracies of the proposed edRVFL-FIS and the baseline models on Category-2 UCI datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	BLS [3]	H-ELM [206]	dRVFL [2]	edRVFL [2]	Fuzzy BLS [56]	edEGERVFL [205]	edRVFL-FIS-R†	edRVFL-FIS-K†	edRVFL-FIS-C†
car	96.54	96.34	96.64	95.53	97.8	98.03	96.64	97.8	98.09	97.69	**98.32**
cardiotocography-10clases	82.2	82.85	82.36	82.77	82.3	82.11	82.77	82.86	82.91	82.93	**83.24**
cardiotocography-3clases	91.57	91.7	91.78	91.89	91.67	92.56	91.79	92.23	92.18	**92.74**	92.66
chess-krvkp	98.39	97.87	98.39	98.39	98.4	98.69	98.42	**98.87**	98.68	98.68	98.78
contrac	53	50.77	52.11	**52.35**	52.73	52.63	52.47	51.77	52.49	52.92	52.77
mushroom	99.2	99	99	99.2	99.9	**100**	**100**	**100**	**100**	99	**100**
titanic	78.15	77.95	79.56	78.35	78.95	78.59	78.69	**81.23**	78.95	80.02	80.58
wine-quality-red	65.29	66.11	65.35	65.78	64.3	67.11	65.02	65.69	**67.78**	66.65	**67.78**
yeast	59.05	59.94	59.57	59.78	60.27	59.01	59.57	58.92	60.02	60.26	**60.57**
Average Accuracy	80.38	80.28	80.53	80.45	80.7	80.97	80.6	81.04	81.23	81.21	**81.63**
Average Rank	8.39	8.56	7.89	7.67	6.56	5.44	6.83	5.06	3.83	3.94	**1.83**

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

Table 7.6: Wilcoxon-signed-rank test of the proposed edRVFL-FIS models *w.r.t.* baseline models on Category-2 UCI datasets. † represents the proposed models.

Proposed →	edRVFL-FIS-R†		edRVFL-FIS-K†		edRVFL-FIS-C†	
Baseline ↓	p-value	Null hypothesis	p-value	Null hypothesis	p-value	Null hypothesis
RVFL [1]	0.03714	Rejected	0.01953	Rejected	0.01246	Rejected
RVFLwoDL [22]	0.003906	Rejected	0.01415	Rejected	0.008789	Rejected
BLS [3]	0.01277	Rejected	0.01403	Rejected	0.00861	Rejected
H-ELM [206]	0.008789	Rejected	0.01752	Rejected	0.008969	Rejected
dRVFL [2]	0.04181	Rejected	0.1229	Not Rejected	0.008969	Rejected
edRVFL [2]	0.2924	Not Rejected	0.4061	Not Rejected	0.01403	Rejected
Fuzzy BLS [56]	0.01403	Rejected	0.1225	Not Rejected	0.01391	Rejected
edEGERVFL [205]	0.4002	Not Rejected	0.5136	Not Rejected	0.04206	Rejected

7.3.4 Experimental Results, Discussions and Statistical Analysis on Category-1 UCI Dataset

The proposed edRVFL-FIS models are evaluated against existing baseline models using Category-1 datasets from the UCI repository. The accuracy of the models based on Category-1 datasets is reported in Table 7.3. The results presented in Table 7.3 indicate that the accuracy of all the proposed models surpasses that of the baseline models. Notably, edRVFL-FIS-C achieved the highest average accuracy with 85.14%, whereas edRVFL-FIS-K and edRVFL-FIS-R followed closely with an average accuracy of 84.75% and 84.27%, respectively. The proposed edRVFL-FIS demonstrate exceptional generalization performance, characterized by consistently higher accuracy in each dataset as well as higher average accuracy compared to baseline models. The results imply that the edRVFL-FIS models excel in feature extraction by leveraging fuzzy and defuzzification processes.

The proposed models consistently secure the lowest average ranks, emphasizing their superior generalization performance. Specifically, edRVFL-FIS-C achieves the lowest average rank at 1.79, followed by edRVFL-FIS-K at 3.26 and edRVFL-FIS-C at 4.58. In contrast, RVFLwoDL performs least favourably, with an average rank of 9.39. This underscores the clear superiority of the proposed edRVFL-FIS models over existing shallow and deep SOTA variants of the RdNN family.

For the Friedman Test, in our experiment, $\mathfrak{M} = 11$ (no. of models) and $\mathfrak{D} = 19$ (no. of datasets) , therefore, we get $\chi_F^2 = 91.83$ and $F_F = 16.84$. Referring to the F-distribution table, we note that $F_F(10, 180) = 1.88$ at a 5% significance level. The null hypothesis is rejected because $16.84 > 1.88$, indicating significant differences among the models. Now, we employ the Wilcoxon-signed-rank test to assess the pairwise significant distinction between the proposed edRVFL-FIS models and existing baseline models. This test operates on the assumption that baseline models and the proposed edRVFL-FIS models exhibit equivalence performance under the null hypothesis. For each dataset, the absolute difference in ranks between the two models is ranked, considering the sign of the differences. If the p-value of the model comparison falls below 0.05, the null hypothesis is rejected, indicating that the model with the lower rank is statistically superior to the one with the higher rank. In Table 7.4, the p-values and the corresponding status of the null hypothesis are shown. The pairwise comparisons between the proposed edRVFL-FIS models and baseline models underscore a compelling narrative, *i.e.*, the consistent rejection (with most of the baseline models) of the null hypothesis signifies a statistical superiority of all three proposed models, *i.e.,* edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C over SOTA RdNN models. This analysis solidifies the commendable performance and statistical excellence of the proposed edRVFL-FIS models across the board.

7.3.5 Experimental Results, Discussions and Statistical Analysis on Category-2 UCI Dataset

The performance evaluation of models based on Category-2 datasets is detailed in Table 7.5. Among the models assessed, the proposed edRVFL-FIS-C, edRVFL-FIS-K, and edRVFL-FIS-R exhibit the highest average accuracies at 81.63%, 81.21%, and 81.23%, respectively. In contrast, the neuro-fuzzy-based BLS model, *i.e.,* Fuzzy BLS, displays the

lowest accuracy among all compared models. This underscores the significance of the learning process and adaptation of fuzzy rules within the FIS framework, as their efficacy directly impacts model performance. The analysis affirms that the proposed edRVFL-FIS models adeptly extract deep features from dataset samples by effectively employing fuzzification and defuzzification processes. A parallel observation emerges when considering the average ranks of the models, as outlined in the final row of Table 7.5. The proposed edRVFL-FIS-C and edRVFL-FIS-K models boast the lowest ranks, standing at 1.83 and 3.94, respectively, while the edRVFL-FIS-R model follows closely with a rank of 3.83. In contrast, RVFLwoDL lags behind with the least favourable ranking, averaging 8.56.

In our experiment, where $\mathfrak{M} = 11$ and $\mathfrak{D} = 9$, we obtained $\chi_F^2 = 38.59$ and $F_F = 6.01$. From F-distribution table, we get $F_F(10, 80) = 3.35$ at a 5% significance level. The rejection of the null hypothesis is warranted, as the observed value of 6.01 exceeds the critical value of 3.35, indicating substantial differences among the models. To delve deeper into these distinctions, we employed the Wilcoxon-signed-rank test to assess pairwise significance between the proposed edRVFL-FIS models and existing baseline models. The corresponding p-values are presented in Table 7.6.

Upon closer examination, it is evident that the proposed edRVFL-FIS-C model demonstrates a statistically significant difference compared to all baseline models. Similarly, the edRVFL-FIS-R and edRVFL-FIS-K models exhibit greater statistical significance with most baseline models. However, the edRVFL-FIS-R and edRVFL-FIS-K models do not show a significant difference when compared to edEGERVFL and edRVFL. Additionally, the edRVFL-FIS-K model also does not show a significant difference when compared to Fuzzy BLS and dRVFL. However, it is crucial to highlight that despite the absence of clear statistical differences in certain comparisons, all proposed edRVFL-FIS models consistently achieve the lowest rank. This underscores their superior generalization performance across

Table 7.7: Classification accuracies of the proposed edRVFL-FIS models and the baseline models on NDC datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	BLS [3]	H-ELM [206]	dRVFL [2]	edRVFL [2]	edRVFL-FIS-R†	edRVFL-FIS-K†	edRVFL-FIS-C†
NDC-10K	96.86	95.63	96.86	96.91	97	97.5	97.81	98.86	**99.03**
NDC-25K	94.84	94.56	95.91	97.32	96.35	97.52	97.52	**98.84**	98.42
NDC-50K	95.91	95.85	97.09	97.59	98.45	98.45	98.89	**99.91**	99.7
NDC-75K	95.69	94.86	97	96.98	97.89	97.93	97.93	98.56	**98.62**
NDC-100K	96.89	96.63	96.95	97.03	96.95	96.95	96.41	98.89	**98.95**
NDC-500K	95.89	95.89	96.83	95.97	96.83	97.03	98.47	**98.83**	98.32
NDC-1M	95.84	95.16	95.89	96.93	97.12	97.88	99	98.89	**99.3**
Average Accuracy	95.99	95.51	96.65	96.96	97.23	97.61	98	**98.97**	98.91
Average Rank	7.86	8.79	6.43	5.71	5.14	4.07	3.71	1.71	**1.57**

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

the evaluated models.

7.3.6 Experimental Results, Discussions, Statistical Analysis on NDC Dataset

Our experiments are conducted on big datasets generated using David Musicant's NDC Data Generator [204]. This experiment provides a comprehensive understanding of the performance of the proposed edRVFL-FIS models across a varied range of data samples, spanning from $10,000$ to 1 million. For example, NDC-10K, NDC-75K, and NDC-1M indicate that the dataset comprises $10,000$, $75,000$, and 1 million data samples, respectively. The results, presented in Table 7.7, provide insights into the models' performance on NDC datasets.

Upon analysis of Table 7.7, it's evident that the proposed edRVFL-FIS-K model excels as the top performer, achieving the highest average accuracy at 98.97%. Following suit, edRVFL-FIS-C and edRVFL-FIS-R secure the second and third positions with average accuracies of 98.91% and 98.00%, respectively. This consistent trend is reaffirmed by their average ranks, where edRVFL-FIS-K, edRVFL-FIS-C, and edRVFL-FIS-R outshine all other models with ranks of 1.71, 1.57, and 3.71, respectively. This compelling performance underscores the efficacy of our ensemble-based edRVFL-FIS models, particularly over extended feature layers. They adeptly leverage fuzzification and defuzzification fea-

tures to significantly enhance generalization performance, especially on large datasets such as NDC. This establishes the proposed edRVFL models' superiority over competing models and highlights the reliability of our proposed approach in handling substantial and complex data sets.

7.4 Discussion, Sensitivity Analysis and Erosion Experiment

In this section, firstly, we discuss some noteworthy remarks about the proposed models based on the above experiments and results. Then, we discuss the importance of hyper-parameter setting in the proposed edRVFL-FIS model in the following manner: sensitivity analysis based on the number of fuzzy rules and the number of hidden layer selections. Later, we perform the erosion experiment by introducing the Gaussian noise at different levels, 5%, 10%, 15%, and 20%, to corrupt the features of the datasets to assess the robustness of the proposed model.

7.4.1 Discussions Based on the Results on Category-1 Datasets

- The proposed edRVFL-FIS models excel in both average accuracy and average ranks on Category-1 datasets.

- Statistical analyses, including the Friedman and Wilcoxon-signed-rank tests, affirm the proposed models' statistical superiority over existing baseline models.

- The findings suggest that the proposed model can be preferred over the SOTA graph-based deep model, *i.e.,* edEGERVFL, due to its superior performance in accuracy, ranking, and standard deviation.

7.4.2 Discussions Based on the Results on Category-2 Datasets

- The proposed edRVFL-FIS models excel in extracting nuanced information from the augmented feature space, achieved through concatenating original and fuzzy features. This underscores the effectiveness of their fuzzification and defuzzification processes. In contrast, the diminished performance of Fuzzy BLS highlights its limitations in learning fuzzy rules compared to the proposed edRVFL-FIS models.

- The RVFL model always outperforms the RVFLwoDL (ELM) model in terms of average accuracy and average rank. This trend emphasizes the superior generalization performance of RVFL models over ELM and thus also shows the significance of a direct link.

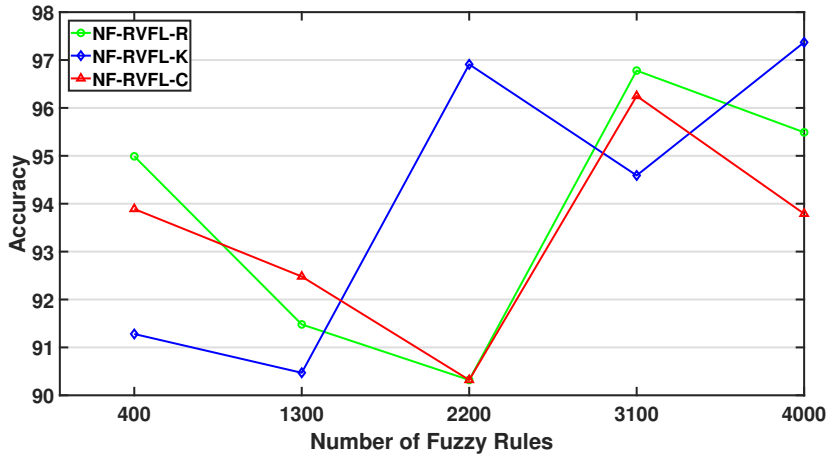
7.4.2.1 Number of Fuzzy Rules (Clusters) V/S Performance of edRVFL-FIS Models

To examine the impact of fuzzy rules (clusters) on the performance of the proposed edRVFL-FIS models with varying sample sizes in the datasets, we conduct experiments by varying the number of fuzzy rules (K) from $K = 400$ to $K = 4000$ with step size 900. The resulting accuracy of the edRVFL-FIS models is investigated and visualized in Figure 7.2 *w.r.t.* the NDC datasets of size 10K, 100K, and 1M.

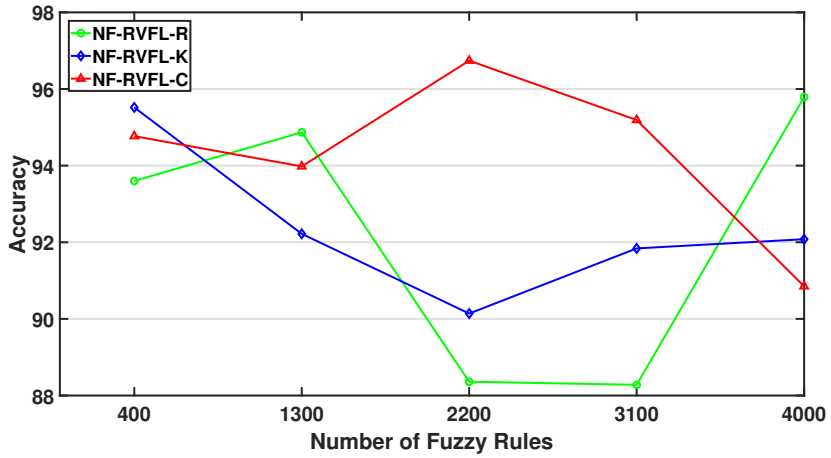
The noteworthy analyses are as follows:

[1] edRVFL-FIS-C: The accuracy of edRVFL-FIS-C shows a notable increase with the rise in fuzzy rules. Notably, tuning the fuzzy rules for edRVFL-FIS-C between 2200 and 3100 tends to yield optimal performance.

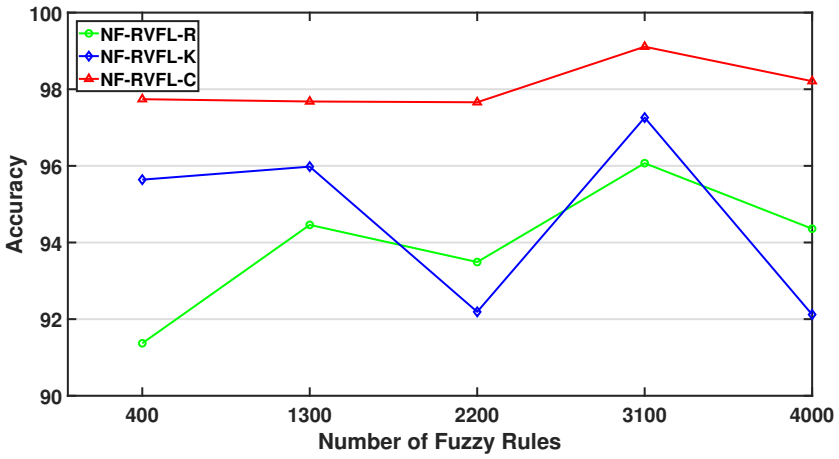
[2] edRVFL-FIS-R: The optimal performance range for edRVFL-FIS-R is generally observed at the higher values. Fine-tuning in the neighbourhood of 3100 and 4000 tends to yield the highest accuracy.



(a) NDC-10K

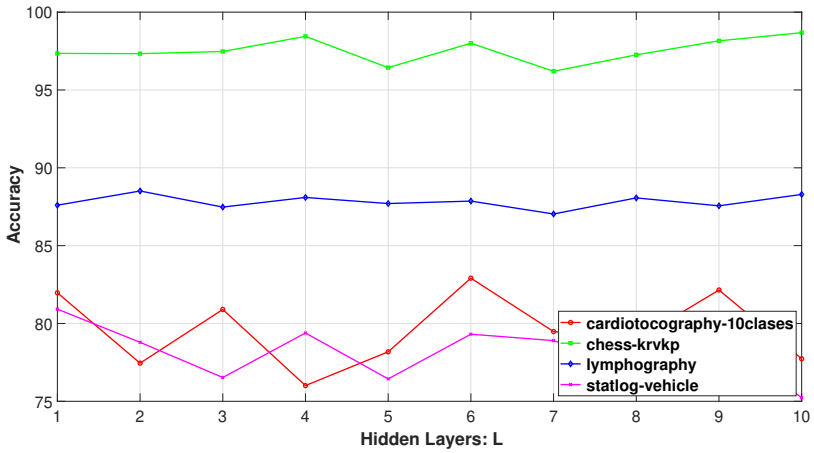


(b) NDC-100K

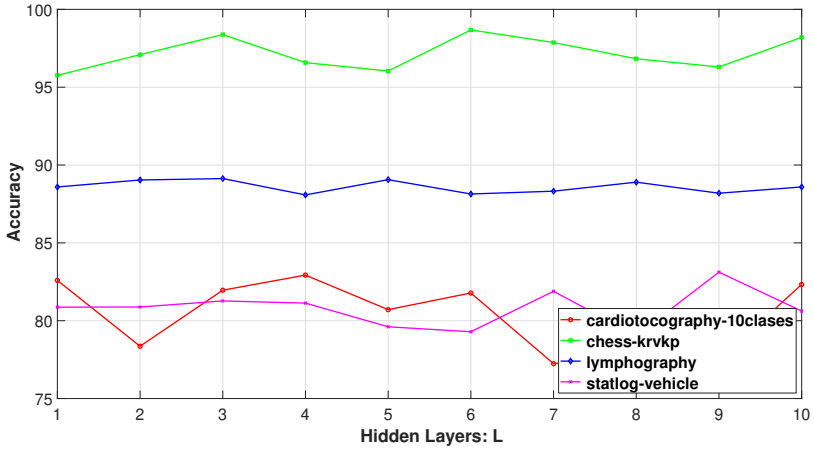


(c) NDC-1M

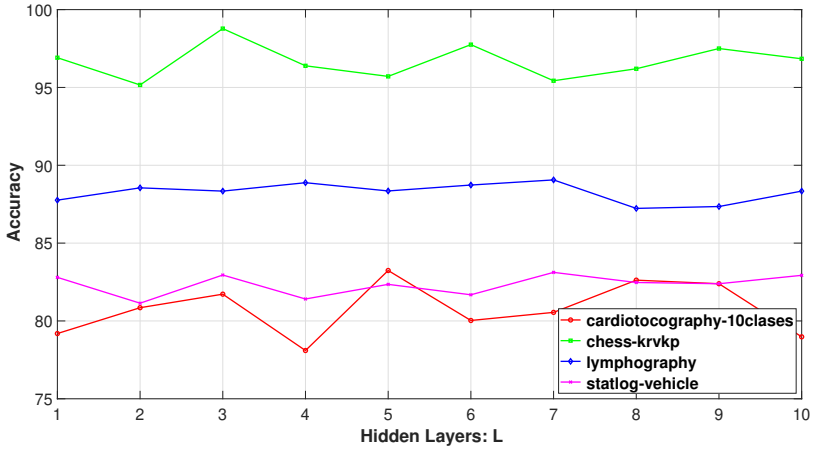
Figure 7.2: (a), (b), and (c) showcase the accuracy of the proposed edRVFL-FIS models on NDC-10K, NDC-100K, and NDC-1M datasets, respectively, *w.r.t.* the number of fuzzy rules.



(a) edRVFL-FIS-R



(b) edRVFL-FIS-K



(c) edRVFL-FIS-C

Figure 7.3: (a), (b), and (c) showcase the accuracy of the proposed edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C, respectively *w.r.t.* hidden layers.

[3] edRVFL-FIS-K: The performance pattern for edRVFL-FIS-K exhibits variations. For NDC-10K datasets, the optimal performance is at $K = 2200$; for NDC-100K, it peaks at $K = 400$; for NDC-1M, the best performance is at $K = 3100$. The mixed patterns in edRVFL-FIS-K highlight the dataset-dependent nature of optimal fuzzy rule selection.

Observations reveal discernible patterns in the performance of the proposed edRVFL-FIS-R and edRVFL-FIS-C models based on fuzzy rule variations. However, the performance varies across datasets and applications in accordance with the No Free Lunch Theorem [182]. Hence, a tailored approach to tuning fuzzy rules is recommended for optimizing the generalization performance of the proposed edRVFL-FIS models.

7.4.3 Number of Hidden Layers V/S Performance of edRVFL-FIS Models

To analyze how the number of hidden layers impacts the performance of our proposed edRVFL-FIS models, we conducted experiments where we varied the number of hidden layers (L). We then examined the resulting accuracy of these models concerning various UCI datasets. We selected a total of four datasets, evenly split between Category-1 and Category-2.

In Figure 7.3, we visualize the relationship between the number of hidden layers and accuracy for the edRVFL-FIS models. Specifically, Figures 7.3(a), 7.3(b), and 7.3(c) correspond to edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C, respectively. Upon examining these figures, it becomes evident that the performance of all models is highly dependent on the number of hidden layers. However, we did not observe a consistent pattern across all datasets. Therefore, we recommend that users tune the number of hidden layers to achieve an optimized solution.

Table 7.8: Performance of the proposed edRVFL-FIS models over NDC datasets with varying Gaussian noise

	Model	**edRVFL-FIS-R**	**edRVFL-FIS-K**	**edRVFL-FIS-C**
Dataset	**Noise** %	**Acc.**	**Acc.**	**Acc.**
NDC-10K	5	97.56	98.05	98.76
NDC-10K	10	97.35	97.52	97.23
NDC-10K	15	97.56	97.02	97.15
NDC-10K	20	96.43	96.89	97.09
NDC-25K	5	97.12	98.44	98.41
NDC-25K	10	96.97	98.23	98.15
NDC-25K	15	96.15	97.45	97.65
NDC-25K	20	95.87	97.23	97.23
NDC-50K	5	98.19	99.55	99.51
NDC-50K	10	97.85	98.58	98.96
NDC-50K	15	97.41	98.12	98.14
NDC-50K	20	96.86	97.64	97.59
NDC-75K	5	97.16	97.76	97.85
NDC-75K	10	96.78	97.15	97.25
NDC-75K	15	96.45	96.69	96.5
NDC-75K	20	96.13	96.01	96.11
NDC-100K	5	96.13	97.79	97.86
NDC-100K	10	96.01	96.84	97.43
NDC-100K	15	95.46	96.55	97.13
NDC-100K	20	94.81	96.52	97.25
NDC-500K	5	97.65	98.29	97.87
NDC-500K	10	96.74	96.88	97.24
NDC-500K	15	95.85	96.16	97.06
NDC-500K	20	95.19	95.75	96.69
NDC-1M	5	98.5	98.55	98.75
NDC-1M	10	97.2	97.77	97.14
NDC-1M	15	97.33	97.25	96.67
NDC-1M	20	96.77	96.93	96.27

7.5 Evaluation of the proposed models on UCI datasets with Gaussian Noise: Erosion experiments

In our research, we utilized real-world datasets that accurately represent various scenarios. However, it's important to acknowledge that data collected in real-world settings often contains impurities or noise due to various factors. In such situations, it becomes crucial to develop robust models capable of handling these challenges effectively. To demonstrate the efficacy of our proposed edRVFL-FIS models under challenging circumstances, we introduced Gaussian noise into NDC datasets. For a thorough analysis, we introduced Gaussian noise at different levels, 5%, 10%, 15%, and 20%, to corrupt the features of these datasets. This approach allows us to assess how well our models perform when faced with varying degrees of noise and highlight their superiority even in challenging scenarios.

Analysis: The accuracy of all proposed edRVFL-FIS models on the NDC datasets with 5%, 10%, 15%, and 20% noise levels are summarized in Table 7.8.

[1] The results reveal that despite increasing noise levels from 5% to 20%, the performance of the proposed edRVFL-FIS models remains relatively stable. For instance, in the NDC-10K datasets, the accuracy of the edRVFL-R models only decreases by 1.13% as noise levels rise from 5% to 20%. This resilience underscores the adaptability and robustness of our proposed models.

[2] Additionally, our edRVFL-FIS models exhibit excellent prediction stability, with a maximum accuracy reduction of only 2.5% observed for edRVFL-FIS-K on the NDC-500K dataset.

We subjected our models to challenging conditions and observed their ability to maintain effectiveness and stability in noisy environments. The results showcase the robustness and superior performance of the proposed edRVFL-FIS models due to the incorporation of FIS,

emphasizing their capability to excel even in adverse scenarios with high accuracy along with exhibiting minimal variation in predictions.

7.6 Summary

In this chapter, we proposed several extensions of deep and ensemble random vector functional link (RVFL) networks to enhance their expressive capability, interpretability, and robustness. First, we introduced the edRVFL-FIS model, which integrates a fuzzy inference system (FIS) within the ensemble deep RVFL framework to improve feature representation. The proposed model combines deep learning, ensemble learning, and fuzzy logic to generate an enhanced feature space consisting of original features, unsupervised fuzzified features, and supervised defuzzified features. To construct the fuzzy layer, different clustering techniques, R-means, K-means, and Fuzzy C-means, are employed, resulting in three variants: edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C. Extensive experiments conducted on UCI datasets demonstrate that the proposed edRVFL-FIS models effectively exploit fuzzy feature augmentation to improve predictive performance. Additional experiments on large-scale NDC datasets further confirm their scalability and strong generalization ability across datasets ranging from $10,000$ to 1 million samples.

Chapter 8

Robust Deep, Ensemble Deep and Broad Randomized Neural Networks

Outcome of this Chapter

1. Publication: M. Sajid, A. Quadir, A. Rahaman and M. Tanveer (2026). "Robust Ensemble Deep Randomized Neural Networks" [**Under Review**]

Source Code: https://github.com/mtanveer1/IF-edRVFL

2. Publication: M. Sajid, A. K. Malik and M. Tanveer (2024). "Intuitionistic Fuzzy Broad Learning System: Enhancing Robustness Against Noise and Outliers," in ***IEEE Transactions on Fuzzy Systems***, v. 32, no. 8, pp. 4460 - 4469, **DOI:** https://doi.org/10.1109/TFUZZ.2024.3400898 [**SCI Indexed with Impact Factor:** 10.2, Q1]

Source Code: https://github.com/mtanveer1/IF-BLS

The previous chapter enhances the expressiveness of ensemble deep RdNNs. However, deep, ensemble deep and broad RdNN frameworks, despite their effectiveness, treat all training samples as equally important, making them less robust when dealing with real-world datasets containing noise and outliers. Moreover, existing deep randomized neural networks treat all training samples uniformly, rendering them sensitive to noise and outliers and allowing corrupted representations to propagate across hidden layers. To overcome these limitations, this chapter proposes two complementary extensions of the edRVFL framework. First, we propose intuitionistic fuzzy-based deep and ensemble RVFL models, namely IF-dRVFL and IF-edRVFL, to improve robustness against noise and outliers. These models assign adaptive weights to training samples using intuitionistic fuzzy membership and non-membership measures derived from class centroid distances and neighborhood heterogeneity. Second, we propose two robust extensions: the fuzzy broad learning system (F-BLS) and the intuitionistic fuzzy broad learning system (IF-BLS). These models are specifically designed to mitigate the adverse effects of noise and outliers while enhancing overall robustness. In the F-BLS model, a fuzzy membership mechanism is employed to incorporate neighborhood information of samples based on their proximity to class centers within the intrinsic feature space of the BLS framework. Furthermore, the IF-BLS model extends this formulation by incorporating intuitionistic fuzzy concepts, including membership, non-membership, and score functions. It effectively captures both homogeneity and heterogeneity within sample neighborhoods in the kernel space, enabling more reliable weighting of training samples. The performance of the proposed IF-dRVFL, IF-edRVFL, F-BLS and IF-BLS models is evaluated on UCI benchmark datasets under both noise-free and Gaussian noise conditions. Additionally, Fuzzy BLSs are evaluated in a real-world application: the diagnosis of Alzheimer's disease (AD). Experimental results, supported by statistical analyses, consistently demonstrate the superior generalization performance of the proposed

methods compared to baseline models across all scenarios.

8.1 Background

In the previous chapter, we noticed that deep and ensemble RVFL models improve expressive capability, but their learning process can still be significantly affected by noisy or outlier-contaminated data. Noisy samples may propagate through successive hidden layers, contaminating higher-level representations and adversely affecting classification performance. Moreover, conventional dRVFL-based models typically assign equal importance to all training samples, disregarding the varying reliability of individual data points. This uniform treatment often leads to reduced robustness in challenging environments with significant noise or anomalous samples.

Fuzzy theory is successfully employed in machine learning models [145, 146] to ameliorate the adverse effects of noise or outliers on model performance. The fuzzy membership function uses sample-to-class center distance to generate a degree of membership to deal with outliers and noise. In [33], intuitionistic fuzzy (IF) score was presented as an advanced version of the fuzzy membership scheme. The IF theory uses the membership and non-membership functions to assign an IF score to each sample to deal with the noise and outliers efficiently. Membership and non-membership values of a sample measure the degree of belongingness and non-belongingness of that sample to a particular class, respectively.

Motivated by these insights, this chapter further introduces robust deep RVFL models based on intuitionistic fuzzy theory, namely IF-dRVFL and IF-edRVFL. In these models, each training sample is assigned an intuitionistic fuzzy score derived from its membership and non-membership values in the feature space. These scores are used to assign adaptive weights to training samples, enabling the network to distinguish between clean and noisy instances. As a result, the proposed models effectively mitigate the propagation of noisy

features through hidden layers and enhance robustness against noise and outliers.

Further, one of the prime advantages of deep learning is its capacity to automatically extract high-level features from raw data, eliminating the need for manual feature engineering. Deep learning also has the upper hand of being able to generalize effectively to new data, which allows it to make correct predictions even on data that has never been shown to it before. Apart from many advantages, there are some challenges of deep learning architectures: (a) Deep learning models consist of numerous layers and millions of parameters, making them extremely complicated. (b) The iterative learning process in deep learning models can be computationally intensive and time-consuming. (c) Training deep learning models often requires powerful hardware, such as GPUs or TPUs. A shortage of such hardware may hinder the training process. These drawbacks confine the applicability of deep learning models primarily to tasks such as speech and image recognition. Consequently, it gives space to other models to spread their wings.

Recently, the BLS [3] (inspired by RVFL), a class of flat neural networks, was proposed. BLS employs a layer-by-layer approach to extract informative features from the input data. BLS has three primary segments: a feature learning segment, an enhancement segment, and an output segment. In the feature learning and enhancement segment, the input data is transformed into a high-dimensional feature space using a set of random projection mappings. As a result of the feature learning and enhancement segment, a collection of high-dimensional features is produced that effectively extract important information from the input data. To train the BLS model, only the output layer parameters need to be computed by the least-squares method. In comparison to traditional deep neural networks (DNNs), BLS has several advantages. For example, it does not employ backpropagation, which might simplify the training process and lessen the likelihood of overfitting; its ability to learn from a small number of training samples; and faster training speed due to closed

form solution. The universal approximation capability of BLS [53] makes it more promising among researchers. The adaptable topology of BLS makes it possible to train and update the model effectively in an incremental way [3], and when training data is scarce, BLS may have superior generalization performance than deep learning models [54].

BLS has a number of benefits over conventional machine learning and deep learning models [77]. However, BLS is susceptible to noise and outliers. Outliers are data points that deviate greatly from the majority of the data points, whereas noise refers to random abnormalities or oscillations in the data [58]. The factors contributing to BLS's susceptibility to noise and outliers are as follows. (i) When the noisy features of a sample propagate through the feature and enhancement groups of the BLS, then all features (pure and impure) get mixed with each other. As a result, noisy features corrupt the higher-level features of enhancement groups, consequently skewing the learning process of the BLS. (ii) In the BLS model, every sample is assigned a uniform weight during training, regardless of the sample's purity or abnormality. Ideally, noisy samples should be given less weight than pure ones. However, BLS overlooks this distinction, resulting in difficulty discerning between pure and impure samples and consequently yielding poor generalization performance in such scenarios.

Overall, the models presented in this chapter collectively address three fundamental challenges in deep, ensemble deep, and broad randomized neural networks: robustness, representation capability, and practical applicability. The IF-dRVFL, IF-edRVFL, F-BLS, and IF-BLS models enhance robustness by incorporating intuitionistic fuzzy weighting mechanisms, enabling the networks to effectively mitigate the influence of noise, uncertainty, and outliers in the training data. Collectively, these approaches contribute to the development of randomized neural network architectures that are more robust, expressive, interpretable, and well-suited for real-world machine learning applications.

The key highlights of this chapter are summarized as follows:

[1] We propose robust randomized neural network models for both deep and broad architectures by integrating fuzzy and intuitionistic fuzzy learning paradigms. Specifically, intuitionistic fuzzy-based deep and ensemble RVFL models (IF-dRVFL and IF-edRVFL), along with fuzzy and intuitionistic fuzzy Broad Learning Systems (F-BLS and IF-BLS), are developed to enhance learning in the presence of uncertainty.

[2] The proposed models employ adaptive sample weighting strategies derived from fuzzy membership or intuitionistic fuzzy membership, non-membership, and score functions. These mechanisms effectively distinguish informative samples from noisy and outlier-contaminated samples by exploiting their proximity to class centers and local neighborhood characteristics in the kernel space, thereby significantly improving model robustness.

[3] Extensive experimental evaluations on benchmark datasets under both clean and noisy conditions demonstrate that the proposed models consistently outperform existing randomized neural network approaches in terms of robustness, feature representation capability, and predictive performance.

[4] The effectiveness of the proposed frameworks is further validated on Alzheimer's disease diagnosis, where they achieve superior classification performance compared to state-of-the-art baseline models, highlighting their practical applicability in real-world biomedical decision-making.

8.2 Proposed Methods: Intuitionistic Fuzzy-Based Deep and Ensemble Deep RVFLs

This section presents the detailed mathematical formulation of the proposed IF-(e)dRVFL models, followed by the description of the IF weighting scheme. Let $s_i = \Theta^{\star}(u_1)$ (as defind in Section 2.6.2) denote the IF score assigned to the training sample u_i, and let $\mathcal{S} = \text{diag}(s_1, s_2, \ldots, s_M)$ represent the corresponding diagonal IF weight matrix for the training set (see Section 2.6.2).

8.2.1 Proposed IF-dRVFL Network

In the standard dRVFL framework, all training samples are treated uniformly, irrespective of their quality or reliability. However, real-world datasets often contain noise and outliers, which can adversely affect the learning capability and generalization performance of dRVFL models. To mitigate the influence of such contaminated samples, we propose the IF-dRVFL network. In the proposed IF-dRVFL model, the fuzzy membership value of a sample is determined based on its proximity to the centroid of its corresponding class, reflecting the degree of belongingness. Conversely, the non-membership value is computed by incorporating neighborhood information, capturing the degree of non-belongingness of the sample. By jointly considering membership and non-membership information, the IF-dRVFL framework assigns adaptive weights to training samples, thereby enhancing robustness against noise and outliers.

Let $E^{(1)}$ represent the first hidden layer matrix, acquired through the projection of the input matrix using randomly initialized weights, followed by the activation function ψ as:

$$E^{(1)} = \psi(U\omega^{(1)}), \tag{8.1}$$

here, $\omega^{(1)} \in \mathbb{R}^{L\times h}$ represents the weights for the first hidden layer, initialized randomly from a uniform distribution of $[-1, 1]$ and h represents the number of hidden nodes.

The deeper hidden layers ($l > 1$) are defined as:

$$E^{(l)} = \psi(E^{(l-1)}\omega^{(l)}), \tag{8.2}$$

where $\omega^{(l)}$ represents the randomly generated weights for the lth hidden layer. The enhanced feature set is defined by concatenating all the original and hidden layer features:

$$E^* = [E^{(\mathbf{L})}, E^{(\mathbf{L}-1)}, \ldots, E^{(1)}, U]. \tag{8.3}$$

Now, we propose the optimization problem of IF-dRVFL as:

$$\begin{aligned} &\min_{\xi} \ \frac{\mathcal{C}}{2}\|\mathcal{S}\eta\|^2 + \frac{1}{2}\|\xi\|^2 \\ &s.t. \ \ E^*\xi - T = \eta, \end{aligned} \tag{8.4}$$

where $\mathcal{C}$ is a regularization parameter and ξ is the output layer weights (unknown) and needs to be calculated. The Lagrangian corresponding to the problem (8.4) is formulated as:

$$L = \frac{\mathcal{C}}{2}\|\mathcal{S}(E^*\xi - T)\|^2 + \frac{1}{2}\|\xi\|^2. \tag{8.5}$$

Now, we differentiate L with respect to ξ and then equate it to zero, we obtain:

$$\frac{\partial L}{\partial \xi} = \xi + \mathcal{C}(\mathcal{S}E^*)^T(\mathcal{S}(E^*\xi - T)) = 0. \tag{8.6}$$

Solving (8.6), we get

$$\begin{aligned}\xi &= \left((\mathcal{S}E^*)^T(\mathcal{S}E^*) + \frac{1}{\mathcal{C}}I\right)^{-1}(\mathcal{S}E^*)^T\mathcal{S}T \\ &= \left(E^{*T}\mathcal{S}^2E^* + \frac{1}{\mathcal{C}}I\right)^{-1}E^{*T}\mathcal{S}^2T. \end{aligned} \tag{8.7}$$

where I is the identity matrix of the appropriate dimension.

Substituting the value $\xi = (\mathcal{S}E^*)^T\eta$ [207], we get

$$(\mathcal{S}E^*)\eta + \mathcal{C}(\mathcal{S}E^*)^T(\mathcal{S}(E^*(\mathcal{S}E^*)^T\eta - T)) = 0. \tag{8.8}$$

After simplifying (8.8), we get:

$$\begin{aligned}\eta &= \left(\mathcal{S}E^*E^{*T}\mathcal{S} + \frac{1}{\mathcal{C}}I\right)^{-1}\mathcal{S}T, \\ &= \mathcal{S}^{-1}\left(E^*E^{*T} + \frac{1}{\mathcal{C}}\mathcal{S}^{-1}\mathcal{S}^{-1}\right)^{-1}T, \\ &= \mathcal{S}^{-1}\left(\mathcal{S}^2E^*E^{*T} + \frac{1}{\mathcal{C}}I\right)^{-1}\mathcal{S}^2T. \end{aligned} \tag{8.9}$$

Finally,

$$\xi = (\mathcal{S}E^*)^T\eta = E^{*T}\left(\mathcal{S}^2E^*E^{*T} + \frac{1}{\mathcal{C}}I\right)^{-1}\mathcal{S}^2T. \tag{8.10}$$

Thus, we can provide the optimal solution of (8.4) as follows:

$$\xi = \begin{cases} \left(E^{*T}\mathcal{S}^2E^* + \frac{1}{\mathcal{C}}I\right)^{-1}E^{*T}\mathcal{S}^2T, & (L + \mathbf{L}h) \leq M, \\ E^{*T}\left(\mathcal{S}^2E^*E^{*T} + \frac{1}{\mathcal{C}}I\right)^{-1}\mathcal{S}^2T, & M < (L + \mathbf{L}h). \end{cases} \tag{8.11}$$

The flowchart of the proposed IF-dRVFL is given in Fig. 8.1 (a).

8.2.2 Proposed IF-edRVFL Network

In the proposed IF-edRVFL, each layer is treated as a base model, thus unifying the ensemble learning principle and intuitionistic fuzzy scheme into deep RVFL. This facilitates the creation of multiple diverse and robust base models (RVFLs) within a single deep framework. The final outcome of IF-edRVFL is determined through either averaging or a majority voting scheme. Within IF-edRVFL, every base model (hidden layer) receives inputs comprising the original features and the randomized features computed by the preceding layer. Furthermore, the optimization problem of each base model possesses the IF score weights. Thus, each base model behaves as a robust base model and enriches the learning process.

The first hidden layer matrix $E^{(1)}$ is defined as in (8.1) and the deeper hidden layers' ($l > 1$) output is defined as:

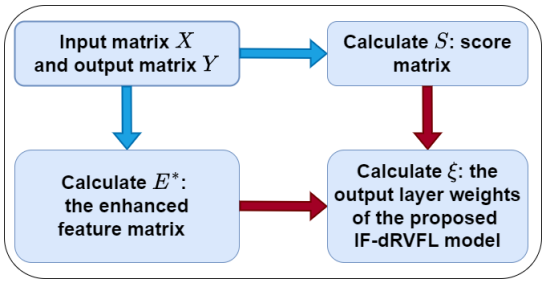
$$E^{(l)} = \psi([E^{(l-1)}, U]\omega^{(l)}), \quad l = 1, 2, \ldots, \mathbf{L}, \tag{8.12}$$

where $\omega^{(1)}$ and $\omega^{(l)}$ ($l > 1$) represent the randomly generated weights for the first and l^{th} ($l > 1$) hidden layers, respectively. The feature set corresponding to the first base model and the lth base model is defined as follows:

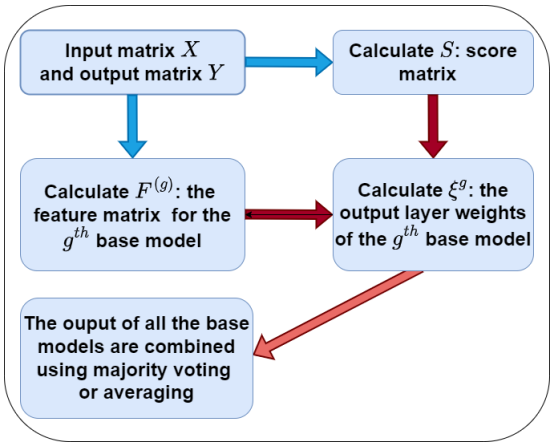
$$F^{(1)} = [E^{(1)}, U] \quad \text{and} \quad F^{(l)} = [E^{(l)}, E^{(l-1)}, U].$$

The proposed optimization problem corresponding to the lth base model is given as follows:

$$\begin{aligned} &\underset{\xi^l}{min} \ \frac{\mathcal{C}}{2}\|\mathcal{S}\eta^l\|^2 + \frac{1}{2}\|\xi^l\|^2 \\ &s.t. \ F^{(l)}\xi^l - T = \eta^l, \end{aligned} \tag{8.13}$$



(a) IF-dRVFL



(b) IF-edRVFL

Figure 8.1: Flowchart of the proposed models.

Table 8.1: Average accuracy and standard deviation with SOTA non-fuzzy RdNNs on 13 KEEL datasets.

Dataset ↓ | Model →	RVFL [1]	RVFLwoDL [22]	BLS [3]	H-ELM [206]	dRVFL [2]	edRVFL [2]	edEGEFRVFL [205]	IF-dRVFL (ours)	IF-edRVFL (ours)
abalone9-18	95.49	94.21	95.18	95.36	96.04	95.63	95.77	96.04	96.04
bupa_or_liver-disorders	71.88	71.88	74.49	73.33	73.62	73.62	73.04	74.2	73.91
cmc	64.58	63.22	65.66	63.22	63.08	63.42	63.56	63.97	64.13
crossplane130	39.23	38.21	39.23	39.23	52.31	52.31	52.31	60.77	60.77
crossplane150	62	62	62	62	62.67	60	62	64.67	67.33
ecoli-0-1-4-7_vs_2-3-5-6	97.31	97.31	97.21	97.31	97.01	97.31	95.83	97.02	97.32
ecoli-0-1-4-7_vs_5-6	97.89	96.81	98.8	97.29	97.89	97.89	97.89	98.5	98.8
ecoli-0-6-7_vs_5	97.27	97.27	98.18	95.91	97.27	96.82	97.27	98.18	97.27
glass2	92.05	92.05	92.05	92.05	92.05	92.05	92.05	92.05	92.05
glass4	95.77	95.77	97.07	95.78	97.65	98.12	97.65	97.67	96.26
glass5	96.72	96.72	98.13	95.79	98.59	98.13	97.65	98.14	98.59
haberman	74.15	74.15	74.15	74.15	73.82	73.49	73.82	74.15	73.82
heart-stat	80	80	80.96	80	80.74	80.74	81.11	80.74	80.74
Average Accuracy	81.87	81.51	82.55	81.65	83.29	83.04	83.07	**84.32**	**84.39**
Average Standard Deviation	5.1	5.13	4.46	5.07	4.99	5.19	5.14	**6.22**	**5.64**

where ξ^l is the output layer weight of the lth base model. The solution of (8.13) follows the same procedure as solved (8.4) of the proposed IF-dRVFL model. The output of all the base models is combined using majority voting to make the final decision of the proposed IF-edRVFL model. The flowchart of the proposed IF-edRVFL is given in Fig. 8.1 (b).

8.2.3 Experimental Results

To assess the effectiveness of the proposed IF-dRVFL and IF-edRVFL models, we perform a comparative analysis with baseline models using benchmark datasets from the UCI [7] and KEEL [8] repository.

8.2.3.1 Experimental Setup

Experimental Software: The experimental setup consists of a PC equipped with an Intel(R) Xeon(R) Gold 6226R CPU operating at a speed of 2.90GHz and featuring 128 GB of RAM. This system runs on the Windows 11 platform and the experiments are executed using MATLAB R2023a.

Experimental Procedure: Grid search with a 5-fold cross-validation technique is used to optimize the model's hyperparameters. In order to accomplish this, the dataset was divided into five distinct, non-overlapping subsets, or "folds". One subset was set to be designated for testing, and the other four were used for training in every phase. For every fold, testing accuracy was separately calculated using different hyperparameter sets. The average testing accuracy was determined by taking the mean of these five accuracies for every combination of hyperparameters. The models' testing accuracy is determined by selecting the model with the highest average testing accuracy.

Hyperparameter Selection: In the proposed IF-dRVFL and IF-edRVFL models, intuitionistic fuzzy weights are generated by transforming the samples into a higher-dimensional space through the use of a kernel function. We used the Gaussian kernel and is given by $K(u_i, u_j) = e^{\frac{-1}{2\sigma^2}\|u_i - u_j\|^2}$. Gaussian kernel parameter σ is selected from the range $\{2^{-5}, 2^{-4}, \ldots, 2^5\}$. The regularization parameters for each model are selected from the set $\mathcal{C} = \{10^{-5}, 10^{-4}, \ldots, 10^{-5}\}$. Following the methodology outlined in [206], the selection of the number of hidden layer nodes "h" in the RVFLwoDL, RVFL, and H-ELM follows the range $h = 100 : 100 : 2000$. Following [205], the graph regularization parameter for the edEGERVFL model is set equal to $\mathcal{C}$. Following [2], a two-stage tuning approach was employed for the dRVFL, edRVFL, edEGERVFL, edRVFL-FIS-C, IF-dRVFL, and IF-edRVFL models. In the first stage, the parameters are fine-tuned to obtain the optimal hidden nodes h^* within the range of $[256, 512, 1024]$, and the regularization parameter $\mathcal{C}^*$ is determined

while keeping the number of hidden layers fixed at two. For IF-dRVFL and IF-edRVFL, $\sigma = 1$ is fixed, and σ is fine-tuned in the second stage. During the second stage, hidden layers within the range of $1:1:10$ are tuned, along with other parameters, in the vicinity of g^* and $\mathcal{C}^*$. Furthermore, for the proposed edRVFL-FIS models, the best number of fuzzy nodes K^* in the first stage is tuned within the range of $5:10:45$, and in the subsequent stage, they are fine-tuned in the vicinity of K^*, specifically within the range of $K^*-4:1:K^*+5$. Feature groups for BLS and F-BLS are selected from the range $N_2 = 1:2:21$, while the selection of feature nodes within each feature group is made from the range $N_1 = 5:5:50$. The NF-BLS model selects the number of fuzzy groups from the range $N_{fg} = 1:2:21$ and the number of fuzzy nodes in each fuzzy group from the range $N_{fn} = 5:5:50$. The number of enhancement nodes for BLS, F-BLS and F-BLS is chosen from the range $N_3 = 5:5:50$.

8.2.3.2 Models in Comparison

The performance of the proposed IF-dRVFL and IF-edRVFL models is compared against 11 baselines with two kinds of models: Firstly, 7 SOTA non-fuzzy RdNNs, including RVFL [1], RVFL without direct link (RVFLwoDL) -also known as extreme learning machine (ELM) [22], broad learning system (BLS) [3], hierarchical ELM (H-ELM) [206], dRVFL [2], edRVFL [2], ensemble deep extended graph embedded RVFL (edEGERVFL) [205]. Secondly, 4 SOTA fuzzy-based baselines such as neuro-fuzzy BLS (NF-BLS) [56], Fuzzy BLS [78], IF twin support vector machine (IF-TSVM) [34] and edRVFL based on fuzzy inference system (edRVFL-FIS-C) [66].

8.2.3.3 Experimental Results and Statistical Analysis on Real-World Datasets Against SOTA Non-Fuzzy RdNNs

In this subsection, we evaluate the proposed IF-dRVFL and IF-edRVFL models against state-of-the-art (SOTA) non-fuzzy randomized deep neural network (RdNN) baselines on 13 KEEL benchmark datasets spanning diverse domains and sample sizes.

Accuracy (ACC): The accuracy in each dataset, average classification accuracy (ACC), standard deviation (Std.), and corresponding ranks are summarized in Table 8.1. It is evident that the proposed IF-edRVFL and IF-dRVFL models achieve the highest and second-highest average ACC values of 84.39% and 84.32%, respectively. In comparison, the baseline RVFL, RVFLwoDL, BLS, H-ELM, dRVFL, edRVFL, NF-BLS, and edEGEFRVFL models attain average ACC values of 81.87%, 80.51%, 82.55%, 81.65%, 8329%, 83.04%, and 81.07%, respectively. These results clearly demonstrate the superior and more robust performance of the proposed intuitionistic fuzzy models across most datasets.

Statistical ranking: Since average accuracy may be influenced by exceptionally strong performance on individual datasets, we further assess the models using a rank-based evaluation. The ranking analysis suggests that the proposed IF-dRVFL and IF-edRVFL models obtain the lowest average ranks of 3.12 and 3.23, respectively, outperforming all competing methods. This confirms their superior generalization ability over diverse real-world datasets.

Friedman test: To statistically validate the observed performance differences, we apply the Friedman test [153]. For $q = 9$ classifiers evaluated on $P = 13$ datasets, the obtained Friedman statistic and its corresponding F-value are $\chi_F^2 = 25.78$ and $F_F = 3.96$, respectively. Since F_F exceeds the critical value $F_F(8, 96) = 2.04$ at the 5% significance level, the null hypothesis of equal performance among the models is rejected, indicating statistically significant differences.

Table 8.2: Average accuracy, standard deviation and rank comparison with SOTA fuzzy baselines on 12 UCI datasets.

Dataset ↓ | Model →	NF-BLS [56]	F-BLS [78]	IF-TSVM [34]	edRVFL-FIS-C [66]	IF-dRVFL (ours)	IF-edRVFL (ours)
acute_nephritis	100	100	95	100	100	100
blood	77.57	75.7	76.24	76.77	76.37	76.37
breast_cancer	70.18	72.29	70.18	70.18	70.53	70.18
echocardiogram	81.62	84.67	80.09	86.21	84.67	86.21
haberman_survival	73.49	69.66	73.49	73.49	73.49	73.49
heart_hungarian	68.99	78.94	64.41	71.36	76.87	78.89
hepatitis	88.39	84.52	79.35	86.45	87.1	87.87
molec_biol_promoter	65.11	84.94	55.71	64.33	86.19	89.83
pittsburg_bridges_T_OR_D	89.14	88.19	86.14	89.14	86.14	86.14
spect	68.68	69.43	58.49	67.92	70.57	71.55
tic_tac_toe	82.45	97.81	65.31	99.37	99.58	99.68
titanic	77.33	77.92	73.92	77.1	78.28	77.92
Average Accuracy	78.58	82	73.19	80.19	**82.48**	**83.18**
Average Standard Dev.	12.61	8.98	25.47	12.16	**11.76**	**12.09**
Average Rank	3.42	3.42	5.46	3.46	**2.75**	**2.46**

Nemenyi post hoc test: Following the Friedman test, we conduct the Nemenyi post hoc analysis to examine pairwise performance differences among the classifiers. At a 5% significance level, the obtained critical difference ($C.D.$) value is 3.33. The rank differences demonstrate that both IF-dRVFL and IF-edRVFL are statistically superior to several baseline models, including RVFLwoDL and H-ELM. Overall, the consistently lower average ranks achieved by the proposed methods confirm their statistically significant superiority over existing non-fuzzy RdNN baselines.

8.2.3.4 Experimental Results and Statistical Analysis on Real-World Datasets Against SOTA Fuzzy Baselines

The results in Table 8.2 clearly demonstrate the superior performance of the proposed IF-dRVFL and IF-edRVFL models over SOTA fuzzy baselines on 12 UCI datasets. IF-edRVFL achieves the highest average accuracy of 83.18%, followed by IF-dRVFL with 82.48%, consistently outperforming NF-BLS, F-BLS, IF-TSVM, and edRVFL-FIS-C. This improvement highlights the effectiveness of incorporating intuitionistic fuzzy modeling within deep randomized neural networks for handling uncertainty and data ambiguity.

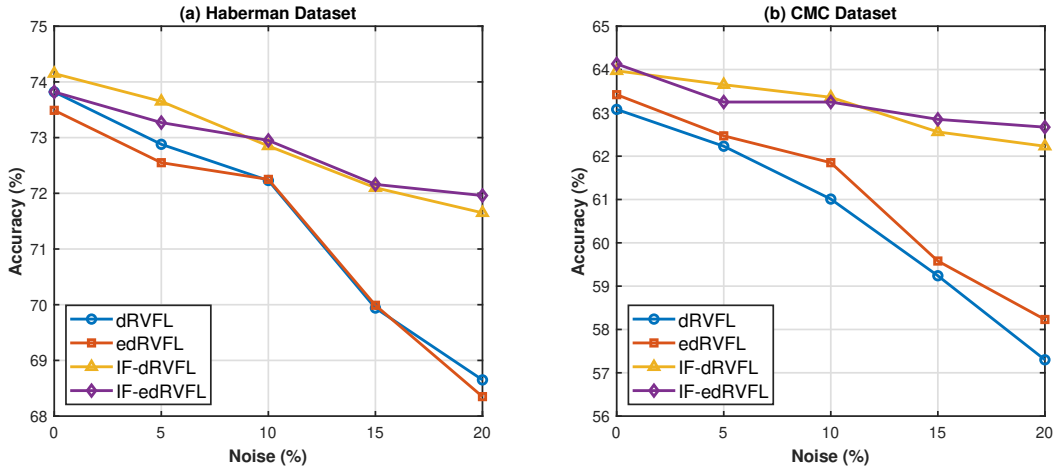
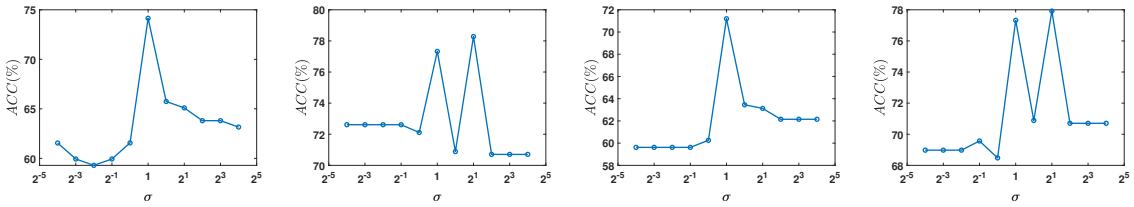


Figure 8.2: Robustness: Effect of different noise levels on the performance of dRVFL, edRVFL, IF-dRVFL (proposed) and IF-edRVFL (proposed) models.

From the statistical analysis, IF-edRVFL and IF-dRVFL obtain the lowest average ranks of 2.46 and 2.75, respectively, indicating superior generalization and stability across datasets. The Friedman test ($\chi_F^2 = 18.14$, $F_F = 4.77$) rejects the null hypothesis at the 5% significance level, confirming statistically significant differences among the models. Furthermore, the Nemenyi post hoc test ($C.D. = 2.17$) verifies that the proposed models significantly outperform IF-TSVM. Overall, the proposed intuitionistic fuzzy deep RVFL frameworks deliver higher accuracy, improved robustness, and statistically superior performance compared to existing fuzzy baselines.

8.2.3.5 Robustness Evaluation of Proposed Models on Datasets with Gaussian Noise

To assess the robustness of the proposed IF-dRVFL and IF-edRVFL models in comparison to the baseline models, namely dRVFL and edRVFL. We deliberately introduce Gaussian noise to the "haberman" and "cmc" datasets at varying levels of 5%, 10%, 15%, and 20%. The results illustrated in Fig. 8.2 reveal a clear robustness advantage of the proposed IF-dRVFL and IF-edRVFL models under increasing noise levels for both the Haberman and CMC datasets. As the noise percentage increases from 0% to 20%, the performance degra-



(a) haberman (IF-dRVFL) (b) titanic (IF-dRVFL) (c) haberman (IF-edRVFL) (d) titanic (IF-edRVFL)

Figure 8.3: Effect of parameter σ on the performance of the proposed IF-dRVFL and IF-edRVFL models.

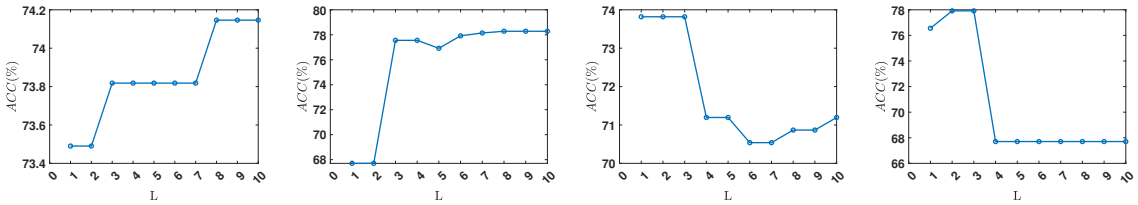
dation of the e(d)RVFL models is pronounced, with a sharp decline in accuracy, particularly beyond 10% noise. In contrast, the proposed intuitionistic fuzzy models exhibit a significantly slower and more controlled reduction in accuracy, maintaining consistently higher performance across all noise levels.

For the Haberman dataset, IF-dRVFL and IF-edRVFL demonstrate remarkable stability, with accuracy reductions limited to approximately 2–3% at 20% noise, whereas dRVFL and edRVFL suffer drops exceeding 5%. A similar trend is observed for the CMC dataset, where the baseline models experience steep performance deterioration under noise, while the proposed models preserve higher accuracy margins throughout. Overall, these findings confirm that embedding intuitionistic fuzzy modeling within deep RVFL architectures substantially enhances robustness to noise, leading to more reliable and stable predictions.

8.2.3.6 Sensitivity Analyses

We conduct the sensitivity analyses to delve deeper into the behaviour of the models: (i) by investigating the impact of intuitionistic fuzzy Gaussian kernel parameter σ, (ii) and by exploring the impact of the number of hidden layers L.

(1) Influence of intuitionistic fuzzy Gaussian kernel parameter σ: The impact of σ on the performance of the proposed models is shown in Fig. 8.3. Specifically, Fig. 8.3 (a) and (b) pertain to the IF-dRVFL models on the haberman and titanic datasets, respectively,



(a) haberman (IF-dRVFL) (b) titanic (IF-dRVFL) (c) haberman (IF-edRVFL) (d) titanic (IF-edRVFL)

Figure 8.4: Effect of parameter $\mathbf{L}$ on the performance of the proposed IF-dRVFL and IF-edRVFL models.

while Fig. 8.3 (c) and (d) focus on the IF-edRVFL models on the same datasets. We note that the performance of all proposed models improves as σ increases from 2^{-5}. Optimal performance is often achieved when σ is set to $2^0 = 1$ or $2^1 = 2$. Subsequently, as σ further increases, performance begins to diminish. Therefore, we recommend using $\sigma = 1$ or 2 for optimal results, although fine-tuning may be necessary depending on the dataset's characteristics.

(2) Influence of the number of hidden layers $\mathbf{L}$***:*** The impact of the hyperparameter $\mathbf{L}$ is shown in Fig. 8.4. Our analysis reveals the following: (a) For the proposed IF-dRVFL model, performance consistently improves with an increase in the number of hidden layers until reaching a plateau. Optimal performance is typically achieved with higher values of $\mathbf{L}$, such as 8 or larger. (b) In the case of the proposed IF-edRVFL model, we observe that performance peaks at $\mathbf{L} = 3$ and then gradually declines as $\mathbf{L}$ increases further. Therefore, to achieve the best performance from the IF-edRVFL model, we recommend using $\mathbf{L} = 3$. However, we recommend fine-tuning the hyperparameters to achieve the best performance for the proposed models for specific tasks.

8.3 Proposed Method Fuzzy and Intuitionistic Fuzzy BLS

In this section, we go through the architecture of IF-BLS along with its mathematical formulation.

8.3.1 Proposed F-BLS and IF-BLS

Let $\{(X,T) \mid X \in \mathbb{R}^{M\times D},\, T \in \mathbb{R}^{M\times C}\}$ be the training set, where X is the input matrix and T is the target matrix. Here, N denotes the total number of training samples, each of which has dimension D and C number of classes. In BLS, each data sample is given the same weight irrespective of its nature. Naturally, a dataset is never pure, $i.e.$, the involvement of noise and outliers in a dataset is a normal phenomenon. Although the occurrence of these noises and outliers is natural, it has a detrimental impact on the traditional BLS. Therefore, to deal with noisy samples and outliers present in the dataset, we propose two models:

[1] Fuzzy BLS (F-BLS) by employing the fuzzy membership scheme discussed in the subsection 2.6.1.

[2] Intuitionistic fuzzy BLS (IF-BLS) using the IF scheme discussed in 2.6.2.

The fuzzy membership value in the proposed F-BLS and IF-BLS models is based on how close a sample is to the corresponding class center in the original feature space and kernel space, respectively. These fuzzy membership values represent the degree to which a sample belongs to a certain class. In IF-BLS, the fuzzy non-membership value is determined by considering the sample's neighborhood information, indicating its extent of non-belongingness to a specific class.

For the broad matrix G (see Section 2.5.1) defined from the input matrix U, we define

the optimization problem of the proposed F-BLS and IF-BLS models as:

$$W_{T_{\min}} = \underset{W_T}{\operatorname{argmin}} \frac{\mathcal{C}}{2}\|\mathcal{S}\zeta\|_2^2 + \frac{1}{2}\|W_T\|_2^2,$$
$$\text{s.t. } GW_T - T = \zeta, \tag{8.14}$$

where $\mathcal{S}$ denotes the fuzzy or intuitionistic fuzzy score matrix (see Section 2.6). For F-BLS, the score matrix is obtained from the Subsection 2.6.1, whereas for IF-BLS, $\mathcal{S}$ is taken from the Subsection 2.6.2. Here, ζ refers to the error matrix. Equivalently, the formulation (8.14) can be written as:

$$W_{T_{\min}} = \underset{W_T}{\operatorname{argmin}} \frac{\mathcal{C}}{2}\|\mathcal{S}(GW_T - T)\|_2^2 + \frac{1}{2}\|W_T\|_2^2. \tag{8.15}$$

Problem (8.15) is the convex quadratic programming problem (QPP) and hence possesses a unique solution. The Lagrangian of (8.14) is written as:

$$\mathcal{L}(W_T, \zeta, \lambda) = \frac{\mathcal{C}}{2}\|\mathcal{S}\zeta\|_2^2 + \frac{1}{2}\|W_T\|_2^2 - \lambda^t(GW_T - T - \zeta), \tag{8.16}$$

where λ is the Lagrangian multiplier and $(\cdot)^t$ is the transpose operator. Differentiating $\mathcal{L}$ partially w.r.t. each parameters, $i.e.$, W_T, ζ and λ; and equating them to zero, we obtain

$$\frac{\partial \mathcal{L}}{\partial W_T} = 0 \Rightarrow W_T - G^t\lambda = 0 \Rightarrow W_T = G^t\lambda, \tag{8.17}$$
$$\frac{\partial \mathcal{L}}{\partial \zeta} = 0 \Rightarrow C\mathcal{S}^t(\mathcal{S}\zeta) + \lambda = 0 \Rightarrow \lambda = -C\mathcal{S}^t\mathcal{S}\zeta, \tag{8.18}$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \Rightarrow GW_T - T - \zeta = 0 \Rightarrow \zeta = GW_T - T. \tag{8.19}$$

Substituting Eq. (8.19) in (8.18), we get

$$\lambda = -C\mathcal{S}^t\mathcal{S}(GW_T - T). \tag{8.20}$$

On substituting the value of λ obtained in Eq. (8.17), we obtain

$$W_T = G^t \left(-C\mathcal{S}^t\mathcal{S}(GW_T - T)\right), \tag{8.21}$$

$$\Rightarrow W_T = -CG^t\mathcal{S}^2GW_T + CG^t\mathcal{S}^2T, \tag{8.22}$$

$$\Rightarrow \left(I + CG^t\mathcal{S}^2G\right)W_T = CG^t\mathcal{S}^2T, \tag{8.23}$$

$$\Rightarrow W_T = \left(G^t\mathcal{S}^2G + \frac{1}{\mathcal{C}}I\right)^{-1} G^t\mathcal{S}^2T, \tag{8.24}$$

where I is the identity matrix of the appropriate dimension. Substituting the values of (8.19) and (8.17) in (8.18), we get

$$\lambda = -\mathcal{C}\mathcal{S}^t\mathcal{S}(GG^t\lambda - T), \tag{8.25}$$

$$\Rightarrow \lambda + \mathcal{C}\mathcal{S}^2GG^t\lambda = C\mathcal{S}^2T, \tag{8.26}$$

$$\Rightarrow C\left(\frac{1}{\mathcal{C}}I + \mathcal{S}^2GG^t\right)\lambda = C\mathcal{S}^2T, \tag{8.27}$$

$$\Rightarrow \lambda = \left(\frac{1}{\mathcal{C}}I + \mathcal{S}^2GG^t\right)^{-1}\mathcal{S}^2T. \tag{8.28}$$

Putting the value of λ obtained in Eq. (8.28) in (8.17), we get

$$W_T = G^t\left(\frac{1}{\mathcal{C}}I + \mathcal{S}^2GG^t\right)^{-1}\mathcal{S}^2T. \tag{8.29}$$

We get two distinct formulas, (8.24) and (8.29), that can be utilized to determine W_T. It is worth noting that both formulas involve the calculation of the matrix inverse. If the number of features $(mp+lq)$ in G is less than or equal to the number of samples (M), we employ the formula (8.24) to compute W_T. Otherwise, we opt for the formula (8.29) to calculate W_T. As a result, we possess the advantage of calculating the matrix inverse either in the feature or sample space, contingent upon the specific scenario. Therefore, the optimal solution of

(8.14) is given as:

$$W_T = \begin{cases} \left(G^t\mathcal{S}^2G + \frac{1}{\mathcal{C}}I\right)^{-1} G^t\mathcal{S}^2T, & \text{if } (mp+lq) \leq M, \\ G^t\left(\frac{1}{\mathcal{C}}I + \mathcal{S}^2GG^t\right)^{-1}\mathcal{S}^2T, & \text{if } M < (mp+lq). \end{cases} \tag{8.30}$$

Remarks: The Gaussian kernel function is employed to project the input samples into a higher-dimensional space. The Gaussian kernel is defined as: $K(u_1, u_2) = exp(-\frac{\|u_1-u_2\|^2}{\mu^2})$, where μ is the kernel parameter.

The flowchart presented in Figure 8.5 illustrates the operational steps outlined for the Algorithm 8.5 of the proposed models.

Algorithm 8.5 Algorithm of the proposed F-BLS and IF-BLS models.

1: **Input:** Dataset of M sample, L features and C classes.
2: **Input:** Input matrix $U \in \mathbb{R}^{M\times L}$ and output matrix $T \in \mathbb{R}^{M\times C}$.
3: **Parameters:** $\mathcal{C}, \mu, m, p, l$ and q represents regularization parameter, kernel parameter, #feature groups, #feature nodes, #enhancement groups and #enhancement nodes, respectively.
4: **Input Sample:** Let u_r be the rth training sample, where $r = 1, 2, \ldots, M$.
5: **Find**: $F_i \in \mathbb{R}^{M\times p}, \ \ i = 1, 2, \ldots, m$, using Eq. (2.17).
6: **Find**: $F^m \in \mathbb{R}^{M\times mp}$ using Eq. (2.18).
7: **Find**: $E_j \in \mathbb{R}^{M\times q}, \ \ j = 1, 2, \ldots, l$, using Eq. (2.19).
8: **Find**: $E^l \in \mathbb{R}^{M\times lq}$ using Eq. (2.20).
9: **Find**: $G \in \mathbb{R}^{M\times(mp+lq)}$ using Eq. (2.22).
10: **If Model=F-BLS, follow steps (11) - (12), else Model=IF-BLS, follow steps (13) - (16).**
11: **Find**: Membership value $\vartheta(u_r)$ for u_r using Eq. (2.23).
12: **Find**: Score matrix $\mathcal{S}$ using Eq. (2.26).
13: **Find**: Membership value $\theta(u_r)$ for u_r using Eq. (2.27).
14: **Find**: Non-membership value $\Theta(u_r)$ for u_r using Eq. (2.28).
15: **Find**: Intutionistic fuzzy score value $\Theta^\star(u_r)$ using (2.29).
16: **Find**: Score matrix $\mathcal{S}$ using Eq. (2.30).
17: **Output**: Output weight matrix W_T using Eq. (8.30).

Source code link: https://github.com/mtanveer1/IF-BLS.

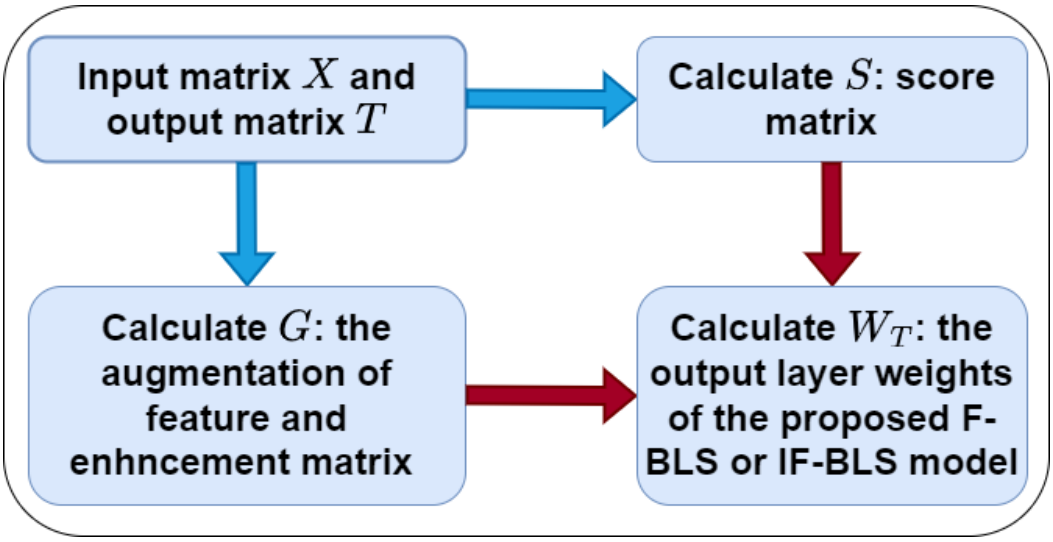


Figure 8.5: The flowchart of finding the output weight matrix W_T of the proposed F-BLS and IF-BLS models.

8.3.2 Complexity Analysis of the Proposed F-BLS and IF-BLS

Consider a dataset consisting of M samples. In IF-BLS, both membership and non-membership values are computed for each individual sample. As a result, the computational effort required to calculate these values is of $\mathcal{O}(M)$ [34]. The complexity of the proposed IF-BLS primarily depends on the matrix inversion computation. Following the standard procedure [208], the complexity of inverting a matrix of size $M \times M$ is $\mathcal{O}(M^3)$. For large values of M, $\mathcal{O}(M)$ becomes negligible in comparison to the $\mathcal{O}(M^3)$. Consequently, considering the complexity of the entire IF-BLS model, it can be approximated as $\mathcal{O}(M^3)$ due to the dominant contribution of the matrix inversion term. Since, in F-BLS, only membership values need to be calculated, therefore, the computational complexity of fuzzy membership is also $\mathcal{O}(M)$. Similarly, the overall computational complexity of F-BLS is also $\mathcal{O}(M^3)$.

8.3.3 Experiments and Results Analysis

To test the efficacy of proposed models, *i.e.,* F-BLS and IF-BLS, we compare them to baseline models on publicly available UCI [7] datasets with and without added Gaussian

Table 8.3: Hyperparameters' description and range for the baseline and proposed models.

Models	Parameters' description	Parameters' Range
BLS	$\mathcal{C}$: Regularization parameter m: Number of feature groups p: Number of feature nodes in each groups q: Number of enhancement nodes	$\mathcal{C} = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $m = 1:2:21$ $p = 5:5:50$ $q = 5:10:105$
ELM	$\mathcal{C}$: Regularization parameter h_l: Number of hidden nodes	$\mathcal{C} = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $h_l = 5:10:205$
NeuroFBLS	$\mathcal{C}$: Regularization parameter N_{fg}: Number of fuzzy groups N_{fn}: Number of fuzzy nodes in each groups q: Number of enhancement nodes	$\mathcal{C} = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $N_{fg} = 1:2:21$ $N_{fn} = 5:5:50$ $q = 5:10:105$
IF-TSVM	$\mathcal{C}_1$: Regularization parameter for the positive class $\mathcal{C}_2$: Regularization parameter for the negative class μ: Intuitionistic fuzzy Kernel parameter	$\mathcal{C}_1 = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $\mathcal{C}_2 = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $\mu = [2^{-5}, 2^{-4}, \cdots, 2^{5}]$
H-ELM	$\mathcal{C}$: Regularization parameter h_l: Number of hidden nodes	$\mathcal{C} = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $h_l = 5:10:205$
F-BLS	$\mathcal{C}$: Regularization parameter m: Number of feature groups p: Number of feature nodes in each groups q: Number of enhancement nodes	$\mathcal{C} = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $m = 1:2:21$ $p = 5:5:50$ $q = 5:10:105$
IF-BLS	$\mathcal{C}$: Regularization parameter m: Number of fuzzy groups p: Number of fuzzy nodes in each groups q: Number of enhancement nodes μ: Intuitionistic fuzzy Kernel parameter	$\mathcal{C} = [10^{-6}, 10^{-4}, \cdots, 10^{6}]$ $m = 1:2:21$ $p = 5:5:50$ $q = 5:10:105$ $\mu = [2^{-5}, 2^{-4}, \cdots, 2^{5}]$

Table 8.4: The testing accuracy of the proposed F-BLS and IF-BLS models along with the baseline models on UCI datasets.

Dataset ↓ | Model →	**#Samples**	**#Features**	**BLS [3]**	**ELM [22]**	**NeuroFBLS [56]**	**IF-TSVM [34]**	**H-ELM [206]**	**F-BLS †**	**IF-BLS †**
acute_inflammation	120	7	**100**	**100**	**100**	40.8333	**100**	**100**	100
acute_nephritis	120	7	**100**	**100**	**100**	51.6667	82.5	**100**	**100**
bank	4521	17	89.7366	89.4051	89.5817	88.4981	88.6308	**89.759**	89.4051
breast_cancer	286	10	69.8851	66.6727	70.1754	70.1754	**84.271**	72.2868	83.1579
breast_cancer_wisc	699	10	88.4183	87.9897	**90.7081**	66.6783	85.4162	88.2775	88.9938
chess_krvkp	3196	37	84.3862	72.0312	70.4004	25.1017	82.1028	84.2921	**84.9184**
conn_bench_sonar_mines_rocks	208	61	69.2451	60.5807	60.6272	22.7758	75.4936	69.1521	**80.2091**
credit_approval	690	16	87.5362	85.3623	84.4928	37.5362	86.8116	86.3768	**88.5507**
cylinder_bands	512	36	69.9258	66.2193	69.5336	60.8719	63.242	69.1338	**72.8536**
echocardiogram	131	11	83.9316	83.9031	80.9402	80.0855	84.6724	84.6724	**88.49**
fertility	100	10	90	89	**92**	88	89	91	91
haberman_survival	306	4	70.275	73.4902	73.4902	73.4902	74.146	69.6563	**75.4574**
hepatitis	155	20	85.1613	83.2258	**87.7419**	79.3548	82.5806	84.5161	**87.7419**
hill_valley	1212	101	**82.0185**	77.9706	78.9583	53.717	71.6155	81.6002	79.6249
horse_colic	368	26	86.1422	84.7908	83.9726	63.0285	86.1459	86.1385	**86.6938**
mammographic	961	6	78.6712	79.0889	79.504	53.689	72.3208	78.6744	**79.8192**
molec_biol_promoter	106	58	83.9394	68.961	65.1515	25.7143	82.1212	84.9351	**88.7879**
monks_1	556	7	75.3314	83.2497	**86.1277**	40.6403	66.7165	77.1396	77.6705
musk_1	476	167	75.8311	67.864	67.8487	22.1053	**89.4868**	76.6754	78.9846
oocytes_merluccius_nucleus_4d	1022	42	**82.7776**	79.8407	80.142	67.219	70.6428	81.8011	80.9201
oocytes_trisopterus_nucleus_2f	912	26	**78.9371**	76.4211	75.9911	58.1229	71.3799	78.6123	75.9899
pima	768	9	72.0058	71.4888	72.7918	65.1006	70.3081	71.4846	**73.3079**
pittsburg_bridges_T_OR_D	102	8	89.1905	88.1429	90.1429	86.1429	87.2857	88.1905	**90.2381**
spambase	4601	58	89.0021	87.1125	83.2657	60.6087	90.5259	89.4582	**90.7407**
spect	265	23	69.434	67.9245	67.9245	58.4906	66.7925	69.434	**72.4528**
statlog_heart	270	14	82.2222	80	80.7407	55.5556	80.7407	82.2222	**84.0741**
tic_tac_toe	958	10	**98.4315**	86.0068	82.7623	65.3125	75.7292	97.8081	97.0768
titanic	2201	4	77.9168	77.9168	78.0537	73.9173	77.3264	77.9168	**79.0532**
Average Accuracy			82.5126	79.8093	80.1096	58.3726	79.9287	<u>82.5434</u>	**84.5076**
Average Standard Deviation			6.5829	6.7853	7.4441	17.6829	10.3037	<u>6.1989</u>	**6.0494**
Average Rank			<u>3.0893</u>	4.6071	3.8036	6.8036	4.5357	3.2679	**1.8929**

The boldface and underline in the row denote the best and second-best performed model corresponding to the datasets. † represents the proposed models.

Table 8.5: Wilcoxon signed-rank test of the proposed F-BLS and IF-BLS models *w.r.t.* baseline models on UCI datasets.

Baseline →	**BLS**		**ELM**		**NeuroFBLS**		**IF-TSVM**		**H-ELM**	
Proposed ↓	p-value	Null hypothesis	p-value	Null hypothesis	p-value	Null hypothesis	p-value	Null hypothesis	p-value	Null hypothesis
F-BLS	0.4525	Not Rejected	0.001234	Rejected	0.2801	Not Rejected	7.94E-06	Rejected	0.00951	Rejected
IF-BLS	0.01917	Rejected	2.41E-05	Rejected	0.0009616	Rejected	2.81E-06	Rejected	1.28E-05	Rejected

Table 8.6: Pairwise win-tie-loss of proposed and existing models on UCI datasets.

	BLS [3]	**ELM [22]**	**NeuroFBLS [56]**	**IF-TSVM [34]**	**H-ELM [206]**	**F-BLS †**
ELM [22]	$[3, 3, 22]$					
NeuroFBLS [56]	$[10, 2, 16]$	$[15, 4, 9]$				
IF-TSVM [34]	$[2, 0, 26]$	$[1, 1, 26]$	$[0, 2, 26]$			
H-ELM [206]	$[7, 1, 20]$	$[11, 2, 15]$	$[10, 2, 16]$	$[28, 0, 0]$		
F-BLS †	$[9, 5, 14]$	$[21, 3, 4]$	$[16, 2, 10]$	$[27, 0, 1]$	$[19, 2, 7]$	
IF-BLS †	$[21, 2, 5]$	$[23, 3, 2]$	$[20, 3, 5]$	$[28, 0, 0]$	$[25, 1, 2]$	$[20, 3, 5]$

noise. Moreover, we implement the proposed models on the Alzheimer's disease (AD) dataset, available on the ADNI (adni.loni.usc.edu).

8.3.3.1 Setup for Experiments

The experimental procedures are executed on a computing system possessing MATLAB R2023a software, an 11th Gen Intel(R) Core(TM) i7-11700 processor operating at 2.50GHz with 16.0 GB RAM, and a Windows-11 operating platform.

Our study involves a comparative analysis among the proposed F-BLS and IF-BLS models and the compared baseline models, namely ELM [22], BLS [3], NeuroFBLS [56], intuitionistic fuzzy twin support vector machine (IF-TSVM) [34] and hierarchical extreme learned machine (H-ELM) [206].

We use the 5-fold cross-validation technique and grid search to fine-tune the hyperparameters of models. This involved splitting the dataset into five distinct, non-overlapping folds. For each set of hyperparameters, we calculate the testing accuracy on each fold separately. Average testing accuracy is determined for each set by taking the mean of these five accuracies. The highest average testing accuracy is reported as the accuracy of the models. The hyperparameters setting's range and description are reported in the Table 8.3.

8.3.3.2 Results and Statistical Analyses on UCI Dataset

We select 28 benchmark datasets available in the UCI repository [7], covering diverse domains. We include 14 datasets with sample sizes below 500 and 14 datasets whose sample sizes exceed 500, with feature counts ranging from 4 to 167. For detailed dataset characteristics, please refer to columns 2 and 3 of Table 8.4.

The performance of these models is assessed using accuracy and is presented in Table 8.4. The average accuracies for the existing models are as follows: BLS with an average

accuracy of 82.5126%, ELM with 79.8093%, NeuroFBLS with 80.1096%, IF-TSVM with 58.3726%, and H-ELM with 79.9287%. On the other hand, the proposed IF-BLS and F-BLS models achieved accuracies of 84.5076% and 82.5434%, respectively. In terms of accuracy, the proposed IF-BLS and F-BLS ranked first and second, respectively. A notable finding is that the proposed IF-BLS and F-BLS models showed the most minimal standard deviation values among the compared models, with 6.0494 and 6.1989, respectively. This shows that the proposed IF-BLS and F-BLS models possess a high degree of certainty in their predictions.

The proposed IF-BLS and F-BLS models attained an average rank of 1.8929 and 3.2679, respectively. This positions the IF-BLS model as the top-performing model, followed by the F-BLS model, making them the first and third best models, respectively.

The Friedman test compares the average ranks of models and determines whether the models have significant differences based on their rankings. The Friedman test follows the chi-squared distribution, and for $\mathscr{D} = 7$ and $\mathscr{K} = 28$, we get $\chi_F^2 = 86.1618$ and $F_F = 28.4265$. According to the statistical $F-$distribution table, $F_F(6, 162) = 2.1549$ at 5% level of significance. The null hypothesis is rejected since $28.4265 > 2.1549$. As a result, models differ significantly. Since the null hypothesis is rejected, we use the Wilcoxon signed-rank test [153] to assess the pairwise significant distinction among the proposed and baseline models. The Wilcoxon signed-rank test results are presented in Table 8.5; the p-values and the corresponding status of the null hypothesis in pairwise comparisons between the proposed and baseline models underscore a compelling narrative. The consistent rejection of the null hypothesis of the IF-BLS model $w.r.t.$ all the baseline models signifies a statistical superiority of the IF-BLS model, whereas the Wilcoxon signed-rank test confirms that the proposed F-BLS is superior to baseline ELM, IF-TSVM, and H-ELM models. This analysis solidifies the commendable performance and statistical excellence of the proposed

F-BLS and IF-BLS models across the board.

In addition, to analyze the models, we use pairwise win-tie-loss sign test. For $\mathscr{K} = 28$, the threshold for determining statistical difference according to the win-tie-loss test is equal to $\mathscr{K}/2 + 1.96\sqrt{\mathscr{K}}/2 = 28/2 + 1.96\sqrt{28}/2 = 19.1857$. The pairwise win-tie-loss of models are noted in Table 8.6. If either of the two models achieves victories in a minimum of 20 datasets, it establishes a clear statistical distinction between them. Upon examination of Table 8.6, we observe that the proposed IF-BLS model wins over the BLS, ELM, NeuroFBLS, IF-TSVM, H-ELM, and F-BLS models by securing a number of victories: 21, 23, 20, 28, 25, and 20 respectively. Thereby substantiating the superiority of the proposed IF-BLS model over all the baseline models and proposed F-BLS counterparts. This outcome reinforces the notion that the IF-BLS model possesses notable advantages and stands as a compelling choice in terms of its overall effectiveness.

Analysis: An intriguing finding emerges from comparing the proposed F-BLS models with the proposed IF-BLS. As expected, the proposed IF-BLS outperforms the proposed F-BLS model, suggesting a plausible explanation. The F-BLS model intelligently handles noise and outliers by assigning fuzzy values to each sample based on membership value considerations. However, relying only on fuzzy membership without incorporating the measure of the extent of non-belongingness to a class is not the optimal way to handle noise and outliers effectively. The remarkable performance of the IF-BLS model offers a compelling solution. By considering both membership and non-membership values when calculating the degree of fuzzy values, the IF-BLS model demonstrates its indispensability in effectively addressing the challenges posed by noise and outliers. This emphasizes how important it is to take into account both membership and non-membership values in order to combat the impact of noise and outliers.

Table 8.7: The classification accuracies of the proposed F-BLS and IF-BLS models along with the existing models, *i.e.,* BLS, ELM, NeuroFBLS, IF-TSVM, and H-ELM on UCI dataset with varying levels of 5%, 10%, 15%, and 20% Gaussian noise.

Dataset ↓ | Model →	**Noise**	**BLS [3]**	**ELM [22]**	**NeuroFBLS [56]**	**IF-TSVM [34]**	**H-ELM [206]**	**F-BLS †**	**IF-BLS †**
breast_cancer	5%	70.2359	69.4737	70.1754	70.1754	**87.0175**	71.2704	82.807
	10%	71.9722	67.0175	70.1754	70.1754	**83.8838**	72.6558	82.4622
	15%	70.8832	66.6667	70.5263	70.1754	70.5263	72.3351	**86.6667**
	20%	71.9419	67.3926	70.5263	70.1754	76.4912	70.5384	**81.7604**
conn_bench_sonar_mines_rocks	5%	68.734	57.7003	61.5447	17.9675	73.9141	68.6527	**78.4204**
	10%	67.2009	57.2474	59.0825	18.9547	76.8293	66.8293	**78.2695**
	15%	67.3171	67.3403	58.2346	14.5645	**81.2311**	65.8653	77.8513
	20%	67.7352	60.5923	61.0221	16.0279	73.1243	68.2695	**75.4704**
hill_valley	5%	81.4359	79.0416	79.784	51.6536	74.1764	**82.3433**	81.9311
	10%	**80.9417**	76.4847	78.1356	51.4067	71.7832	79.4603	80.7758
	15%	80.2809	75.579	76.7354	50.1666	70.9693	79.3725	**80.3639**
	20%	79.2089	74.337	76.4837	50.0027	71.2135	79.4538	**80.0337**
pittsburg_bridges_T_OR_D	5%	89.1905	86.1429	89.2381	86.1429	86.2857	88.1429	**93.1429**
	10%	88.1905	87.2857	90.1429	86.1429	86.1429	88.2381	**92.1429**
	15%	88.2381	86.1905	87.2381	86.1429	86.1429	87.1905	**92.1429**
	20%	87.2381	86.1429	92.1429	86.1429	86.1905	90.1429	**92.1905**
tic_tac_toe	5%	**97.7018**	81.9181	83.3922	65.4167	67.1673	97.284	97.3882
	10%	**97.3893**	83.5951	80.8901	65.3125	76.3367	97.3887	97.2846
	15%	96.0318	82.4689	78.9998	65.3125	67.5125	95.9271	**96.448**
	20%	94.6755	79.2174	76.8161	65.3125	73.6447	93.9431	**95.8246**
Average Accuracy		80.8272	74.5917	75.5643	57.8686	77.0292	80.7652	**86.1689**

The boldface in each row denotes the best-performed model corresponding to each dataset. † represents the proposed models.

8.3.3.3 Evaluation on UCI Datasets with Gaussian Noise

While the UCI datasets used in our study reflect real-world scenarios, it is worth noting that impurities or noise in collected data can increase for various reasons. In such circumstances, it becomes imperative to develop a robust model capable of effectively handling such challenging scenarios. To demonstrate the superiority of the proposed F-BLS and IF-BLS models even in adverse situations, we introduce Gaussian noise to selected UCI datasets. We have chosen 5 diverse UCI datasets for our comparative analysis. The reasoning behind selecting the datasets is as follows: the chosen 5 diverse UCI datasets include breast_cancer, conn_bench_sonar_mines_rocks, hill_valley, pittsburg_bridges_T_OR_D, and tic_tac_toe, each representing a different domain. In order to ensure fairness in evaluating the models, we have selected 3 datasets where none of the proposed F-BLS and IF-BLS models achieve the highest performance at the noise level of 0% (indicating a loss for the proposed models). Additionally, we have chosen 2 datasets where the proposed F-BLS or

IF-BLS model outperforms other models at the 0% noise level (indicating a win for the proposed models). For detailed information regarding the selected datasets, please refer to Table 8.8. To conduct a comprehensive analysis, we introduce Gaussian noise with varying levels of 5%, 10%, 15%, and 20% to corrupt the features of these datasets.

Table 8.8: Description of UCI datasets chosen for experiments with added Gaussian noise.

Dataset	Subject Area	Win/Loss of the proposed IF-BLS model in 0% noise
breast_cancer	Life	Loss
conn_bench_sonar_mines_rocks	Physical	Win
hill_valley	Other (Synthetic)	Loss
pittsburg_bridges_T_OR_D	Civil Engineering	Win
tic_tac_toe	Game	Loss

Comparative analysis: The accuracy of all the models in the chosen datasets with 5%, 10%, 15%, and 20% noise are presented in Table 8.7.

[1] The IF-BLS model consistently achieved the top position in the conn_bench_sonar_mines_rocks and pittsburg_bridges_T_OR_D at 0% noise. Remarkably, it maintained its superior performance even when noise is introduced. In the conn_bench_sonar_mines_rocks dataset, the IF-BLS model demonstrated the best performance with an accuracy of 75.4704% at 20% noise, which is approx 8% higher than the standard BLS model. We can observe an even better pattern for F-BLS in comparison to BLS. At 0% noise, BLS outperformed the proposed F-BLS, but as the noise increases, the F-BLS starts catching up, and at 20% noise, the F-BLS outperforms the BLS and shows superiority in dealing with noise. Similar results are observed with the pittsburg_bridges_T_OR_D dataset.

[2] In Table 8.7, we observe interesting patterns in breast_cancer, hill_valley, and tic_tac_toe datasets. Initially, the proposed F-BLS and IF-BLS models are not the top performers at 0% noise. However, as the level of noise increased, proposed models began to outperform the baseline models. For instance, in the hill_valley dataset,

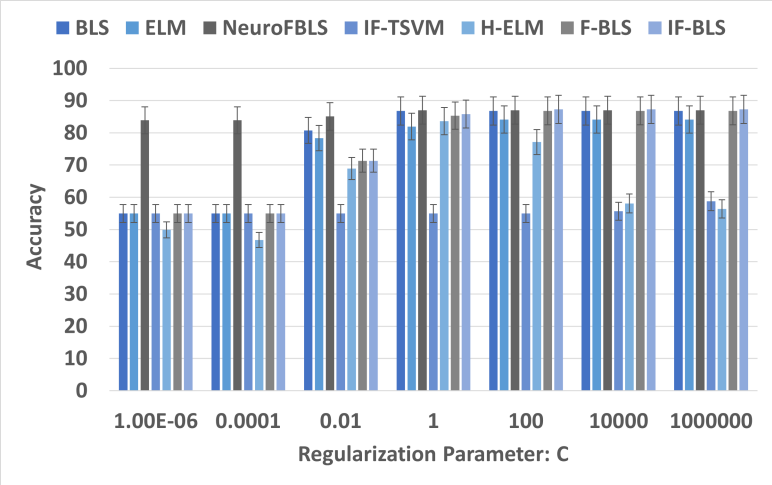
the BLS outperforms the proposed models at 0% noise. However, as noise increases, the IF-BLS model becomes the top performer starting from a noise level of 15%, while the F-BLS model surpasses BLS at a 20% noise level. Other baseline models do not come close to matching the performance of the proposed models. This trend is consistent across other datasets as well.

[3] The proposed IF-BLS and F-BLS models have secured the first and third positions in terms of average accuracy with values of 86.1689% and 80.7652%, respectively. The proposed IF-BLS model's average accuracy is around 6%, 12%, 11%, 29% and 9% higher than that of baseline models: BLS, ELM, NeuroFBLS, IF-TSVM, H-ELM, respectively. Whereas the proposed F-BLS has around 6%, 5%, 23%, and 3% higher average accuracy than that of baseline models: ELM, NeuroFBLS, IF-TSVM, H-ELM, respectively. Furthermore, the proposed F-BLS and IF-BLS models exhibit high prediction certainty, with the lowest standard deviations of 9.4511 and 8.8589, respectively, compared to the baselines. The combination of the highest average accuracy and the lowest standard deviation indicates that the proposed IF-BLS and F-BLS models are less noise-sensitive and exhibit robust performance in handling varying conditions.

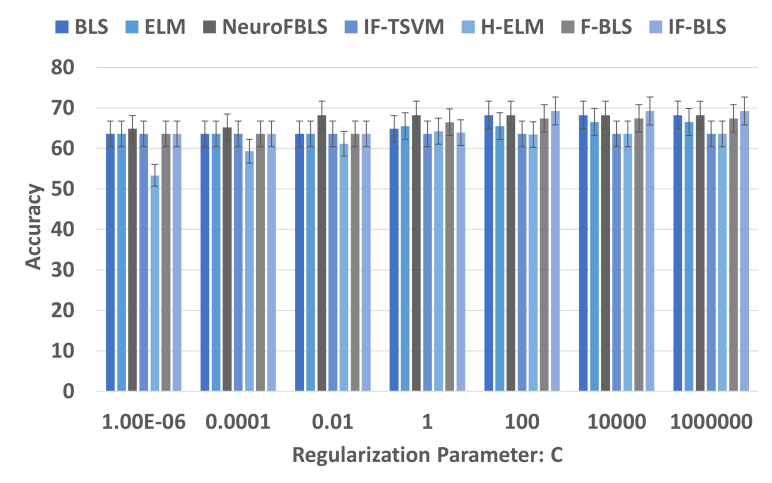
We subjected the proposed models to challenging conditions and found that the proposed F-BLS and IF-BLS models showcased their robustness and superior performance in unfavorable scenarios. This highlights their adeptness in navigating and excelling in noise-affected environments.

8.3.3.4 Evaluation on ADNI Dataset

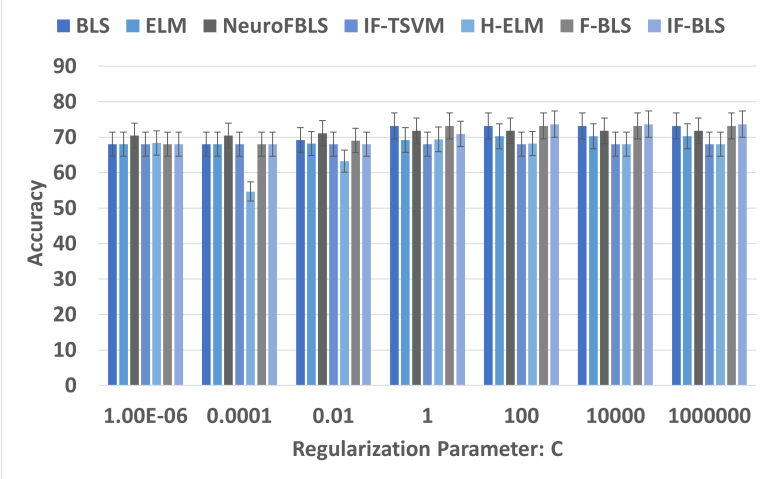
Alzheimer's Disease (AD) is a degenerative neurological disorder that leads to the gradual deterioration of the brain and is the primary cause of dementia, accounting for 60-80%



(a) CN_VS_AD



(b) CN_VS_MCI



(c) MCI_VS_AD

Figure 8.6: Accuracy along with the standard deviation (shown in thin lines on the top) with respect to the regularization parameter $\mathcal{C}$ on the ADNI dataset.

Table 8.9: The testing accuracy of the proposed F-BLS and IF-BLS models and the baseline models on ADNI datasets.

Dataset ↓ | Model →	**BLS [3]**	**ELM [22]**	**NeuroFBLS [56]**	**IF-TSVM [34]**	**H-ELM [206]**	**F-BLS †**	**IF-BLS †**
CN_VS_AD	88.4337	86.747	88.6747	58.7952	82.8916	88.6387	**88.8137**
CN_VS_MCI	70.7695	69.0108	70.4495	63.5733	65.6533	70.2883	**71.0844**
MCI_VS_AD	75.0427	73.3333	74.5299	68.0342	71.6239	74.8718	**76.2437**
Average Accuracy	78.082	76.3637	77.8847	63.4676	73.3896	77.9329	**78.7139**

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

of dementia cases [209]. Predominantly affecting individuals aged 60 and older, it primarily impairs memory and various cognitive functions. AD can lead to profound changes in behavior, personality, and overall quality of life. Despite decades of research, there is currently no cure for AD, and available treatments only provide temporary symptomatic relief [210]. Therefore, understanding the complex underlying mechanisms of the disease remains a critical area of investigation to alleviate the burden of AD on individuals, families, and societies.

To train the proposed F-BLS and IF-BLS models, we utilize scans from the ADNI dataset, which can be accessed at $adni.loni.usc.edu$. The inception of the ADNI project dates back to 2003 as a public-private partnership. Wherein its fundamental objective centered around the thorough examination and exploration of diverse neuroimaging techniques, encompassing magnetic resonance imaging (MRI), positron emission tomography (PET), and other diagnostic assessments, to elucidate the intricate nuances of AD at its early stage, characterized by mild cognitive impairment (MCI). The dataset encapsulates three distinctive scenarios, specifically the comparative analysis of control normal (CN) versus MCI (CN_VS_MCI), MCI versus AD (MCI_VS_AD), and CN versus AD (CN_VS_AD).

Table 8.9 shows the accuracies achieved by different models for diagnosing AD. The proposed IF-BLS excels with a maximum average testing accuracy of 78.7139%, whereas the proposed F-BLS secures third position with an average accuracy of 77.9329%. The proposed IF-BLS model has the highest accuracy 88.8137% for the CN_VS_AD case, fol-

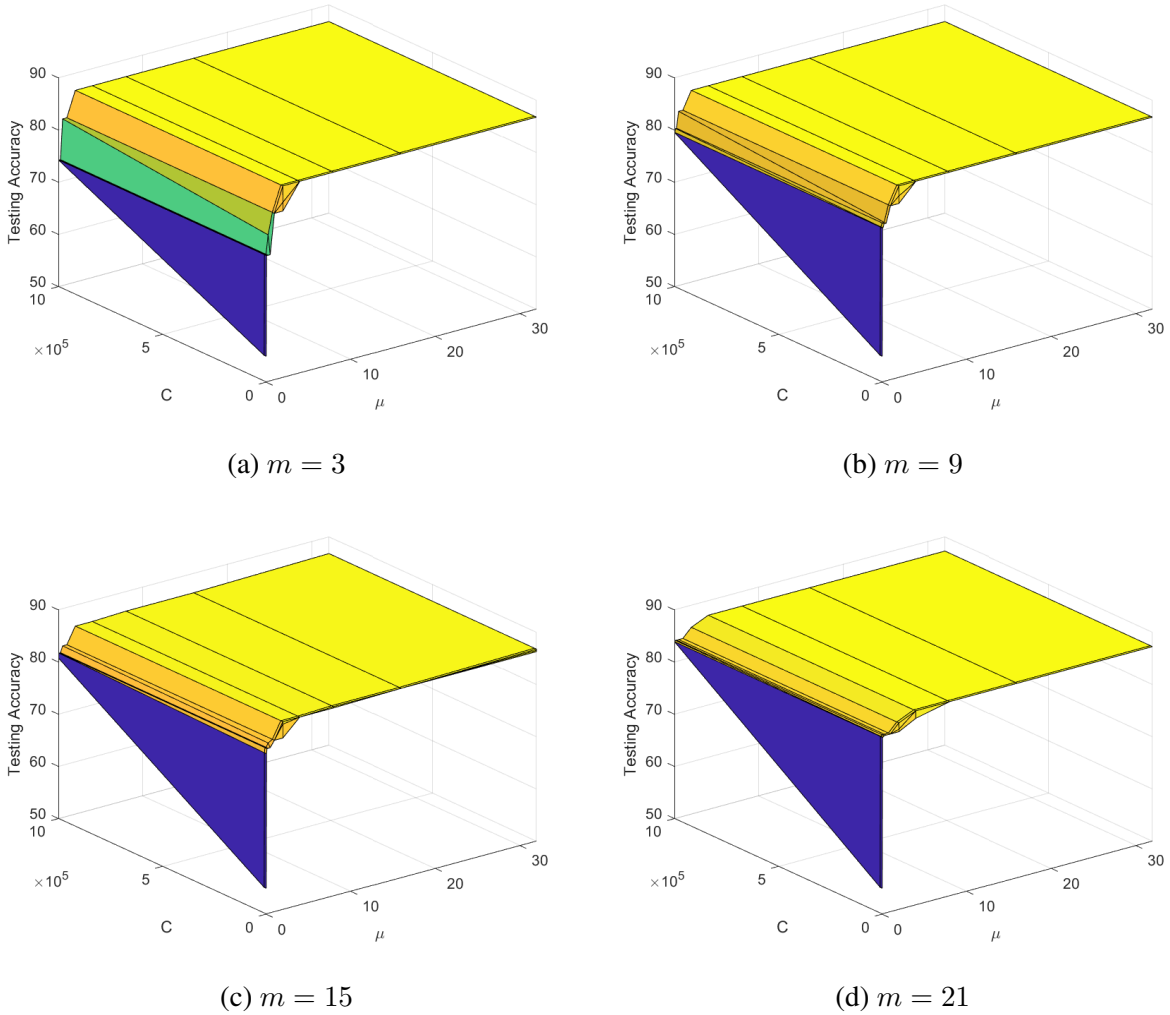
lowed by NeuroFBLS and F-BLS with accuracy 88.6747% and 88.6387%, respectively. For the CN_VS_MCI case, the proposed IF-BLS again comes out on top with the average testing accuracy of 71.0844%. In the MCI_VS_AD scenario, the IF-BLS model achieves the highest accuracy of 76.2437%, while the F-BLS model follows closely with an accuracy of 74.8718%, securing third place. In the overall comparison, the proposed IF-BLS model emerges as a top-performing model by consistently achieving high accuracies across different cases, and the proposed F-BLS shows a competitive nature. These findings highlight the effectiveness of the IF-BLS and F-BLS models in accurately distinguishing individuals with CN, MCI, and AD cases.

8.3.3.5 Sensitivity Analyses of the Hyperparameters of the Proposed Models

To understand the behavior of the proposed models, it is essential to understand the dependence of models on their hyperparameters. Thus, we conduct the following sensitivity analyses of the hyperparameters to gain a comprehensive perspective of the proposed models: (i) investigating the impact of varying the number of regularization parameters ($\mathcal{C}$) on model performance, (ii) assessing the dependence of the proposed IF-BLS models on Feature Groups (m), kernel parameter (μ), and $\mathcal{C}$, (iii) exploring the influence of the number of feature nodes (p) on both F-BLS and IF-BLS models.

8.3.3.5.1 Performance Dependence on Regularization Parameter ($\mathcal{C}$)

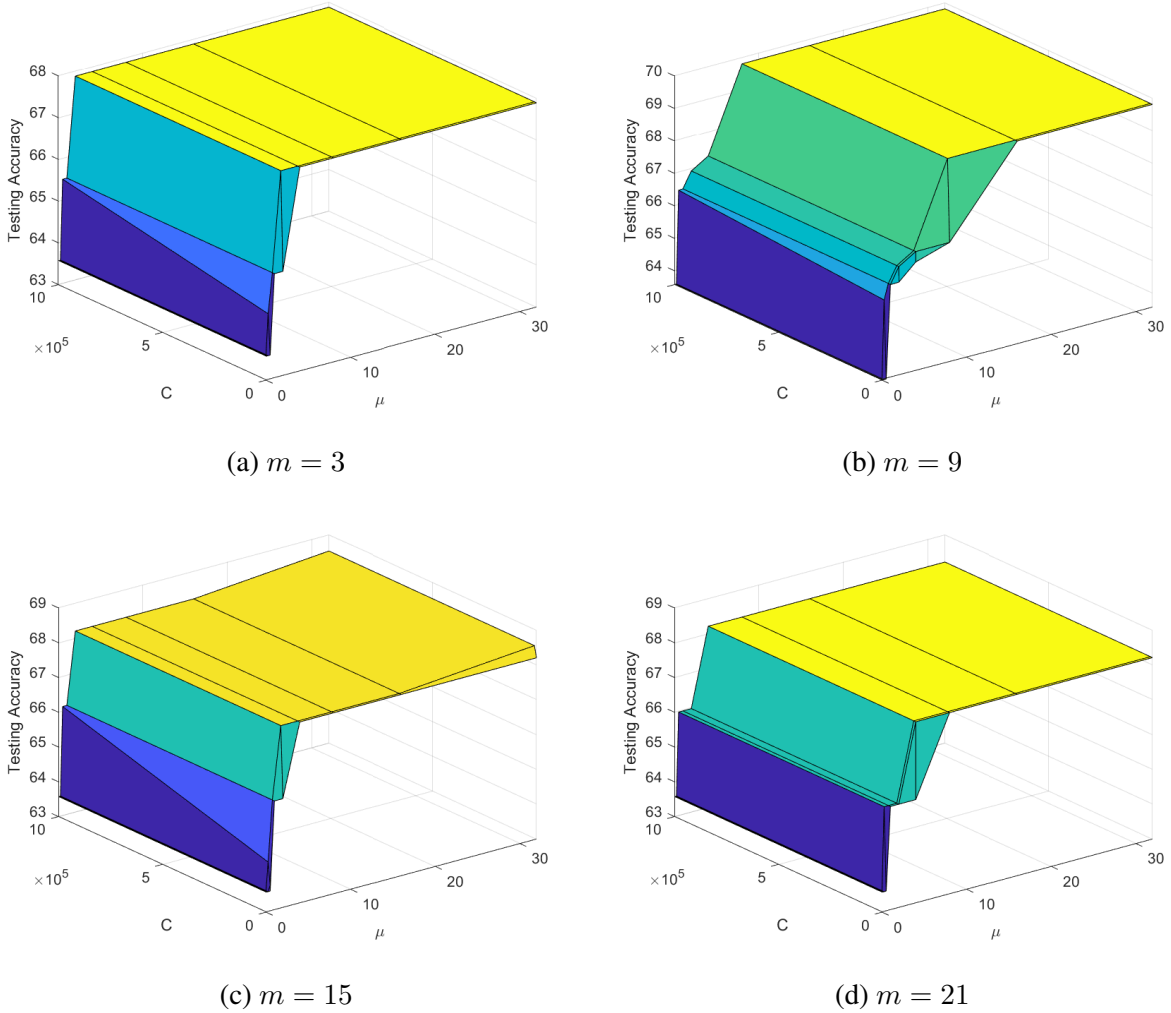
To investigate the impact of varying the number of regularization parameters ($\mathcal{C}$), we draw Figure 8.6, which shows the testing accuracy along with the standard deviation (shown in thin lines on the top) of each model with respect to the regularization parameter $\mathcal{C}$ on the ADNI dataset. For each case, we fix the remaining hyperparameters to their best values. From Figure 8.6, we observe that, with the increase in values of $\mathcal{C}$, testing accuracies of each model are also increasing. At $\mathcal{C} = 10^6$, IF-BLS outperforms other models with the



(a) $m = 3$ (b) $m = 9$

(c) $m = 15$ (d) $m = 21$

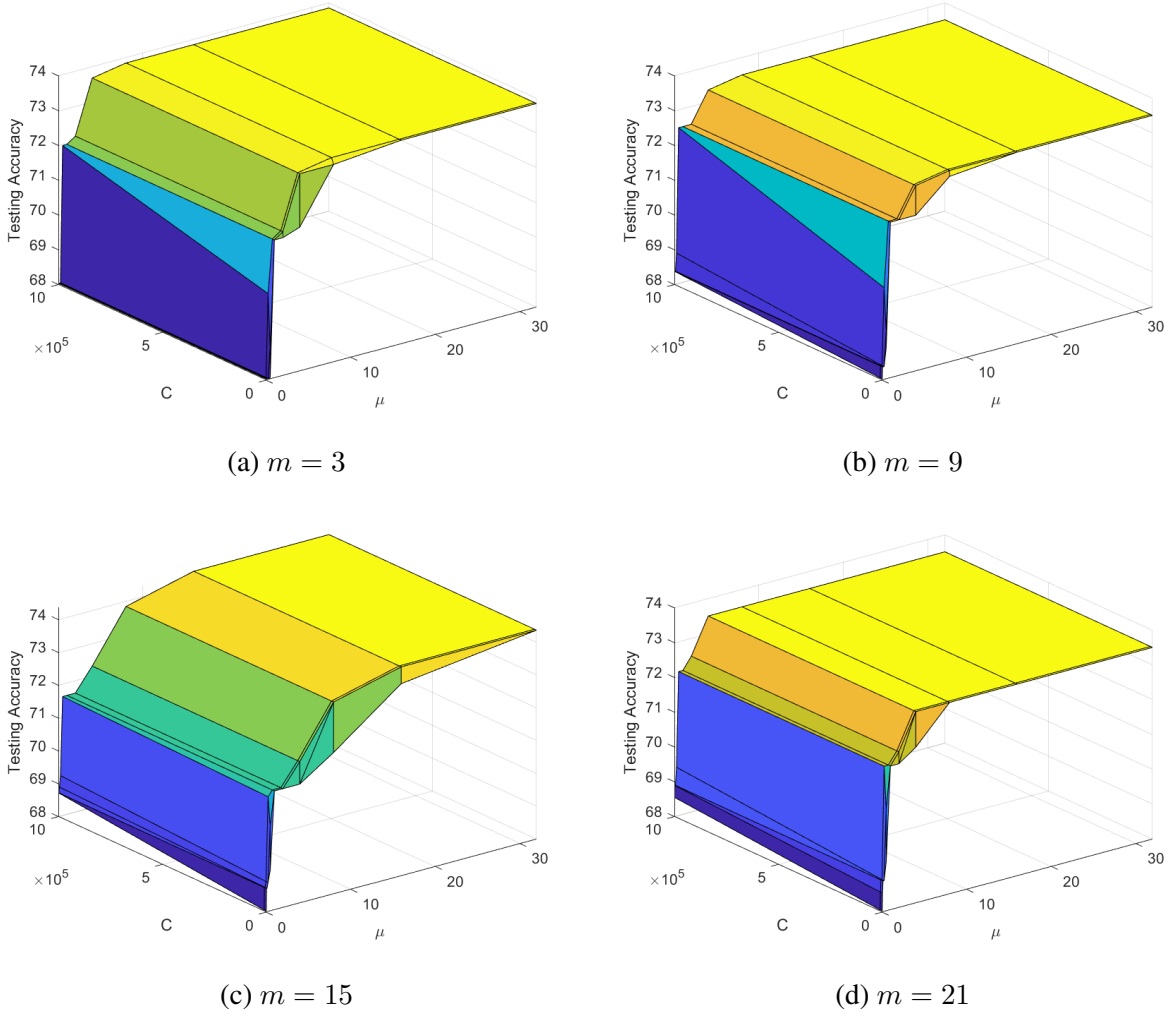
Figure 8.7: Evaluation of the proposed IF-BLS model's testing accuracy on the CN_VS_AD case for Alzheimer's disease diagnosis, considering different numbers of feature groups (m), while varying the Kernel parameter (μ) and regularization parameter ($\mathcal{C}$).



(a) $m = 3$ (b) $m = 9$

(c) $m = 15$ (d) $m = 21$

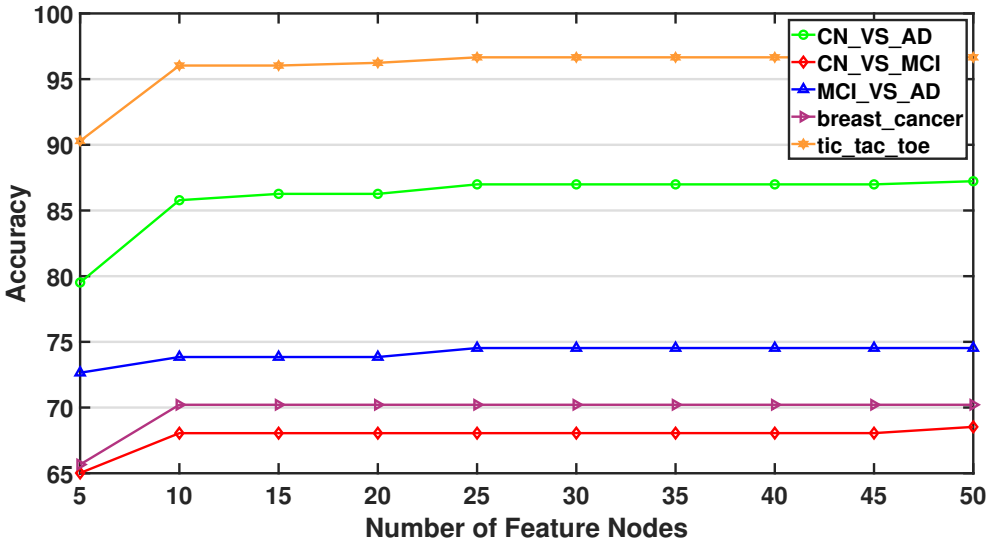
Figure 8.8: Evaluation of the proposed IF-BLS model's testing accuracy on the CN_VS_MCI case for Alzheimer's disease diagnosis, considering different numbers of feature groups (m), while varying the Kernel parameter (μ) and regularization parameter (C).



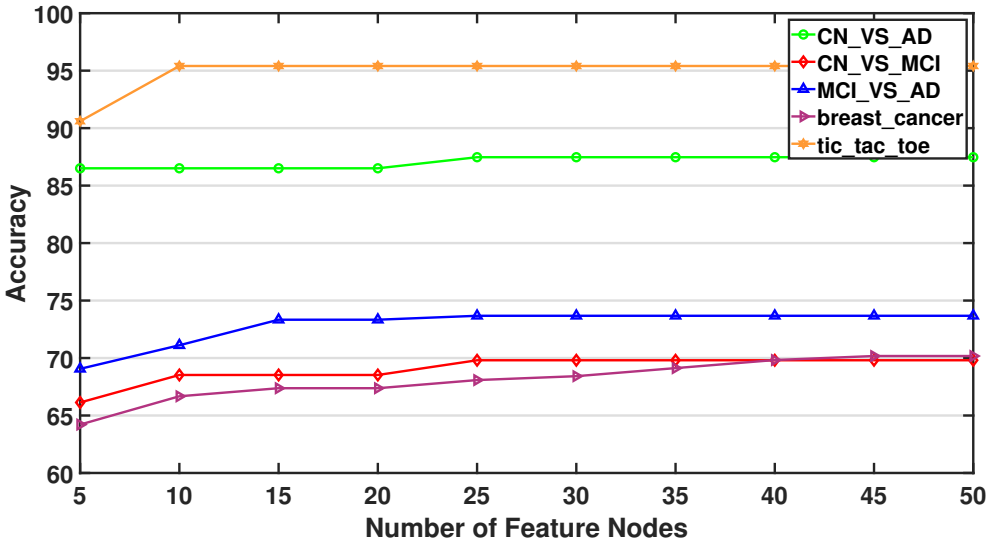
(a) $m = 3$ (b) $m = 9$

(c) $m = 15$ (d) $m = 21$

Figure 8.9: Evaluation of the proposed IF-BLS model's testing accuracy on the MCI_VS_AD case for Alzheimer's disease diagnosis, considering different numbers of feature groups (m), while varying the Kernel parameter (μ) and regularization parameter ($\mathcal{C}$).



(a) F-BLS



(b) IF-BLS

Figure 8.10: Evaluation of the proposed F-BLS and IF-BLS models' testing accuracy while varying the number of feature nodes (p).

highest testing accuracy in each case.

8.3.3.5.2 Performance Dependence on Feature Groups (m), the Kernel Parameter (μ) and Regularization Parameter ($\mathcal{C}$)

The testing accuracy of the proposed IF-BLS model for Alzheimer's disease diagnosis is depicted in Figures 8.7, 8.8, and 8.9 for the CN_VS_AD, CN_VS_MCI, and MCI_VS_AD cases, respectively. The accuracy is evaluated by considering different numbers of feature groups (m) while varying the Kernel parameter (μ) and regularization parameter ($\mathcal{C}$). One can observe that as the values of μ and $\mathcal{C}$ increase, the accuracy also increases. However, beyond a certain point, the accuracy plateau is reached, indicating that the accuracy becomes less responsive to increases in μ and $\mathcal{C}$. Therefore, more attention is required to choose the hyperparameters.

8.3.3.5.3 Performance Dependence on Number of Feature Nodes (p)

Moreover, to assess the impact of varying the number of feature nodes on the performance of the proposed F-BLS and IF-BLS models, Figure 8.10 illustrates the relationship between the number of feature nodes and accuracy across five datasets: CN_VS_AD, CN_VS_MCI, MCI_VS_AD, breast_cancer, and tic_tac_toe. The figure shows that increasing the number of feature nodes initially boosts accuracy. However, once a certain point is reached, increasing the number of feature nodes doesn't lead to much improvement in accuracy.

Specifically, for the F-BLS model, optimal accuracy is attained with just 25 nodes across all datasets. Conversely, the IF-BLS model exhibits varying patterns: for CN_VS_AD, tic_tac_toe, and MCI_VS_AD datasets, peak accuracy is also achieved with 25 nodes, whereas for CN_VS_MCI and breast_cancer datasets, the optimal accuracy is achieved with 40 nodes. Consequently, a general recommendation emerges: fine-tuning the number of feature nodes from 25 onwards offers a promising avenue towards achieving optimal accu-

racy. However, it's important to note that the model's performance is dataset-dependent, necessitating careful hyperparameter tuning for each dataset to maximize accuracy.

8.4 Summary

In this chapter, to improve robustness against noisy and outlier-contaminated data, we proposed intuitionistic fuzzy-based deep and ensemble RVFL models, namely IF-dRVFL and IF-edRVFL. These models incorporate intuitionistic fuzzy theory to assign adaptive weights to training samples based on membership and non-membership degrees derived from class centroid distances and neighborhood heterogeneity. This weighting strategy enables the models to distinguish between reliable, noisy, and outlier samples, thereby mitigating the adverse impact of noise propagation across hidden layers. Experimental evaluations on UCI and KEEL benchmark datasets, under both clean and noisy settings, demonstrate that the proposed IF-dRVFL and IF-edRVFL models achieve superior performance compared to several state-of-the-art randomized neural network and fuzzy learning approaches.

Further, we propose the fuzzy BLS (F-BLS) and intuitionistic fuzzy BLS (IF-BLS) models. The proposed F-BLS model considers the distance from samples to the class center in the sample space while assigning weight to each sample. Meanwhile, IF-BLS assigns scores to training points in the kernel space. The effectiveness of the proposed F-BLS and IF-BLS models is demonstrated by applying them to standard benchmark datasets obtained from the UCI repository. Through statistical analyses such as average accuracy, ranking scheme, Friedman test, Wicoxon-signed rank test, and win-tie-loss sign test, both the proposed F-BLS and IF-BLS models exhibit superiority over baseline models. To test the robustness of the proposed F-BLS and IF-BLS models, we introduced Gaussian noise to various UCI datasets. Results indicate that the proposed models exhibit accuracy improvements ranging from 3% to 29% compared to baseline models. The proposed F-BLS and IF-BLS models

are also employed for AD diagnosis. The proposed F-BLS shows a competitive nature and the IF-BLS model achieved the highest average accuracy as well as the highest testing accuracy across all three cases (AD_VS_MCI, AD_VS_CN, and CN_VS_MCI). In line with the research findings [191], the MCI_VS_AD case is identified as the most challenging case in AD diagnosis. However, the proposed F-BLS and IF-BLS models attained the top positions with testing accuracy of 74.8718% and 76.2437%, respectively. The proposed F-BLS and IF-BLS models consistently show lower standard deviations compared to baseline models, indicating their higher certainty and robust performance across diverse conditions.

Chapter 9

Comprehensive Evaluation of the Proposed Methods

This chapter presents a comprehensive empirical evaluation of the randomized neural network models developed throughout this thesis. The proposed methods encompass interpretable randomized neural networks, robust and scalable learning frameworks, geometry- and class imbalance-aware models, complex-valued randomized neural networks, expressive deep and ensemble randomized neural networks, and broad randomized neural networks. Extensive experiments are conducted on a diverse collection of 123 benchmark datasets, including binary UCI datasets, multiclass UCI datasets, KEEL imbalanced datasets, and large-scale tabular datasets. The proposed models are compared with their respective baseline algorithms using multiple evaluation metrics, including Accuracy, Area Under the ROC Curve (AUC), Sensitivity, Specificity, Precision, F-measure, and G-mean. The experimental results demonstrate that the proposed methods consistently achieve competitive or superior predictive performance across different learning scenarios while preserving robustness, interpretability, scalability, and computational efficiency.

9.1 Introduction

The previous chapters presented a series of randomized neural network models designed to address several important challenges in machine learning, including interpretability, robustness, scalability, class imbalance, geometric representation learning, complex-valued feature transformation, expressive deep architectures, and broad learning systems. Each proposed model was experimentally validated on benchmark datasets corresponding to its specific research objective. However, a comprehensive evaluation of all the developed models within a unified experimental framework provides a clearer understanding of their overall effectiveness and practical applicability.

This chapter consolidates the experimental results of all the proposed randomized neural network models presented throughout this thesis. The objective is to provide a unified comparative evaluation using diverse benchmark datasets covering binary classification, multiclass classification, imbalanced learning, and large-scale tabular learning problems. Such a comprehensive evaluation enables the assessment of the proposed methods under different data characteristics, including a class imbalance scenario.

Depending on the characteristics of the datasets, the evaluation is performed using several widely adopted performance metrics, including Accuracy, Area Under the Receiver Operating Characteristic Curve (AUC), Sensitivity, Specificity, Precision, F-measure, and G-mean. The comprehensive experimental study presented in this chapter further validates the effectiveness of the proposed randomized neural network frameworks and demonstrates their capability to achieve improved predictive performance while maintaining the computational efficiency inherent to randomized learning methods.

9.2 Experimental Evaluation

This section presents a comprehensive experimental evaluation of all the randomized neural network models proposed throughout this thesis. To provide a unified and systematic assessment, extensive experiments are conducted on a total of 123 benchmark datasets. The experimental results are organized into four benchmark categories. First, the proposed models are evaluated on 27 binary UCI datasets to assess their performance on binary classification tasks. Subsequently, experiments are conducted on 30 multiclass UCI datasets to investigate their generalization capability in multiclass learning scenarios. The robustness of the proposed methods under severe class imbalance is further examined using 56 KEEL imbalanced datasets, where multiple evaluation metrics beyond classification accuracy are considered. Finally, the scalability and computational effectiveness of the proposed models are demonstrated through experiments on 10 large-scale NDC datasets containing millions of instances. For each experimental setting, the proposed methods are compared with their corresponding baseline algorithms under identical experimental protocols and hyperparameter tuning strategies to ensure a fair and consistent comparison.

9.2.1 Discussion on Binary UCI Datasets

Table 9.1 presents the classification performance of the proposed randomized neural network models, namely NF-RVFL-R, NF-RVFL-K, NF-RVFL-C, F-BLS, IF-BLS, RVFL-X-N, RVFL-X-A, and Wave-RVFL, on 27 binary UCI benchmark datasets. Overall, all proposed models achieve competitive performance, demonstrating the effectiveness of the different randomized neural network architectures developed throughout this thesis for binary classification problems. In terms of the average classification accuracy, RVFL-X-A attains the highest value of 82.1911%, followed by NF-RVFL-K (82.0035%), NF-RVFL-C (81.9654%), NF-RVFL-R (81.9207%), and RVFL-X-N (81.2490%). The remaining mod-

Table 9.1: Results on 27 binary UCI datasets.

	NF-RVFL-R		NF-RVFL-K		NF-RVFL-C		F-BLS		IF-BLS		RVFL-X-N		RVFL-X-A		Wave-RVFL	
Dataset	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.
acute_inflammation	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0
acute_nephritis	100	0	100	0	100	0	100	0	100	0	100	0	100	0	90	10.8653
bank	89.4269	0.6225	89.4492	0.4345	90.9391	0.6354	89.759	0.6556	89.4051	0.6945	89.8475	0.6102	89.8032	0.4139	88.5202	0.5503
blood	77.838	11.8577	77.8398	13.1888	77.4398	12.6107	75.7002	11.452	77.4398	13.2299	77.0398	14.4892	78.1047	11.0233	76.5065	14.7975
breast_cancer	72.3351	21.0696	72.6921	19.6764	70.8953	21.6404	72.2868	27.5906	83.1579	26.8828	70.1754	44.6249	70.5263	43.9368	92.6316	16.4763
breast_cancer_wisc_prog	83.359	4.4786	81.8846	6.324	80.9103	10.3056	78.3077	3.6281	79.8718	6.2766	81.8718	7.9163	82.8462	6.7628	79.3462	6.3219
congressional_voting	63.908	4.1919	63.926	1.8336	64.1379	3.5799	60.2299	5.7815	59.5402	4.2544	63.6782	2.885	64.8276	2.885	62.5287	3.4097
conn_bench_sonar_mines_rocks	65.8885	16.4039	66.2602	11.3816	63.4611	8.678	69.1521	11.3172	80.2091	13.5518	63.4959	5.7475	74.007	11.9465	94.6341	11.9984
credit_approval	85.942	9.5394	85.942	10.3575	87.8581	8.3691	86.3768	11.125	88.5507	10.3423	85.5072	11.2143	86.087	9.9278	78.5507	11.7472
echocardiogram	85.4416	5.1137	85.4416	5.1137	85.4416	6.9527	84.6724	7.2724	88.49	6.1265	85.4416	6.9527	86.2108	5.909	74.1311	11.302
fertility	91	7.4162	92	6.7082	91	8.2158	91	8.2158	91	8.2158	91	8.2158	91	8.2158	90	7.9057
haberman_survival	75.2396	6.2369	75.1296	8.9272	75.4574	8.021	69.6563	10.0681	75.4574	8.5088	74.146	8.6276	75.4574	9.0446	74.4738	9.2421
heart_hungarian	76.9024	12.2125	76.5576	9.7118	78.2466	16.8612	78.9421	9.5844	83.3255	12.5494	73.1268	12.3092	75.149	20.3987	96.6102	7.5799
hepatitis	85.8065	13.0236	87.0968	7.9016	85.8065	11.081	84.5161	9.5148	87.7419	4.2059	87.0968	9.9426	84.5161	8.9512	84.5161	8.3498
hill_valley	82.8429	6.3314	82.6732	5.1213	83.0874	6.7321	81.6002	3.4346	79.6249	4.279	83.0028	5.2655	85.9742	5.5675	55.8572	3.3437
horse_colic	86.6938	2.5582	86.4272	3.661	86.6901	2.5857	86.1385	6.0478	86.6938	2.5582	86.1459	2.1863	86.4272	3.7837	72.0104	6.1297
ilpd_indian_liver	72.3932	5.6986	72.3917	4.278	72.2178	5.7289	69.4769	4.2284	72.3843	6.3894	72.3814	5.9626	73.4188	5.2027	73.2508	5.3362
ionosphere	88.3461	8.3898	89.7626	7.5323	88.9095	5.93	88.3501	8.7867	89.1871	6.4747	91.7505	3.9124	90.8974	5.3548	70.998	18.0278
mammographic	79.2957	5.0757	79.504	5.222	79.1899	4.649	78.6744	4.8822	79.8192	5.3901	80.2321	5.4083	79.1942	5.2359	68.7997	9.6799
molec_biol_promoter	82.0346	5.3415	82.12	2.856	82.0346	7.9086	84.9351	3.7913	88.7879	9.5415	70.6494	14.2114	73.5065	11.1189	92.381	17.0367
monks_3	91.8436	7.9554	91.6486	6.497	92.1744	4.583	87.3612	2.3232	83.7396	7.7034	92.9615	1.948	93.1433	3.9534	64.0721	8.0867
parkinsons	83.0769	12.6398	83.0769	15.8685	85.5856	13.3584	80	15.7437	82.0513	9.0655	82.5641	12.9988	83.5897	18.8945	87.1795	12.4299
planning	73.018	8.381	71.9369	7.6121	71.9369	7.6121	71.4114	9.5837	74.2342	8.835	72.4925	7.9839	73.033	8.337	73.6036	11.0985
spambase	88.9674	4.6103	88.9583	5.1207	89.3931	4.0775	89.4582	2.8438	90.7407	2.4424	89.2423	4.8786	89.3075	5.1487	99.3913	1.3611
spect	69.0566	10.0373	69.434	3.6293	68.6792	5.7539	69.434	2.7986	72.4528	7.3802	68.3019	3.3752	70.1887	6.4538	66.0377	9.5278
spectf	79.7205	20.7374	79.7205	20.7374	79.7205	20.7374	78.9727	20.4475	79.7345	14.6711	79.7205	20.2598	79.7205	20.2598	81.9846	16.8489
statlog_heart	81.4815	3.7037	82.2222	5.3351	81.8519	0.8282	82.2222	2.8085	84.0741	4.829	81.8519	4.0147	82.2222	4.0572	80.3704	1.0143
Average	81.9207	7.9121	82.0035	7.2233	81.9654	7.6828	81.0605	7.5528	83.2487	7.5703	81.249	8.3682	82.1911	8.992	80.3106	8.9062
Average Ranks	4.5185		4.1852		4.3333		5.6852		3.4074		5.0185		3.5		5.3519	

els, namely IF-BLS, F-BLS, and Wave-RVFL, also exhibit competitive performance with average accuracies exceeding 80%. Furthermore, NF-RVFL-K records the lowest average standard deviation (7.2233), followed by NF-RVFL-C (7.6828), indicating that the neuro-fuzzy variants provide relatively more consistent predictions across diverse binary classification datasets.

Following the statistical analysis described in Section 2.11, the average ranks of all competing methods are also reported in Table 9.1. IF-BLS achieves the lowest average rank (3.4074), closely followed by RVFL-X-A (3.5000), whereas NF-RVFL-K (4.1852), NF-RVFL-C (4.3333), and NF-RVFL-R (4.5185) also obtain competitive rankings across the 27 datasets. The relatively small differences among the average ranks indicate that the proposed models consistently perform well over a broad range of binary classification problems, while the variations in ranking reflect the complementary strengths of the different randomized neural network architectures.

To examine whether the observed performance differences among the proposed models

are statistically significant, the Friedman test is performed considering $M = 8$ models evaluated on $D = 27$ datasets. The obtained statistics are $\chi_F^2 = 21.2413$ and $F_F = 3.2921$. At the 5% significance level, the corresponding critical value is $F(7, 182) = 2.06$. Since $3.2921 > 2.06$, the null hypothesis is rejected, indicating that statistically significant differences exist among the performances of the proposed randomized neural network models.

Subsequently, the Nemenyi post hoc test is employed for pairwise comparisons with a critical difference of 2.0207 at $\alpha = 0.05$. The critical difference analysis shows that several proposed models belong to statistically overlapping groups, suggesting that their performance differences are not significant despite variations in average accuracy and rank. Overall, the comprehensive evaluation demonstrates that the proposed randomized neural network models consistently achieve strong predictive performance on binary classification tasks, with different architectures exhibiting complementary advantages in terms of accuracy, stability, and overall ranking across the benchmark datasets.

9.2.2 Discussion for Multiclass Datasets

Table 9.2 presents the classification performance of the proposed randomized neural network models, namely NF-RVFL-R, NF-RVFL-K, NF-RVFL-C, RVFL-X-N, RVFL-X-A, edRVFL-FIS-R, edRVFL-FIS-K, and edRVFL-FIS-C, on 30 multiclass UCI benchmark datasets. Overall, all proposed models exhibit competitive performance, demonstrating their ability to effectively handle multiclass classification problems with varying degrees of complexity. Among the competing methods, edRVFL-FIS-R achieves the highest average classification accuracy of 67.8729%, followed by edRVFL-FIS-K (66.9677%), edRVFL-FIS-C (67.1066%), RVFL-X-A (64.7611%), and RVFL-X-N (64.1747%). The NF-RVFL variants also maintain competitive performance, with average accuracies ranging from 63.6038% to 63.9770%. In terms of prediction stability, NF-RVFL-C records the lowest average stan-

Table 9.2: Results on 30 multiclass UCI datasets.

	NF-RVFL-R		NF-RVFL-K		NF-RVFL-C		RVFL-X-N		RVFL-X-A		edRVFL-FIS-R		edRVFL-FIS-K		edRVFL-FIS-C	
Dataset	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.
audiology_std	71.3846	11.1849	70.3718	7.2987	67.8077	6.8874	71.359	11.4941	70.3462	13.0009	72.3846	15.5048	72.9231	10.2782	68.2949	11.1114
balance_scale	98.4	0.8	98.24	1.0431	98.24	1.3145	98.72	1.4533	98.56	0.6693	98.56	1.1866	98.56	1.0431	98.88	1.2133
breast_tissue	26.6234	22.287	25.671	26.4705	25.671	28.7299	19.9567	12.393	21.8615	22.0195	25.671	20.4265	21.8615	18.059	21.8615	17.0913
car	70.0139	12.932	70.8337	6.0725	71.0626	7.5582	73.2048	7.5928	72.1642	6.1585	72.5083	8.3953	72.0446	8.8089	70.0139	12.932
cardiotocography_10clases	64.8151	6.9018	62.8425	7.7672	69.2395	5.9656	71.6383	5.8385	71.3568	4.3991	70.9328	4.2722	69.945	3.3651	71.1205	3.3795
cardiotocography_3clases	84.6213	8.6027	84.2003	8.8942	85.4674	7.7637	86.3151	7.8296	86.2216	6.7854	85.8921	7.5537	86.8781	6.7161	86.8325	7.1237
chess_krvk	3.9814	3.4391	4.1739	4.9529	3.0939	2.962	1.9212	4.296	1.6824	3.762	81.1355	5.9144	81.4174	6.8466	84.8274	5.9368
contrac	41.4934	10.5951	42.3673	5.9684	41.5416	8.5589	41.6089	11.3103	45.6813	13.2705	41.3324	11.5592	41.8727	12.7452	42.0138	9.7884
dermatology	97.5454	1.4855	97.8193	1.5449	97.5379	1.1506	97.8119	1.5643	97.5417	1.7847	97.8119	1.5643	97.8119	1.5643	98.0859	1.2273
ecoli	60.619	35.7146	60.6277	35.2505	62.5405	25.3608	60.9175	35.4923	62.7085	36.4141	61.5101	36.4937	62.0983	37.1682	61.216	35.8605
energy_y1	87.979	2.9121	88.5434	4.2984	89.0595	2.9033	91.7927	4.6162	90.7555	2.3133	91.9226	2.6042	92.3173	4.7221	92.3164	3.0664
energy_y2	89.3269	3.5945	89.1928	5.6606	89.8404	5.2106	91.9243	3.9793	92.0567	3.5483	92.3164	4.6389	92.1857	3.6266	92.5787	4.1547
flags	55.2687	5.1542	55.1957	8.5814	55.2092	9.0911	54.6964	12.1725	54.143	8.829	55.7355	11.999	55.7085	10.3004	58.2861	8.5719
glass	42.3477	23.8658	42.8239	23.0515	40.4873	22.8965	39.546	24.0346	44.1971	29.0731	45.6035	27.9923	44.2082	29.1019	42.3367	27.2694
hayes_roth	61.875	8.0889	64.075	6.313	65	7.7812	65	8.0889	65	8.0889	65.625	8.5582	62.5	10.3645	62.5	5.8463
heart_cleveland	61.0383	6.5277	61.3388	6.9125	60.0437	3.498	61.0383	5.5599	61.6721	6.5652	60.3661	5.5995	61.6885	4.9911	59.7104	6.11
heart_switzerland	47.9	5.2143	46.5333	11.6353	49.7	14.9918	48.8333	3.6931	47.3667	13.4208	49.7	9.897	47.3667	12.1143	47.3333	10.8404
heart_va	41	1.3693	42.5	5.5902	41.5	5.7554	41	6.0208	41.5	4.5415	43	6.2249	42.5	7.7055	41	3.3541
iris	73.3333	22.2361	74.6667	19.5221	74.6667	16.7664	79.3333	17.224	85.3333	15.741	80	20.4124	76	17.062	76	17.5436
led_display	72.7	1.6808	72.2	1.6047	75.6	3.0707	73.6	2.5593	73.5	2.7157	72.9	2.4341	73.2	2.3345	73.1	2.0433
lymphography	85.7701	6.656	87.1034	4.6138	88.5057	3.9375	88.4368	5.3591	87.7931	4.0217	87.1264	5.6022	87.0805	7.5367	86.4138	7.3592
pittsburg_bridges_MATERIAL	77.6623	20.6449	75.9307	30.0712	76.9697	34.9329	75.1948	41.3866	75.1948	41.3866	79.7403	31.8328	75.1948	41.3866	79.7403	31.8328
pittsburg_bridges_REL_L	67.2381	14.0482	69.0476	15.0735	62.1905	14.2921	65.1905	11.5377	68.0952	15.882	72.9524	12.3231	69	4.967	65.0952	14.9257
pittsburg_bridges_TYPE	45.7143	19.518	44.7619	28.6902	44.7619	28.2923	42.8571	33.1628	44.7619	32.0501	47.619	26.5132	41.9048	31.6586	47.619	30.8607
primary_tumor	16.3636	18.7536	15.7576	23.1427	15.4545	22.2784	13.9394	18.4139	14.5455	20.1179	15.4545	20.1407	14.5455	20.4574	15.4545	21.3846
statlog_vehicle	79.1974	1.5671	79.5475	2.82	82.2666	3.0062	83.2175	2.8048	83.3317	2.8607	82.2694	2.8071	82.7421	3.1987	83.5698	2.558
teaching	70.7312	22.8479	70.043	19.0691	70.7097	24.7027	72.0215	23.5513	72.6882	23.5269	71.3763	22.7337	72.043	21.4754	72.043	21.2151
wine_quality_red	60.5106	7.9677	60.8533	3.9867	60.5388	2.5338	60.3495	3.2684	60.2263	4.6251	60.7884	3.9264	60.4142	4.1092	60.6652	3.7188
yeast	55.6593	6.2208	56.0005	3.8753	56.8064	4.0442	57.8171	4.3547	57.5475	4.2804	57.9536	5.0034	58.0198	4.5035	58.2892	3.1668
zoo	97	4.4721	95	6.1237	97.6	7.3082	96	6.5192	95	5	96	5.4772	95	5	96	6.5192
Average	63.6038	10.5761	63.6088	11.1966	63.9704	11.1182	64.1747	11.2524	64.7611	11.8951	67.8729	11.653	66.9677	11.7737	67.1066	11.2668
Average Rank	5.6333		5.3167		5.0333		4.65		4.3167		3.1833		3.9667		3.9	

dard deviation (11.1182), closely followed by RVFL-X-N (11.2524), indicating consistent predictive performance across diverse multiclass datasets.

Following the statistical analysis described in Section 2.10, the average ranks of all competing methods are also reported in Table 9.2. The edRVFL-FIS variants consistently occupy the top positions, with edRVFL-FIS-R attaining the lowest average rank (3.1833), followed by edRVFL-FIS-C (3.9000) and edRVFL-FIS-K (3.9667). Among the remaining methods, RVFL-X-A achieves an average rank of 4.3167, followed by RVFL-X-N (4.6500), whereas NF-RVFL-C (5.0333), NF-RVFL-K (5.3167), and NF-RVFL-R (5.6333) also demonstrate competitive rankings across the 30 multiclass datasets. The relatively close average ranks indicate that the proposed models perform consistently across a wide range of multiclass classification tasks, while the differences in ranking reflect the complementary characteristics of the various randomized neural network architectures.

To investigate whether the observed differences among the proposed models are statis-

Table 9.3: Results on 20 KEEL datasets.

KEEL	GE-IFRVFL-CIL-1	GE-IFRVFL-CIL-2	NF-RVFL-R	NF-RVFL-K	NF-RVFL-C	IF-dRVFL	IF-edRVFL	edRVFL-FIS-R	edRVFL-FIS-K	edRVFL-FIS-C
Dataset	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
abalone9-18	90.45	89.54	95.9	96.31	96.04	96.04	96.04	95.9	95.9	95.9
brwisconsin	98.39	98.79	90.63	91.07	90.34	90.93	90.2	92.68	92.39	92.24
cmc	63.88	68.45	64.71	64.16	63.89	63.97	64.13	64.23	64.1	63.76
crossplane130	100	100	39.23	39.23	39.23	60.77	51.54	82.31	82.31	50
crossplane150	98.22	100	62	62	62	64.67	62	62	62	67.33
ecoli-0-1-4-6_vs_5	100	92.97	98.93	98.93	98.57	98.57	97.14	98.57	98.77	98.75
ecoli-0-1-4-7_vs_2-3-5-6	97.41	95.86	97.62	98.21	97.62	97.18	97.32	97.61	97.91	97.91
ecoli-0-1-4-7_vs_5-6	96.13	100	98.49	98.5	98.5	98.7	98.8	98.5	98.2	98.49
ecoli-0-3-4-7_vs_5-6	94.13	94.13	98.83	98.83	98.83	98.05	98.56	98.83	98.83	98.83
ecoli-0-4-6_vs_5	96.95	83.84	98.51	98.51	98.51	97.52	98.17	98.17	97.52	98.51
ecoli-0-6-7_vs_3-5	97.26	96.28	97.77	97.76	98.21	97.3	98.9	98.9	97.3	97.76
ecoli-0-6-7_vs_5	89.54	83.58	98.18	98.64	98.18	98.18	98.64	97.73	97.73	97.73
glass2	70.3	91.57	92.05	92.05	92.05	92.05	92.05	92.05	92.05	92.05
glass4	92.06	92.06	96.74	96.72	96.26	97.67	96.26	98.6	99.07	99.07
glass5	58.09	95.51	97.19	98.13	97.19	98.14	98.59	98.59	98.13	98.13
haber	74.43	72.26	74.15	74.15	74.15	74.15	74.15	74.15	74.15	74.43
haberman	74.43	72.26	74.15	74.15	74.15	74.15	73.82	73.49	73.49	73.49
heart-stat	84.8	88.28	81.85	81.85	82.22	80.74	82.76	81.11	81.11	81.11
led7digit-0-2-4-5-6-7-8-9_vs_1	96.2	94.73	96.62	96.85	96.62	95.49	96.58	96.62	96.85	97.19
monk1	49.85	50.69	53.04	52.86	52.69	51.43	53.26	51.13	52.15	53.04
Average	86.13	88.04	85.33	85.45	85.26	86.28	85.94	87.56	87.5	86.29

tically significant, the Friedman test is conducted considering $M = 8$ models evaluated on $D = 30$ datasets. The obtained statistics are $\chi_F^2 = 23.3499$ and $F_F = 3.6279$. At the 5% significance level, the corresponding critical value is $F(7, 203) = 2.05$. Since $3.6279 > 2.05$, the null hypothesis is rejected, indicating that statistically significant performance differences exist among the proposed randomized neural network models.

Subsequently, the Nemenyi post hoc test is employed for pairwise comparisons with a critical difference of 1.9170 at $\alpha = 0.05$. The critical difference analysis reveals that several proposed models belong to statistically overlapping groups, indicating that many of the observed differences in average accuracy are not statistically significant. Overall, the experimental results demonstrate that all proposed randomized neural network models provide competitive multiclass classification performance, with the edRVFL-FIS variants exhibiting stronger average predictive capability and ranking, while the NF-RVFL and RVFL-X variants continue to deliver stable and reliable performance across a diverse collection of multiclass benchmark datasets.

9.2.3 Discussion on KEEL Datasets

Table 9.3 presents the classification accuracies of the proposed randomized neural network models on 20 KEEL benchmark datasets. These datasets are highly imbalanced and therefore provide a challenging benchmark for evaluating the robustness and discriminative capability of the developed models. Overall, all methods achieve competitive performance, with average classification accuracies ranging from 85.26% to 88.04%, demonstrating the effectiveness of the proposed architectures in handling imbalanced learning problems. Among all competing methods, GE-IFRVFL-CIL-2 achieves the highest average classification accuracy of 88.04%, followed by edRVFL-FIS-R (87.56%) and edRVFL-FIS-K (87.50%). GE-IFRVFL-CIL-1 also delivers competitive performance with an average accuracy of 86.13%, outperforming several recently developed randomized neural network variants. The superior performance of the GE-IFRVFL-CIL models highlights the effectiveness of integrating geometry-aware representations with class imbalance learning, enabling improved discrimination between majority and minority classes.

An examination of the individual datasets further demonstrates the robustness of the GE-IFRVFL-CIL framework. GE-IFRVFL-CIL-2 consistently achieves the highest or among the highest accuracies on several benchmark datasets, including *crossplane150*, multiple *ecoli* variants, *glass4*, and *led7digit-0-2-4-5-6-7-8-9_vs_1*, while maintaining competitive performance on the remaining datasets. On relatively simpler datasets such as *crossplane130* and *ecoli-0-1-4-6_vs_5*, most models attain near-perfect classification accuracies, whereas more challenging datasets such as *cmc*, *glass2*, *haberman*, and *monk1* exhibit greater variability among the competing architectures. Even on these difficult datasets, the GE-IFRVFL-CIL variants remain consistently competitive, reflecting their ability to effectively learn from highly skewed data distributions.

The remaining proposed models also exhibit strong predictive capability across the

Table 9.4: Accuracies of the proposed GB-RVFL models on 10 large NDC datasets.

Model →	**GB-RVFLwoDL**	**GB-RVFL**	**GE-GB-RVFL**
Dataset ↓	**ACC**	**ACC**	**ACC**
NDC-50K	82.1687	84.4884	**85.3785**
NDC-100K	88.2494	88.4044	**83.8742**
NDC-500K	80.7758	83.2042	**85.3393**
NDC-1M	83.1739	85.2399	**86.1404**
NDC-3M	80.7375	82.6701	**83.9588**
NDC-5M	83.9787	84.8743	**85.3805**
NDC-10M	86.1019	**86.6179**	85.7042
NDC-30M	82.6621	83.4257	**83.4441**
NDC-50M	80.9622	83.5944	**83.7941**
NDC-100M	80.9646	84.4071	**85.0702**
Average ACC	82.9775	84.6926	**84.8084**

KEEL benchmark suite. The edRVFL-FIS variants consistently achieve high accuracies and closely follow the GE-IFRVFL-CIL models in terms of average performance, while the NF-RVFL and IF-based variants demonstrate stable and reliable behavior across datasets with varying imbalance characteristics. Overall, the results indicate that the proposed randomized neural network architectures complement one another, with the GE-IFRVFL-CIL framework providing the strongest overall performance on imbalanced classification problems and the remaining models offering competitive alternatives across diverse data distributions.

9.2.4 Discussion on Large-Scale NDC Datasets

Table 9.4 presents the classification accuracies of the proposed GB-RVFL family, comprising GB-RVFLwoDL, GB-RVFL, and GE-GB-RVFL, on 10 large-scale NDC datasets. These datasets range from 50 thousand to 100 million instances, providing a comprehensive assessment of the scalability and predictive capability of the proposed broad randomized

neural network architectures. Overall, all three models maintain stable performance as the dataset size increases, demonstrating that the proposed broad learning framework scales effectively to large-volume classification tasks.

Among the competing variants, GE-GB-RVFL achieves the highest average classification accuracy of 84.8084%, followed closely by GB-RVFL with an average accuracy of 84.6926%, whereas GB-RVFLwoDL attains an average accuracy of 82.9775%. The geometry-enhanced variant produces the best classification performance on seven of the ten benchmark datasets, including NDC-50K, NDC-500K, NDC-1M, NDC-3M, NDC-5M, NDC-50M, and NDC-100M, illustrating the effectiveness of incorporating geometric information into the broad randomized learning framework. In contrast, GB-RVFL achieves the highest accuracy on NDC-100K and NDC-10M, while both methods exhibit nearly identical performance on the remaining datasets. These observations indicate that the geometric enhancement consistently improves predictive performance without compromising scalability.

A comparison of the three architectures further highlights the contribution of the proposed components. The inclusion of deep latent representations in GB-RVFL results in a noticeable improvement over GB-RVFLwoDL across all benchmark datasets, confirming the benefit of deep feature extraction within the broad learning framework. Furthermore, the additional geometric representation employed in GE-GB-RVFL provides further performance gains on the majority of datasets, particularly as the dataset size increases. Overall, the results demonstrate that the proposed GB-RVFL family provides an effective and scalable solution for large-scale classification, with GE-GB-RVFL offering the most consistent and accurate performance across diverse NDC datasets.

Table 9.5: AUC, Sensitivity, Specificity, Precision, F-Measure, and G-Mean for the GE-IFRVFL-CIL models on 56 KEEL datasets.

	GE-IFRVFL-CIL-1						GE-IFRVFL-CIL-2					
Dataset	AUC	Sensitivity	Specificity	Precision	F-Measure	G-Mean	AUC	Sensitivity	Specificity	Precision	F-Measure	G-Mean
abalone9-18	0.8319	0.75	0.9139	0.3333	0.4615	0.5	0.8272	0.75	0.9043	0.3103	0.439	0.4825
aus	0.8449	0.8723	0.8174	0.7961	0.8325	0.8334	0.8545	0.883	0.8261	0.8058	0.8426	0.8435
brwisconsin	0.9876	1	0.9752	0.9184	0.9574	0.9583	0.9907	1	0.9814	0.9375	0.9677	0.9682
checkerboard_Data	0.8449	0.8723	0.8174	0.7961	0.8325	0.8334	0.8545	0.883	0.8261	0.8058	0.8426	0.8435
cleve	0.8189	0.775	0.8627	0.8158	0.7949	0.7951	0.7797	0.775	0.7843	0.7381	0.7561	0.7563
cmc	0.6262	0.712	0.5405	0.524	0.6037	0.6108	0.6864	0.6739	0.6988	0.6139	0.6425	0.6432
crossplane130	1	1	1	1	1	1	1	1	1	1	1	1
crossplane150	0.9821	0.9643	1	1	0.9818	0.982	1	1	1	1	1	1
ecoli-0-1_vs_2-3-5	0.7143	0.4286	1	1	0.6	0.6547	0.6429	0.2857	1	1	0.4444	0.5345
ecoli-0-1_vs_5	0.8259	0.6667	0.9851	0.8	0.7273	0.7303	0.8259	0.6667	0.9851	0.8	0.7273	0.7303
ecoli-0-1-4-6_vs_5	1	1	1	1	1	1	1	1	1	1	1	1
ecoli-0-1-4-7_vs_2-3-5-6	0.85	0.7	1	1	0.8235	0.8367	0.8	0.6	1	1	0.75	0.7746
ecoli-0-1-4-7_vs_5-6	0.8642	0.75	0.9785	0.75	0.75	0.75	0.8696	0.75	0.9892	0.8571	0.8	0.8018
ecoli-0-2-3-4_vs_5	0.9825	1	0.9649	0.7143	0.8333	0.8452	0.9649	1	0.9298	0.5556	0.7143	0.7454
ecoli-0-2-6-7_vs_3-5	0.9924	1	0.9848	0.75	0.8571	0.866	0.9773	1	0.9545	0.5	0.6667	0.7071
ecoli-0-3-4-6_vs_5	0.8333	0.6667	1	1	0.8	0.8165	0.825	0.6667	0.9833	0.6667	0.6667	0.6667
ecoli-0-3-4-7_vs_5-6	0.8783	0.8	0.9565	0.7273	0.7619	0.7628	0.8783	0.8	0.9565	0.7273	0.7619	0.7628
ecoli-0-4-6_vs_5	0.9195	0.8571	0.9818	0.8571	0.8571	0.8571	0.8468	0.8571	0.8364	0.4	0.5455	0.5855
ecoli-0-6-7_vs_3-5	0.9292	0.875	0.9833	0.875	0.875	0.875	0.8125	0.625	1	1	0.7692	0.7906
ecoli-0-6-7_vs_5	0.8675	0.8333	0.9016	0.4545	0.5882	0.6155	0.8347	0.8333	0.8361	0.3333	0.4762	0.527
ecoli0137vs26	0.9578	0.9412	0.9744	0.8889	0.9143	0.9147	0.9706	0.9412	1	1	0.9697	0.9701
ecoli2	0.9271	0.9091	0.9451	0.6667	0.7692	0.7785	0.8996	0.9091	0.8901	0.5	0.6452	0.6742
ecoli3	0.9271	0.9091	0.9451	0.6667	0.7692	0.7785	0.8996	0.9091	0.8901	0.5	0.6452	0.6742
ecoli4	0.8845	0.8	0.9691	0.5714	0.6667	0.6761	0.8845	0.8	0.9691	0.5714	0.6667	0.6761
glass2	0.8387	1	0.6774	0.1667	0.2857	0.4082	0.7258	0.5	0.9516	0.4	0.4444	0.4472
glass4	0.5836	0.2	0.9672	0.3333	0.25	0.2582	0.5836	0.2	0.9672	0.3333	0.25	0.2582
glass5	0.7813	1	0.5625	0.0667	0.125	0.2582	0.9766	1	0.9531	0.4	0.5714	0.6325
haber	0.6085	0.32	0.8971	0.5333	0.4	0.4131	0.5938	0.32	0.8676	0.4706	0.381	0.3881
haberman	0.6085	0.32	0.8971	0.5333	0.4	0.4131	0.5938	0.32	0.8676	0.4706	0.381	0.3881
heart-stat	0.8505	0.8286	0.8723	0.8286	0.8286	0.8286	0.886	0.8571	0.9149	0.8824	0.8696	0.8697
iono	0.8792	0.8	0.9583	0.9032	0.8485	0.85	0.7812	0.7429	0.8194	0.6667	0.7027	0.7037
led7digit-0-2-4-5-6-7-8-9_vs_1	0.9339	0.9	0.9677	0.6923	0.7826	0.7894	0.9258	0.9	0.9516	0.6	0.72	0.7348
monk1	0.5	0.5714	0.4286	0.5	0.5333	0.5345	0.5	0.1786	0.8214	0.5	0.2632	0.2988
monk2	0.8059	0.7813	0.8305	0.7143	0.7463	0.747	0.7778	0.8438	0.7119	0.6136	0.7105	0.7196
monk3	0.5	0.8333	0.1667	0.5	0.625	0.6455	0.5	1	0	0.5	0.6667	0.7071
new-thyroid1	1	1	1	1	1	1	1	1	1	1	1	1
pima	0.8035	0.8228	0.7843	0.6633	0.7345	0.7387	0.787	0.8354	0.7386	0.6226	0.7135	0.7212
ripley	0.9128	0.8796	0.9459	0.9438	0.9106	0.9111	0.9154	0.8848	0.9459	0.9441	0.9135	0.914
shuttle-6_vs_2-3	1	1	1	1	1	1	1	1	1	1	1	1
shuttle-c0-vs-c4	0.9865	0.973	1	1	0.9863	0.9864	1	1	1	1	1	1
sonar	0.7507	0.7586	0.7429	0.7097	0.7333	0.7337	0.7419	0.6552	0.8286	0.76	0.7037	0.7056
transfusion	0.6245	0.3864	0.8626	0.4048	0.3953	0.3955	0.6672	0.4773	0.8571	0.4468	0.4615	0.4618
vehicle1	0.8339	0.9041	0.7637	0.6055	0.7253	0.7399	0.7997	0.8082	0.7912	0.6082	0.6941	0.7011
vehicle2	0.9893	0.9839	0.9948	0.9839	0.9839	0.9839	0.9706	0.9516	0.9896	0.9672	0.9593	0.9594
votes	0.9623	0.963	0.9615	0.9455	0.9541	0.9542	0.9558	0.963	0.9487	0.9286	0.9455	0.9456
vowel	0.8889	0.7778	1	1	0.875	0.8819	0.8704	0.7407	1	1	0.8511	0.8607
wpbc	0.6252	0.4545	0.7959	0.3333	0.3846	0.3892	0.6558	0.4545	0.8571	0.4167	0.4348	0.4352
yeast-0-2-5-6_vs_3-7-8-9	0.7544	0.6	0.9087	0.5	0.5455	0.5477	0.7416	0.525	0.9582	0.6563	0.5833	0.587
yeast-0-2-5-7-9_vs_3-6-8	0.9104	0.8462	0.9747	0.7586	0.8	0.8012	0.8912	0.8077	0.9747	0.75	0.7778	0.7783
yeast-0-3-5-9_vs_7-8	0.7	0.7778	0.6222	0.2154	0.3373	0.4093	0.5981	0.3889	0.8074	0.2121	0.2745	0.2872
yeast-0-5-6-7-9_vs_4	0.7067	0.5882	0.8252	0.2857	0.3846	0.41	0.7775	0.7647	0.7902	0.3023	0.4333	0.4808
yeast-2_vs_4	0.8581	0.8571	0.8592	0.375	0.5217	0.5669	0.9326	0.9286	0.9366	0.5909	0.7222	0.7407

yeast1	0.7318	0.8051	0.6585	0.515	0.6282	0.6439	0.7463	0.769	0.7236	0.5561	0.6455	0.6539
yeast2vs8	0.743	0.5	0.9859	0.5	0.5	0.5	0.75	0.5	1	1	0.6667	0.7071
yeast3	0.9034	0.8462	0.9607	0.7857	0.8148	0.8154	0.9298	0.9538	0.9058	0.6327	0.7607	0.7768
yeast5	0.9335	0.9167	0.9504	0.5116	0.6567	0.6848	0.8942	0.8333	0.9551	0.5128	0.6349	0.6537
Average	0.8357	0.7907	0.8807	0.6913	0.7099	0.7234	0.8285	0.7627	0.8942	0.6851	0.6942	0.7085

9.2.5 Performance Evaluation on KEEL Imbalanced Datasets based on Multiple Metrics

Table 9.5 provides a comprehensive comparison of the two proposed GE-IFRVFL-CIL variants on 56 KEEL imbalanced benchmark datasets using six widely adopted evaluation metrics, namely AUC, Sensitivity, Specificity, Precision, F-Measure, and G-Mean. Since overall accuracy alone may be misleading for highly imbalanced classification problems, these complementary performance measures provide a more reliable assessment of the models' ability to correctly identify both majority and minority classes. Overall, both GE-IFRVFL-CIL variants demonstrate strong and balanced predictive performance across the benchmark suite, confirming the effectiveness of the proposed geometry-enhanced class imbalance learning framework.

From the average results, GE-IFRVFL-CIL-1 achieves higher values in terms of AUC (0.8357), Sensitivity (0.7907), Precision (0.6913), and F-Measure (0.7099), indicating its stronger capability to correctly identify minority-class samples while maintaining a favorable balance between precision and recall. In contrast, GE-IFRVFL-CIL-2 attains higher average Specificity (0.8942) and G-Mean (0.7085), demonstrating improved discrimination of majority-class instances while preserving a better balance between class-wise prediction rates. These complementary characteristics suggest that the two variants provide different yet effective trade-offs for handling highly imbalanced learning problems.

A closer inspection of the individual datasets further illustrates the robustness of the proposed framework. On several benchmark datasets, including *crossplane130*, *crossplane150*, *ecoli-0-1-4-6_vs_5*, and *shuttle-0-vs-c4*, both variants achieve perfect or near-perfect scores across multiple evaluation metrics, indicating that the proposed models are capable of accurately separating the underlying classes when sufficient discriminative information is available. Conversely, datasets such as *glass4*, *monk1*, *yeast-0-5-6-7-9-vs-4*, and *yeast-0-3-5-*

9-vs-7-8 remain considerably more challenging, where noticeable reductions are observed across all evaluation metrics. Nevertheless, both variants maintain competitive and well-balanced performance even on these difficult datasets, demonstrating their ability to generalize under severe class imbalance.

Overall, the results indicate that the proposed GE-IFRVFL-CIL framework consistently delivers reliable performance across a diverse collection of imbalanced benchmark datasets. While GE-IFRVFL-CIL-1 exhibits a slight advantage in identifying minority-class samples through higher sensitivity-related measures, GE-IFRVFL-CIL-2 provides improved class balance as reflected by its higher specificity and G-Mean. These complementary characteristics make the proposed framework well suited for real-world imbalanced classification tasks, where different application scenarios may prioritize different evaluation criteria.

9.3 Guidelines for Selecting an Appropriate Randomized Neural Network Framework

The proposed randomized neural network frameworks address different practical challenges encountered in real-world machine learning applications. Consequently, no single model is universally optimal for every problem. Instead, the choice of an appropriate framework should be guided by the characteristics of the target application, including the need for interpretability, robustness, scalability, class imbalance handling, representational capacity, and computational efficiency. This section provides practical guidelines for selecting an appropriate framework based on these characteristics.

9.3.1 Case I: Applications Requiring Explainability

In many high-stakes applications, such as healthcare, finance, and decision support systems, understanding the reasoning behind model predictions is as important as achieving high predictive accuracy. In such scenarios, the NF-RVFL family provides the most suitable solution by integrating fuzzy inference into the RVFL architecture. The generated IF–THEN rules, together with feature importance analysis and partial dependence plots,

provide transparent and human-interpretable explanations while maintaining the computational efficiency of randomized neural networks.

Among the three variants, NF-RVFL-R is suitable when rapid model construction is required, NF-RVFL-K is preferred when compact cluster structures are expected, whereas NF-RVFL-C is more appropriate for datasets containing overlapping decision boundaries.

9.3.2 Case II: Large-Scale Learning Problems

When the primary challenge is the availability of extremely large datasets, computational scalability becomes more important than model complexity. Under such circumstances, the GB-RVFL framework significantly reduces the computational burden by replacing individual samples with granular-ball representations, thereby requiring the inversion of only the granular-ball center matrix instead of the complete training data.

If the intrinsic geometric relationships among data samples are expected to be informative, GE-GB-RVFL should be preferred, as the graph embedding mechanism preserves the local manifold structure while retaining the computational advantages of granular-ball learning.

9.3.3 Case III: Imbalanced Classification Problems

Many practical applications, including disease diagnosis, fraud detection, fault diagnosis, and anomaly detection, suffer from severe class imbalance. Conventional learning algorithms generally bias the majority class and therefore fail to identify minority samples effectively.

For such problems, the GE-IFRVFL-CIL framework is the most appropriate choice. By jointly incorporating class imbalance learning, graph embedding, intuitionistic fuzzy weighting, and discriminative subspace learning, the proposed framework simultaneously improves minority-class recognition while preserving the geometric structure of the data. GE-IFRVFL-CIL-1 is generally preferred when global discriminative structures dominate, whereas GE-IFRVFL-CIL-2 becomes more suitable when preserving local neighborhood relationships is essential.

9.3.4 Case IV: Noisy and Outlier-Contaminated Data

Real-world datasets frequently contain noisy observations, corrupted measurements, or mislabeled samples. Since conventional randomized neural networks treat every sample equally, their performance may deteriorate significantly in such situations.

When robustness against noise and outliers is the primary objective, intuitionistic fuzzy based frameworks provide a more reliable solution by assigning adaptive importance weights according to the membership and non-membership characteristics of individual samples. For broad learning architectures, IF-BLS offers an efficient and robust alternative, whereas IF-dRVFL and IF-edRVFL are preferable when deeper hierarchical representations are also required.

9.3.5 Case V: Complex Nonlinear Data Distributions

Certain learning problems involve highly nonlinear feature interactions that cannot be adequately represented by shallow randomized neural networks. Examples include biomedical signal analysis, high-dimensional biological datasets, and multimodal learning tasks.

For such problems, the edRVFL-FIS family provides enhanced representation learning by integrating fuzzy feature extraction within a deep ensemble architecture. The combination of hierarchical random representations and fuzzy inference enables improved discrimination while simultaneously enhancing interpretability.

9.3.6 Case VI: Rich Feature Representation

When the underlying data contain intricate feature interactions that are difficult to capture in the real-valued domain, RVFL-X provides a more expressive learning framework. By transforming real-valued observations into complex-valued representations before randomized learning, RVFL-X substantially increases the representational capability of the original RVFL architecture without introducing excessive computational complexity. Consequently, RVFL-X is particularly suitable for signal processing, EEG analysis, communication systems, and other applications involving complex latent structures.

9.3.7 Summary of Model Selection

Table 9.6 summarizes the recommended randomized neural network framework under different application scenarios.

Table 9.6: Guidelines for selecting the proposed randomized neural network frameworks according to application characteristics.

Application scenario	Recommended framework	Primary advantage
Interpretable decision making	NF-RVFL-R/K/C	Fuzzy IF–THEN rules with explainable predictions and feature interpretation.
Large-scale datasets	GB-RVFL, GE-GB-RVFL	Granular-ball learning significantly improves computational scalability while preserving predictive performance.
Highly imbalanced datasets	GE-IFRVFL-CIL-1/2	Joint class imbalance learning, graph embedding, and intuitionistic fuzzy weighting improve minority-class recognition.
Noisy or outlier-contaminated datasets	IF-BLS, IF-dRVFL, IF-edRVFL	Adaptive intuitionistic fuzzy weighting reduces the influence of unreliable samples.
Complex nonlinear feature distributions	edRVFL-FIS-R/K/C	Deep ensemble learning combined with fuzzy feature representations improves expressive capability.
Rich latent feature representation	RVFL-X	Complex-valued representations provide higher expressive power than conventional real-valued RVFL.
Fast broad learning	F-BLS	Efficient broad architecture with fuzzy neighborhood modeling.

9.4 Summary

This chapter presented a comprehensive empirical evaluation of all the randomized neural network frameworks proposed throughout this thesis on a diverse collection of benchmark datasets encompassing binary classification, multiclass classification, imbalanced

learning, and large-scale classification problems. The experimental results demonstrated that each proposed framework effectively addresses a specific learning challenge while maintaining the inherent advantages of randomized neural networks, including computational efficiency and strong generalization. The NF-RVFL family exhibited excellent interpretability through fuzzy reasoning, the GB-RVFL variants achieved superior scalability on large-scale datasets, the GE-IFRVFL-CIL framework consistently handled severe class imbalance, RVFL-X enhanced expressive capability via complex-valued representations, the deep and ensemble RVFL variants improved hierarchical feature learning and robustness, and the fuzzy broad learning frameworks effectively mitigated the influence of noise and outliers. Collectively, these results confirm that the proposed methodologies provide a unified yet versatile family of randomized neural network models capable of addressing a broad spectrum of real-world machine learning problems. Furthermore, the practical model selection guidelines presented in this chapter offer a systematic framework for choosing the most appropriate randomized neural network architecture according to the characteristics and requirements of a given application.

Chapter 10

Conclusions and Future Work

This chapter synthesizes the key contributions presented throughout the thesis and provides a comprehensive overview of the advancements achieved in RdNNs. The proposed methodologies have systematically addressed fundamental challenges related to interpretability, robustness, expressivity, and scalability. Building upon these contributions, this chapter first summarizes the major findings and insights, and subsequently outlines promising future research directions that can further extend the scope and applicability of the proposed frameworks.

10.1 Conclusion

Randomized neural networks (RdNNs) have attracted considerable attention owing to their simple architecture, analytical training mechanism, and computational efficiency. However, despite these advantages, their practical applicability has been limited by several fundamental challenges, including limited interpretability, sensitivity to noisy and imbalanced data, insufficient representation capability, and the difficulty of scaling to increasingly complex learning tasks. The primary objective of this thesis was to address these limitations while preserving the computational efficiency that distinguishes randomized neural networks from conventional deep learning models.

To achieve this objective, this thesis developed a unified collection of methodologies that enhance randomized neural networks from multiple complementary perspectives. Rather

than modifying a single component of the learning process, the proposed approaches improve different stages of the randomized learning pipeline, including data representation, feature representation, optimization, uncertainty modeling, and network architecture. Collectively, these contributions transform conventional randomized neural networks into more robust, interpretable, expressive, and scalable learning systems.

A major contribution of this thesis is the incorporation of interpretable reasoning into randomized neural networks through fuzzy inference systems. By integrating IF–THEN rule-based reasoning within shallow, deep, and ensemble randomized architectures, the proposed neuro-fuzzy models enable transparent decision-making while simultaneously enriching feature representations. This demonstrates that interpretability and predictive performance need not be mutually exclusive in randomized neural networks.

To improve robustness, several complementary strategies were proposed that address different sources of uncertainty encountered during learning. Granular-ball computing provides robust data abstraction by reducing the influence of noisy samples and outliers before learning begins, while graph embedding preserves the intrinsic geometric relationships among granular-ball representations during optimization. Furthermore, intuitionistic fuzzy weighting enables adaptive treatment of uncertain and imbalanced samples, whereas the Wave loss function improves optimization robustness through bounded and noise-insensitive learning. Together, these techniques significantly enhance the reliability of randomized neural networks under challenging real-world conditions.

The thesis also extends the representational capability of randomized neural networks by introducing complex-valued learning for real-valued tabular data. Through the proposed RVFL-X framework, complex-valued feature representations are incorporated into randomized learning while preserving the simplicity of the original RVFL architecture. This demonstrates that richer feature representations can be achieved without abandoning the computational advantages of randomized neural networks.

Another important contribution lies in improving scalability while maintaining efficient learning. Granular-ball representations substantially reduce computational complexity by operating on compact data summaries rather than individual samples, whereas analytical optimization is preserved whenever possible. Even in scenarios requiring iterative opti-

mization, such as the Wave-RVFL model, efficient optimization strategies ensure practical applicability to larger datasets. Consequently, the proposed methods preserve one of the principal advantages of randomized neural networks, namely fast and scalable learning.

The proposed methodologies were extensively evaluated on numerous benchmark datasets and real-world biomedical applications, including breast cancer diagnosis and Alzheimer's disease classification. Across these diverse experimental settings, the proposed models consistently demonstrated improvements in predictive performance, robustness, interpretability, and computational efficiency compared with existing randomized neural network models. These experimental results validate the effectiveness of the proposed methodologies and illustrate their potential for deployment in practical machine learning applications.

Overall, this thesis establishes a comprehensive framework for advancing randomized neural networks by integrating fuzzy reasoning, granular computing, graph embedding, intuitionistic fuzzy theory, robust optimization, and complex-valued representation learning within a unified randomized learning paradigm. By simultaneously addressing interpretability, robustness, representation learning, and scalability, the proposed research significantly broadens the applicability of randomized neural networks and provides a strong foundation for the development of trustworthy, efficient, and intelligent learning systems.

10.2 Future Research Directions

Although this thesis substantially advances randomized neural networks, several important research challenges remain. Future work can be broadly organized into the following interconnected directions.

- **Towards Adaptive and Explainable Neuro-Fuzzy Randomized Neural Networks.** Although the proposed NF-RVFL and edRVFL-FIS frameworks successfully integrate fuzzy inference systems into shallow, deep, and ensemble randomized neural networks to improve interpretability while maintaining competitive predictive performance, several research challenges remain. Both frameworks rely on clustering-based fuzzy partitioning to construct the fuzzy rule base, where the quality and number of

fuzzy rules significantly influence interpretability, predictive performance, and computational complexity. Moreover, all input features are currently treated with equal importance during the learning process, which may introduce redundant information and limit the transparency of the extracted fuzzy knowledge, particularly for high-dimensional datasets. While the proposed models provide post hoc feature importance analysis and interpretable IF–THEN reasoning, they do not jointly optimize feature selection, fuzzy rule generation, and network learning within a unified framework. Future research may therefore focus on developing adaptive neuro-fuzzy randomized neural networks that simultaneously perform feature selection, automatic fuzzy rule generation, rule pruning, and parameter optimization to obtain compact, highly interpretable, and computationally efficient models. Furthermore, extending fuzzy reasoning to deeper randomized architectures through hierarchical and adaptive fuzzy representations could further enhance the interpretability of deep and ensemble randomized neural networks. Another promising direction is the incorporation of biologically inspired neuron models, neuromorphic computing principles, and uncertainty-aware fuzzy reasoning to better mimic human cognitive decision-making while preserving the computational efficiency of randomized learning. Such developments would provide a unified explainable neuro-fuzzy framework capable of improving transparency, scalability, robustness, and generalization across a wide range of real-world applications.

- **Towards More Robust and Scalable Randomized Neural Networks.** Although the proposed GB-RVFL, GE-GB-RVFL, and Wave-RVFL models significantly improve the robustness and scalability of randomized neural networks through granular computing, graph embedding, and robust loss optimization, several important research challenges remain. The incorporation of graph embedding enhances the discriminative capability of GE-GB-RVFL by preserving the intrinsic geometric structure of the data; however, graph construction and regularization introduce additional computational overhead, particularly for large-scale datasets. Likewise, although the Wave loss function improves robustness against noise and outliers, its non-convex opti-

mization currently requires iterative gradient-based optimization, thereby sacrificing the analytical closed-form solution that is one of the major advantages of randomized neural networks. Furthermore, the current granular-ball construction strategy is static and primarily designed for supervised learning, which limits its adaptability to evolving data distributions and scenarios with limited labeled data. Future research may therefore focus on developing computationally efficient graph learning strategies, such as sparse or adaptive graph embedding techniques, to preserve geometric information while reducing computational complexity. Another promising direction is the design of novel robust loss functions that retain the desirable noise-insensitive properties of the Wave loss while admitting efficient analytical or semi-analytical optimization schemes. In addition, adaptive and probabilistic granular-ball construction methods, together with self-organizing granularity learning, could further improve both representation quality and computational efficiency. Extending these frameworks to semi-supervised, self-supervised, and unsupervised learning by integrating modern representation learning techniques, such as contrastive learning, would substantially broaden their applicability in data-scarce environments. Finally, incorporating granular computing, graph representation learning, and robust optimization into deep, ensemble, and online randomized neural networks, as well as validating these models on large-scale, high-dimensional, and temporal datasets, represents a promising direction towards the development of next-generation robust and scalable randomized learning systems.

- **Towards Robust Learning for Imbalanced and Uncertain Data.** Although the proposed GE-IFRVFL-CIL models effectively address class imbalance by integrating graph embedding, intuitionistic fuzzy theory, and class weighting within the RVFL framework, several challenges remain. The current models are primarily designed for binary imbalanced classification and rely on predefined class weighting and graph construction strategies, which may not fully capture complex imbalance patterns in highly skewed or large-scale datasets. Furthermore, their applicability to evolving data distributions and scenarios with limited labeled samples has not been explored.

Future research may therefore focus on developing adaptive imbalance-aware randomized neural networks that automatically learn class-specific weights from data. Extending the proposed framework to multiclass, multi-label, semi-supervised, and continual learning settings would further broaden its applicability. Moreover, integrating advanced representation learning techniques, such as contrastive learning and graph neural networks, with intuitionistic fuzzy reasoning could improve the discrimination of minority classes, thereby enabling more reliable learning in challenging real-world imbalanced applications.

- **Towards Advanced Complex-valued Representation Learning for Randomized Neural Networks.** The proposed RVFL-X framework demonstrates the feasibility of extending complex-valued learning to real-valued tabular datasets by constructing informative complex representations. Future research may focus on developing more expressive strategies for generating the imaginary component using advanced representation learning techniques, such as denoising autoencoders, variational autoencoders, sparse autoencoders, contrastive learning, and self-supervised learning frameworks, to obtain richer and more discriminative latent representations. Another important direction is the theoretical investigation of complex-valued representation learning for randomized neural networks, including analyses of expressivity, approximation capability, optimization behavior, and generalization performance, to better understand the advantages of complex-domain learning over conventional real-valued models. Furthermore, extending the proposed transformation framework to other randomized neural network architectures, deep and ensemble learning models, and multimodal learning scenarios could further improve representation learning, robustness, and predictive performance. Such developments would contribute towards establishing a unified framework for complex-valued randomized neural networks applicable to a broad range of real-world machine learning problems.

- **Towards Unified and Adaptive Randomized Neural Network Frameworks.** The proposed models improve different aspects of randomized neural networks through fuzzy reasoning, granular computing, graph embedding, intuitionistic fuzzy theory,

robust loss functions, and complex-valued representations. An important future direction is the development of a unified randomized neural network framework capable of integrating these complementary techniques within a single architecture. Such a framework should automatically adapt its architecture, learning strategy, and hyperparameters according to the characteristics of the data. Incorporating ideas from AutoML, meta-learning, and neural architecture search could significantly improve the adaptability and usability of randomized neural networks.

- **Theoretical Understanding of Randomized Learning.** Although the proposed models demonstrate consistent empirical improvements, many theoretical questions remain open. Future research should establish rigorous analyses of approximation capability, convergence behavior, optimization properties, generalization bounds, stability, and robustness under noisy and uncertain environments. Developing a comprehensive theoretical foundation would provide deeper insights into the behavior of randomized neural networks and strengthen their applicability in safety-critical domains.

- **Large-scale Biomedical and Real-world Applications.** The encouraging performance obtained on biomedical datasets demonstrates the practical potential of the proposed methodologies. Future work should extend these models to multimodal healthcare applications involving EEG, MRI, CT, electronic health records, genomic data, and wearable sensor information. In parallel, establishing standardized benchmark datasets, evaluation protocols, and reproducible implementations would facilitate fair comparisons and accelerate future research on randomized neural networks across diverse application domains.

Bibliography

[1] 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.

[2] Q. Shi, R. Katuwal, P. N. Suganthan, M. Tanveer, Random vector functional link neural network based ensemble deep learning, Pattern Recognition 117 (2021) 107978.

[3] C. P. Chen, Z. Liu, Broad learning system: An effective and efficient incremental learning system without the need for deep architecture, IEEE Transactions on Neural Networks and Learning Systems 29 (2017) 10–24.

[4] L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (1997) 111–127.

[5] 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 (2007) 40–51.

[6] S. Rezvani, X. Wang, Intuitionistic fuzzy twin support vector machines for imbalanced data, Neurocomputing 507 (2022) 16–25.

[7] D. Dua, C. Graff, UCI machine learning repository, Available: http://archive.ics.uci.edu/ml (2017).

[8] J. Derrac, S. Garcia, L. Sanchez, F. Herrera, KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, J. Mult. Valued Logic Soft Comput 17 (2015).

[9] 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) e00938.

[10] K. C. Luk, J. E. Ball, A. Sharma, An application of artificial neural networks for rainfall forecasting, Mathematical and Computer Modelling 33 (2001) 683–693.

[11] W. G. Baxt, Application of artificial neural networks to clinical medicine, The Lancet 346 (1995) 1135–1138.

[12] R. Dase, D. Pawar, Application of artificial neural network for stock market predictions: A review of literature, International Journal of Machine Intelligence 2 (2010) 14–17.

[13] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks 9 (1998) 987–1000.

[14] M. Tanveer, M. Ganaie, I. Beheshti, T. Goel, N. Ahmad, K.-T. Lai, K. Huang, Y.-D. Zhang, J. Del Ser, C.-T. Lin, Deep learning for brain age estimation: A systematic review, Information Fusion (2023).

[15] F. Adolfi, J. S. Bowers, D. Poeppel, Successes and critical failures of neural networks in capturing human-like speech recognition, Neural Networks 162 (2023) 199–211.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[17] 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 32 (2024) 2738–2749.

[18] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 76–86.

[19] W. F. Schmidt, M. A. Kraaijveld, R. P. Duin, Feed forward neural networks with random weights, in: International Conference on Pattern Recognition, IEEE Computer Society Press, 1992, pp. 1–1.

[20] P. N. Suganthan, R. Katuwal, On the origins of randomization-based feedforward neural networks, Applied Soft Computing 105 (2021) 107239.

[21] 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.

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

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

[24] M. J. Kearns, U. Vazirani, An introduction to computational learning theory, MIT press, 1994.

[25] 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 143 (2023) 110377.

[26] 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). doi:10.1109/TCBB.2022.3170351.

[27] 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.

[28] P.-B. Zhang, Z.-X. Yang, A new learning paradigm for random vector functional-link network: RVFL+, Neural Networks 122 (2020) 94–105.

[29] J. Li, C. Hua, Y. Yang, X. Guan, Bayesian block structure sparse based T-S fuzzy modeling for dynamic prediction of hot metal silicon content in the blast furnace, IEEE Transactions on Industrial Electronics 65 (2017) 4933–4942.

[30] 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 34 (2022) 6847–6859.

[31] T. Chakravorti, P. Satyanarayana, Non linear system identification using kernel based exponentially extended random vector functional link network, Applied Soft Computing 89 (2020) 106117.

[32] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, A. D. N. I. 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) 1–12.

[33] M. Ha, C. Wang, J. Chen, The support vector machine based on intuitionistic fuzzy number and kernel function, Soft Computing 17 (2013) 635–641.

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

[35] 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.

[36] 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.

[37] 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), 2020, pp. 42–48. doi:10.1109/BigDataSecurity-HPSC-IDS49724.2020.00019.

[38] 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.

[39] M. A. Ganaie, M. Sajid, A. K. Malik, M. Tanveer, Graph embedded intuitionistic fuzzy random vector functional link neural network for class imbalance learning, IEEE Transactions on Neural Networks and Learning Systems 35 (2024) 11671–11680.

[40] U. Mishra, D. Gupta, B. B. Hazarika, An efficient angle-based twin random vector functional link classifier, Applied Soft Computing 164 (2024) 112005.

[41] Q. Wang, Y. Ma, K. Zhao, Y. Tian, A comprehensive survey of loss functions in machine learning, Annals of Data Science 9 (2020) 187–212.

[42] K. Wang, H. Pei, J. Cao, P. Zhong, Robust regularized extreme learning machine for regression with non-convex loss function via DC program, Journal of the Franklin Institute 357 (2020) 7069–7091.

[43] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, Y. Luo, Granular ball computing classifiers for efficient, scalable and robust learning, Information Sciences 483 (2019) 136–152.

[44] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, Z. Chen, Ball k-means: Fast adaptive clustering with no bounds, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 87–99.

[45] S. Xia, S. Zheng, G. Wang, X. Gao, B. Wang, Granular ball sampling for noisy label classification or imbalanced classification, IEEE Transactions on Neural Networks and Learning Systems 34 (2023) 2144–2155.

[46] S. Xia, X. Lian, G. Wang, X. Gao, J. Chen, X. Peng, Gbsvm: an efficient and robust support vector machine framework via granular-ball computing, IEEE Transactions on Neural Networks and Learning Systems 36 (2024) 9253–9267.

[47] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, A. Graves, Associative long short-term memory, in: International Conference on Machine Learning, PMLR, 2016, pp. 1986–1994.

[48] S. Yadav, K. R. Jerripothula, FCCNs: Fully complex-valued convolutional networks using complex-valued color model and loss function, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 10689–10698.

[49] T. A. Plate, Holographic Reduced Representation: Distributed representation for cognitive structures, volume 150, CSLI Publications Stanford, 2003.

[50] D. P. Reichert, T. Serre, Neuronal synchrony in complex-valued deep networks, arXiv preprint arXiv:1312.6115 (2013).

[51] G. Shi, M. M. Shanechi, P. Aarabi, On the importance of phase in human speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 14 (2006) 1867–1874.

[52] A. V. Oppenheim, J. S. Lim, The importance of phase in signals, Proceedings of the IEEE 69 (1981) 529–541.

[53] C. P. Chen, Z. Liu, S. Feng, Universal approximation capability of broad learning system and its structural variations, IEEE Transactions on Neural Networks and Learning Systems 30 (2018) 1191–1204.

[54] W. Yu, C. Zhao, Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability, IEEE Transactions on Industrial Electronics 67 (2019) 5081–5091.

[55] W. Chen, K. Yang, W. Zhang, Y. Shi, Z. Yu, Double-kernelized weighted broad learning system for imbalanced data, Neural Computing and Applications 34 (2022) 19923–19936.

[56] S. Feng, C. P. Chen, Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification, IEEE Transactions on Cybernetics 50 (2018) 414–424.

[57] H. Su, Z. Li, W. Liu, J. Chen, A novel broad learning network with residual technique, Pattern Recognition (2025) 112134.

[58] A. Smiti, A critical overview of outlier detection methods, Computer Science Review 38 (2020) 100306.

[59] 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.

[60] R. Gao, R. Li, M. Hu, P. Suganthan, K. F. Yuen, Online dynamic ensemble deep random vector functional link neural network for forecasting, Neural Networks 166 (2023) 51–69.

[61] R. Li, R. Gao, P. N. Suganthan, J. Cui, O. Sourina, L. Wang, A spectral-ensemble deep random vector functional link network for passive brain–computer interface, Expert Systems with Applications 227 (2023) 120279.

[62] 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 27 (2023) 4995–5003. doi:10.1109/JBHI.2022.3215533.

[63] R. Sharma, T. Goel, M. Tanveer, M. Al-Dhaifallah, Alzheimer's disease diagnosis using ensemble of random weighted features and fuzzy least square twin support vector machine, IEEE Transactions on Emerging Topics in Computational Intelligence 9 (2025) 1281–1291.

[64] R. Cheng, R. Gao, K. F. Yuen, Ship order book forecasting by an ensemble deep parsimonious random vector functional link network, Engineering Applications of Artificial Intelligence 133 (2024) 108139.

[65] J. He, X. Li, P. Liu, L. Wang, H. Zhou, J. Wang, R. Tang, Ensemble deep random vector functional link for self-supervised direction-of-arrival estimation, Engineering Applications of Artificial Intelligence 120 (2023) 105831.

[66] M. Sajid, M. Tanveer, P. N. Suganthan, Ensemble deep random vector functional link neural network based on fuzzy inference system, IEEE Transactions on Fuzzy Systems 33 (2025) 479–490.

[67] R. Gao, M. Hu, R. Li, X. Luo, P. N. Suganthan, M. Tanveer, Stacked ensemble deep random vector functional link network with residual learning for medium-scale time-series forecasting, IEEE Transactions on Neural Networks and Learning Systems 36 (2025) 10833–10843.

[68] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information fusion 58 (2020) 82–115.

[69] A. Quadir, M. Sajid, M. Tanveer, Granular ball twin support vector machine, IEEE Transactions on Neural Networks and Learning Systems 36 (2024) 12444–12453.

[70] M. Sajid, A. Quadir, M. Tanveer, GB-RVFL: Fusion of randomized neural network and granular ball computing, Pattern Recognition 159 (2025) 111142.

[71] M. Akhtar, M. Tanveer, M. Arshad, Alzheimer's Disease Neuroimaging Initiative, Advancing supervised learning with the wave loss function: A robust and smooth approach, Pattern Recognition 155 (2024) 110637.

[72] M. Sajid, A. Quadir, M. Tanveer, Wave-rvfl: A randomized neural network based on wave loss function, in: International Conference on Neural Information Processing, Springer, 2024, pp. 242–257.

[73] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[74] M. Sajid, M. Akhtar, A. Quadir, M. Tanveer, Rvfl-x: A novel randomized network based on complex transformed real-valued tabular datasets, in: 2025 International Joint Conference on Neural Networks (IJCNN), 2025, pp. 1–8. doi:10.1109/IJCNN64981.2025.11227691.

[75] 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.

[76] M. Sajid, A. Quadir, A. Rahaman, M. Tanveer, Robust ensemble deep randomized neural networks, [Under Review] (2026).

[77] X. Gong, T. Zhang, C. P. Chen, Z. Liu, Research review for broad learning system: Algorithms, theory, and applications, IEEE Transactions on Cybernetics 52 (2022) 8922–8950.

[78] M. Sajid, A. K. Malik, M. Tanveer, Intuitionistic fuzzy broad learning system: Enhancing robustness against noise and outliers, IEEE Transactions on Fuzzy Systems 32 (2024) 4460–4469.

[79] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.

[80] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[81] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215.

[82] A. B. Arrieta, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[83] C. Zhang, et al., Understanding deep learning (a perspective on generalization and robustness), Proceedings of the IEEE (2021). General perspective article on robustness and generalization.

[84] Y. Xiao, M. Adegoke, C.-S. Leung, K. W. Leung, Robust noise-aware algorithm for randomized neural network and its convergence properties, Neural Networks 173 (2024) 106202. doi:10.1016/j.neunet.2024.106202.

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

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

[87] H. Ren, Q. Xu, et al., A comprehensive evaluation of random vector functional link networks, Information Sciences 367–368 (2016) 1094–1105.

[88] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, Random vector functional link network: Recent developments, applications, and future directions, Applied Soft Computing 143 (2023) 110377.

[89] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117.

[90] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you? explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[91] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, volume 30, 2017.

[92] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3319–3328.

[93] R. R. Selvaraju, et al., Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[94] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.

[95] J.-S. R. Jang, Anfis: Adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (1993) 665–685.

[96] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[97] P. J. Huber, Robust estimation of a location parameter, The Annals of Mathematical Statistics 35 (1964) 73–101.

[98] W. Liu, P. P. Pokharel, J. C. Principe, Correntropy: Properties and applications in non-gaussian signal processing, IEEE Transactions on Signal Processing 55 (2007) 5286–5298.

[99] Z. Zhang, M. R. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: Advances in Neural Information Processing Systems, 2018.

[100] Y. Wang, et al., Symmetric cross entropy for robust learning with noisy labels, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 322–330.

[101] E. Amid, M. K. Warmuth, Robust bi-tempered logistic loss based on bregman divergences, in: Advances in Neural Information Processing Systems, 2019.

[102] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, 2015.

[103] J. Gawlikowski, et al., A survey of uncertainty in deep neural networks, Artificial Intelligence Review 56 (2023) 1513–1589.

[104] Y. Bengio, et al., Curriculum learning, in: Proceedings of the 26th International Conference on Machine Learning, 2009, pp. 41–48.

[105] M. P. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, in: Advances in Neural Information Processing Systems, 2010.

[106] B. Han, et al., Co-teaching: Robust training of deep neural networks with extremely noisy labels, in: Advances in Neural Information Processing Systems, 2018.

[107] J. Li, et al., Dividemix: Learning with noisy labels as semi-supervised learning, in: International Conference on Learning Representations, 2020.

[108] K. T. Atanassov, Intuitionistic fuzzy sets, Fuzzy Sets and Systems 20 (1986) 87–96.

[109] S. K. Pal, S. Mitra, Neuro-fuzzy pattern recognition: Methods in soft computing, John Wiley & Sons (1997).

[110] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: International Conference on Learning Representations Workshop, 2014.

[111] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proceedings of the 34th International Conference on Machine Learning, 2017.

[112] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, in: International Conference on Learning Representations Workshop, 2015.

[113] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, Smoothgrad: Removing noise by adding noise, arXiv preprint arXiv:1706.03825 (2017).

[114] A. Chattopadhay, et al., Grad-cam++: Improved visual explanations for deep convolutional networks, in: IEEE Winter Conference on Applications of Computer Vision, 2018, pp. 839–847.

[115] H. Wang, et al., Score-cam: Score-weighted visual explanations for convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020.

[116] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, B. Li, Axiom-based grad-cam: Towards accurate visualization and explanation of cnns, in: British Machine Vision Conference, 2020.

[117] M. Muhammad, M. Yeasin, Eigen-cam: Class activation map using principal components, arXiv preprint arXiv:2008.00299 (2020).

[118] J. Adebayo, et al., Sanity checks for saliency maps, in: Advances in Neural Information Processing Systems, volume 31, 2018.

[119] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 1263–1284.

[120] N. V. Chawla, et al., Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[121] H. He, Y. Bai, E. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: IJCNN, 2008.

[122] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: A new over-sampling method in imbalanced data sets learning, in: ICIC, 2005.

[123] T.-Y. Lin, et al., Focal loss for dense object detection, in: ICCV, 2017.

[124] Y. Cui, et al., Class-balanced loss based on effective number of samples, in: CVPR, 2019.

[125] B. Kang, et al., Decoupling representation and classifier for long-tailed recognition, in: ICLR, 2020.

[126] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.

[127] A. Hirose, Complex-Valued Neural Networks, Springer, 2012.

[128] D. P. Mandic, V. S. L. Goh, Complex Valued Nonlinear Adaptive Filters, Wiley, 2009.

[129] C. Trabelsi, et al., Deep complex networks, in: International Conference on Learning Representations, 2018.

[130] A. Hirose, Complex-Valued Neural Networks: Advances and Applications, Wiley, 2013.

[131] J. S. Dramsch, et al., Complex-valued neural networks for machine learning on nonstationary physical data, Computers & Geosciences 130 (2019) 79–89.

[132] D. Brandwood, A complex gradient operator and its application in adaptive array theory, IEE Proceedings F 130 (1983) 11–16.

[133] 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.

[134] 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.

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

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

[137] J. Wang, S. Lu, S.-H. Wang, Y.-D. Zhang, A review on extreme learning machine, Multimedia Tools and Applications 81 (2022) 41611–41660.

[138] J. Du, C.-M. Vong, C. P. Chen, Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification, IEEE Transactions on Cybernetics 51 (2020) 1586–1597.

[139] Z. Gao, W. Dang, M. Liu, W. Guo, K. Ma, G. Chen, Classification of EEG signals on vep-based bci systems with broad learning, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (2021) 7143–7151.

[140] T. Zhang, X. Wang, X. Xu, C. P. Chen, Gcb-net: Graph convolutional broad network and its application in emotion recognition, IEEE Transactions on Affective Computing 13 (2019) 379–388.

[141] J. Wang, P. Gao, J. Zhang, C. Lu, B. Shen, Knowledge augmented broad learning system for computer vision based mixed-type defect detection in semiconductor manufacturing, Robotics and Computer-Integrated Manufacturing 81 (2023) 102513.

[142] J. Liang, W. Ai, H. Chen, G. Tang, Communication-efficient decentralized elastic-net broad learning system based on quantized and censored communications, Applied Soft Computing (2023) 109999.

[143] Z. Wang, H. Liu, X. Xu, F. Sun, Multi-modal broad learning for material recognition, Cognitive Computation and Systems 3 (2021) 123–130.

[144] S. Feng, C. P. Chen, Broad learning system for control of nonlinear dynamic systems, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2018, pp. 2230–2235, https://doi.org/10.1109/SMC.2018.00383.

[145] C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE Transactions on Neural Networks 13 (2002) 464–471.

[146] A. K. Malik, M. A. Ganaie, M. Tanveer, P. N. Suganthan, A. D. N. I. Initiative, Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network, IEEE Transactions on Computational Social Systems (2022, https://doi.org/10.1109/TCSS.2022.3146974) 1–12.

[147] C. F. Lin, S. D. Wang, Fuzzy support vector machines, IEEE Transactions on Neural Networks 13 (2002) 464–471.

[148] K. T. Atanassov, K. T. Atanassov, Intuitionistic Fuzzy Sets, Springer, 1999.

[149] C. Ren, S. He, X. Luan, F. Liu, H. R. Karimi, Finite-time L_2-gain asynchronous control for continuous-time positive hidden markov jump systems via T–S fuzzy model approach, IEEE Transactions on Cybernetics 51 (2020) 77–87.

[150] R. Tabbussum, A. Q. Dar, Performance evaluation of artificial intelligence paradigms—artificial neural networks, fuzzy logic, and adaptive neuro-fuzzy inference system for flood prediction, Environmental Science and Pollution Research 28 (2021) 25265–25282.

[151] H. Ouifak, A. Idri, On the performance and interpretability of mamdani and takagi-sugeno-kang based neuro-fuzzy systems for medical diagnosis, Scientific African 20 (2023) e01610.

[152] F. R. Chung, F. C. Graham, Spectral graph theory, 92, American Mathematical Soc., 1997.

[153] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research 7 (2006) 1–30.

[154] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (1937) 675–701.

[155] R. L. Iman, J. M. Davenport, Approximations of the critical region of the fbietkan statistic, Communications in Statistics-Theory and Methods 9 (1980) 571–595.

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

[157] C. R. Rao, S. K. Mitra, Generalized inverse of a matrix and its applications, in: Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics, volume 6, University of California Press, 1972, pp. 601–621.

[158] 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.

[159] D. Needell, A. A. Nelson, R. Saab, P. Salanevich, O. Schavemaker, Random vector functional link networks for function approximation on manifolds, Frontiers in Applied Mathematics and Statistics 10 (2024) 1284706.

[160] 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.

[161] S. Feng, C. P. Chen, C.-Y. Zhang, A fuzzy deep model based on fuzzy restricted boltzmann machines for high-dimensional data classification, IEEE Transactions on Fuzzy Systems 28 (2019) 1344–1355.

[162] Y. Li, L. Liu, G. Feng, Adaptive finite-time controller design for t–s fuzzy systems, IEEE Transactions on Cybernetics 47 (2017) 2425–2436.

[163] G. Xue, Q. Chang, J. Wang, K. Zhang, N. R. Pal, An adaptive neuro-fuzzy system with integrated feature selection and rule extraction for high-dimensional classification problems, IEEE Transactions on Fuzzy Systems 31 (2023) 2167–2181.

[164] G. Xue, J. Wang, B. Yuan, C. Dai, DG-ALETSK: A high-dimensional fuzzy approach with simultaneous feature selection and rule extraction, IEEE Transactions on Fuzzy Systems (2023).

[165] A. M. AlRassas, M. A. Al-Qaness, A. A. Ewees, S. Ren, R. Sun, L. Pan, M. Abd Elaziz, Advance artificial time series forecasting model for oil production using neuro fuzzy-based slime mould algorithm, Journal of Petroleum Exploration and Production Technology (2022) 1–13.

[166] A. Vlasenko, N. Vlasenko, O. Vynokurova, Y. Bodyanskiy, D. Peleshko, A novel ensemble neuro-fuzzy model for financial time series forecasting, Data 4 (2019) 126.

[167] S. Han, K. Zhu, M. Zhou, X. Liu, Evolutionary weighted broad learning and its application to fault diagnosis in self-organizing cellular networks, IEEE Transactions on Cybernetics 53 (2022) 3035–3047.

[168] A. T. Dosdoğru, Improving weather forecasting using de-noising with maximal overlap discrete wavelet transform and ga based neuro-fuzzy controller, International Journal on Artificial Intelligence Tools 28 (2019) 1950012.

[169] H. Zhao, Y. Wu, W. Deng, An interpretable dynamic inference system based on fuzzy broad learning, IEEE Transactions on Instrumentation and Measurement (2023).

[170] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Transactions on Neural Networks 11 (2000) 748–768.

[171] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.

[172] S. Ghosh, S. K. Dubey, Comparative analysis of k-means and fuzzy c-means algorithms, International Journal of Advanced Computer Science and Applications 4 (2013).

[173] A. Iosifidis, A. Tefas, I. Pitas, Graph embedded extreme learning machine, IEEE Transactions on Cybernetics 46 (2015) 311–324.

[174] M. A. Ganaie, M. Tanveer, P. N. Suganthan, Minimum variance embedded random vector functional link network, in: Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part V 27, Springer, 2020, pp. 412–419.

[175] A. Iosifidis, A. Tefas, I. Pitas, Minimum class variance extreme learning machine for human action recognition, IEEE Transactions on Circuits and Systems for Video Technology 23 (2013) 1968–1979. doi:10.1109/TCSVT.2013.2269774.

[176] C. Molnar, Interpretable machine learning, Lulu. com, 2020.

[177] M. T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you? explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[178] R. Xie, C.-M. Vong, S. Wang, Doubly interpretable fuzzy apriori classifier by successive stacking and one-step wide calculation, IEEE Transactions on Fuzzy Systems 32 (2024) 1653–1667.

[179] D. Alvarez Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, Advances in Neural Information Processing Systems 31 (2018).

[180] D. V. Carvalho, E. M. Pereira, J. S. Cardoso, Machine learning interpretability: A survey on methods and metrics, Electronics 8 (2019) 832.

[181] F. A. Spanhol, L. S. Oliveira, C. Petitjean, L. Heutte, A dataset for breast cancer histopathological image classification, IEEE Transactions on Biomedical Engineering 63 (2015) 1455–1462.

[182] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, M. N. Vrahatis, No free lunch theorem: A review, Approximation and Optimization: Algorithms, Complexity and Applications (2019) 57–82.

[183] S. Sharma, S. Sharma, A. Athaiya, Activation functions in neural networks, Towards Data Sci 6 (2017) 310–316.

[184] 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.

[185] A. Tharwat, T. Gaber, A. Ibrahim, A. E. Hassanien, Linear discriminant analysis: A detailed tutorial, AI Communications 30 (2017) 169–190.

[186] Y. Zhou, H. Yu, X. Cai, A novel k-means algorithm for clustering and outlier detection, in: 2009 Second International Conference on Future Information Technology and Management Engineering, IEEE, 2009, pp. 476–480.

[187] D. R. Musicant, NDC: normally distributed clustered datasets, 1998. URL: www.cs.wisc.edu/dmi/svm/ndc/.

[188] A. Kumari, M. Akhtar, M. Tanveer, M. Arshad, Diagnosis of breast cancer using flexible pinball loss support vector machine, Applied Soft Computing 157 (2024) 111454.

[189] M. Tanveer, T. Goel, R. Sharma, A. K. Malik, I. Beheshti, J. Del Ser, P. N. Suganthan, C. T. Lin, Ensemble deep learning for Alzheimer's disease characterization and estimation, Nature Mental Health (2024) 1–13.

[190] M. Khojaste-Sarakhsi, S. S. Haghighi, S. F. Ghomi, E. Marchiori, Deep learning for Alzheimer's disease diagnosis: A survey, Artificial Intelligence in Medicine 130 (2022) 102332.

[191] M. Tanveer, B. Richhariya, R. U. Khan, A. H. Rashid, P. Khanna, M. Prasad, C. T. 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.

[192] A. M. Martinez, A. C. Kak, PCA versus LDA, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 228–233.

[193] M. Sugiyama, Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis., Journal of Machine Learning Research 8 (2007).

[194] B. B. Hazarika, D. Gupta, P. Borah, An intuitionistic fuzzy kernel ridge regression classifier for binary classification, Applied Soft Computing 112 (2021) 107816.

[195] B. Richhariya, M. Tanveer, A. H. 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.

[196] 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 11 (2024) 4744–4753.

[197] R. Chakraborty, Y. Xing, X. Y. Stella, Surreal: Complex-valued learning as principled transformations on a scaling and rotation manifold, IEEE Transactions on Neural Networks and Learning Systems 33 (2020) 940–951.

[198] P. Zhao, S. Wang, X. Hao, Z. Wang, J. Zou, J. Ren, Y. Dai, An ensemble multi-dimensional randomization network for intelligent recognition of tobacco baking stage, Scientific Reports 15 (2025) 1346.

[199] N. Sabri, S. Aljunid, M. Salim, R. Badlishah, R. Kamaruddin, M. Malek, Fuzzy inference system: Short review and design, Int. Rev. Autom. Control 6 (2013) 441–449.

[200] 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.

[201] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[202] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, M. Palaniswami, Fuzzy c-means algorithms for very large data, IEEE Transactions on Fuzzy Systems 20 (2012) 1130–1146.

[203] M. Hu, R. Gao, P. N. Suganthan, Self-distillation for randomized neural networks, IEEE Transactions on Neural Networks and Learning Systems 35 (2024) 16119–16128.

[204] D. Musicant, NDC: Normally distributed clustered datasets, 1998. Www.cs.wisc.edu/dmi/svm/ndc/.

[205] 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 21 (2024) 546–558.

[206] 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.

[207] R. Gao, M. Ganaie, P. N. Suganthan, Random vector functional link network: recent developments, applications, and future directions, Applied Soft Computing (2023) 110377.

[208] 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.

[209] L. Zhang, M. Wang, M. Liu, D. Zhang, A survey on deep learning for neuroimaging-based brain disorder analysis, Frontiers in Neuroscience 14 (2020) 779.

[210] S. Srivastava, R. Ahmad, S. K. Khare, Alzheimer's disease and its treatment by different approaches: A review, European Journal of Medicinal Chemistry 216 (2021) 113320.