# Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary TRVFL Classifier

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
**Bachelor of Technology**
in
**Computer Science & Engineering**

by
**Prajapati Raman(20223177)**
**Piyush Tanay(20223175)**
**Priyanshu Singh(20223191)**
**Nikhil Jarwar(20223163)**

to the
**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD PRAYAGRAJ
**November, 2025**

# UNDERTAKING

I declare that the work presented in this report titled "*Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary TRVFL Classifier*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

November, 2025
Allahabad

(Prajapati
Raman(20223177)
Piyush Tanay(20223175)
Priyanshu Singh(20223191)
Nikhil Jarwar(20223163) )

# CERTIFICATE

Certified that the work contained in the report titled "*Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary TRVFL Classifier*", by *Prajapati Raman(20223177)*
*Piyush Tanay(20223175)*
*Priyanshu Singh(20223191)*
*Nikhil Jarwar(20223163)* , has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

---

(Prof. Deepak Gupta)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

November, 2025

# Preface

A good B.Tech. thesis is one that helps you in furthering your interest in a specific field of study. Whether you plan to work in an industry or wish to take up academics as a way of life, your thesis plays an important role.

Your thesis should judiciously combine theory with practice. It should result in a realization of reasonably complex system (software and/or hardware). Given various limitations, it is always better to extend your predecessor's work. If you plan it properly, you can really build on the experience of your seniors.

# Acknowledgements

Here it will go something like this...........It is a great pleasure to thank the giants on whose shoulders I stand. First of all, I would like to thank my supervisor ...

# Contents

# Chapter 1

# Introduction

Alzheimer's Disease (AD) is a neurodegenerative disorder characterized by progressive memory impairment, cognitive decline, and an individual's inability to execute the activities of daily living. It is important to identify AD early because effective intervention can delay the progression of the disease and increase quality of life for patients and their families. Recent advances in machine learning and deep learning provide the capability for automated analysis of neuroimaging data, which assists in the early and accurate identification of AD. This project demonstrates a multimodal neuroimaging-based AD diagnostic system based on the ADNI MRI dataset using ResNet-50 neural network feature extraction along with multiple other advanced classifiers RVFL and modified RVFLs (such as EDRVFL, TRVFL, Swift Classifier, and ConvNeXtV2).

## 1.1 Motivation

Alzheimer's Disease (AD) is a progressive neurodegenerative condition that steadily impacts memory, cognition, and behavioral functioning. The increasing elderly population globally increases the prevalence of Alzheimer's Disease at a rapidly growing rate, making it a significant public health issue. According to recent clinical studies, there are millions of people around the globe with AD, and this will increase dramatically in coming decades. Despite significant research efforts, AD does not have a cure, which underscores the significant importance of early and accurate diagnosis. Early diagnosis allows for timely intervention, slows the progression of the disease, and improves the quality of life for the patient.

Magnetic Resonance Imaging has become one of the best non-invasive techniques available to assess structural brain alterations associated with AD; however, the reading of MRIs relies on expert knowledge in radiology and is often based on a highly subjective and time-consuming analysis that also suffers from inter-observer variability. Traditional machine-learning approaches have offered partial solutions, but there remain challenges associated with capturing the complexity of patterns associated with neuroimaging.

Despite such advancements, the unification of deep feature extraction with fast, robust, and generalizable classifiers still remains an open challenge. Specifically, evolutionary variants of TRVFL can be further optimized to improve generalization and reduce overfitting, especially in complex neuroimaging tasks. There is, therefore, a strong motivation to develop a comprehensive multimodal framework that will unify ResNet-50-based feature extraction with advanced evolutionary TRVFL classifiers in the interest of reliable diagnosis of Alzheimer's Disease using the ADNI MRI dataset.

These findings motivate the present work, which aims at bridging existing gaps through using multimodal neuroimaging, advanced deep-learning feature extraction, and evolutionary classification techniques to be able to reach a more efficient, accurate, and automated AD diagnosis system.

## 1.2 Problem Statement

The difficulty in diagnosing Alzheimer's Disease from MRI scans lies in the fact that there are subtle structural brain changes, high-dimensional data, and a high reliance on subjective expert interpretation. Most of the traditional models in machine learning have poor generalization performance when applied to complex neuroimaging datasets. Though deep learning techniques like ResNet-50 extract meaningful features, choosing an efficient classifier with good accuracy is still a main issue. Thus, this project tries to overcome these shortcomings by developing a robust and reliable framework combining deep feature extraction with deep learning and machine learning models.

## 1.3 Objectives

The main objectives of this project are as follows:

- To develop an automated Alzheimer's Disease diagnosis framework using MRI data from the ADNI dataset.
- To extract deep and discriminative features from MRI scans using the ResNet-50 architecture.
- To implement and evaluate multiple classifiers, including RVFL, EDRVFL, TRVFL, Evolutionary TRVFL, Swift Classifier, and ConvNeXtV2.
- To perform hyperparameter tuning for all models to achieve optimal performance and ensure fair comparison.
- To compare the performance of all models and determine the most effective classifier for AD detection using various evaluation metrics such as accuracy, precision, recall, F1-score, specificity, sensitivity, ROC-AUC, and confusion matrix analysis.

# Chapter 2

# Literature Review and Related Work

## 2.1 Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary RVFL Classifier (Goel et al., 2023)

Goel et al. [2] proposed a multimodal neuroimaging framework for early Alzheimer's Disease (AD) diagnosis using an Evolutionary Random Vector Functional Link (E-RVFL) classifier. The study combines structural MRI and PET imaging modalities to capture both anatomical and metabolic alterations associated with AD progression. Deep and handcrafted features extracted from these modalities were fused to create a richer and more discriminative feature representation.

The novelty of the work lies in the evolutionary optimization of RVFL parameters, including weights, biases, and regularization coefficients. This optimization strategy enhances model generalization, reduces sensitivity to random initialization, and improves robustness when handling high-dimensional multimodal data. The E-RVFL classifier demonstrated faster convergence and superior classification performance.

Experiments conducted on the ADNI dataset showed that the proposed E-RVFL model outperformed several baseline classifiers, including traditional RVFL, SVM, and deep CNN-based approaches. The model achieved higher accuracy, sensitivity, specificity, and F1-score, highlighting the effectiveness of combining multimodal imaging with evolutionary learning techniques for reliable and early Alzheimer's Disease diagnosis.

## 2.2 Conv-eRVFL: CNN-Based Ensemble RVFL Classifier for Alzheimer's Disease Diagnosis (Sharma et al., 2023)

Sharma et al. [8] introduced Conv-eRVFL, a hybrid classification framework that integrates Convolutional Neural Networks (CNN) with an Ensemble Random Vector Functional Link (eRVFL) classifier for Alzheimer's Disease (AD) diagnosis. The model leverages both structural MRI and PET modalities to capture complementary anatomical and metabolic biomarkers associated with neurodegeneration.

In this approach, CNN-based deep feature extraction captures rich spatial information from neuroimaging data, while the ensemble RVFL classifier enhances decision robustness by aggregating outputs from multiple RVFL networks. The combination of CNN feature representations with the ensemble RVFL architecture leads to improved learning stability and better generalization in high-dimensional medical imaging environments.

Experiments conducted on the ADNI dataset demonstrated that Conv-eRVFL outperformed traditional RVFL, standalone CNN classifiers, and other baseline models. The proposed framework achieved superior results in terms of accuracy, sensitivity, specificity, and F1-score. The study highlights the effectiveness of integrating deep CNN feature extraction with ensemble RVFL classifiers for reliable, scalable, and efficient Alzheimer's Disease diagnosis.

### 2.2.1 Biomarker Models

Early work on Alzheimer's disease (AD) focused on understanding biomarker progression. Jack et al. [4] proposed a dynamic model describing the sequence of pathological changes in AD. The ADNI study [3] further standardized MRI acquisition and established a widely used benchmark dataset.

### 2.2.2 Classical ML Approaches

Traditional machine learning techniques relied on handcrafted MRI features. Fan et al. [1] demonstrated that feature-selected structural MRI measurements combined with SVMs can effectively classify AD and MCI.

### 2.2.3 Deep Learning Methods

Deep learning significantly improved MRI feature representation. Suk et al. [9] introduced hierarchical deep feature learning with multimodal fusion. Payan and Montana [6] used 3D CNNs to directly learn from volumetric MRI. Liu et al. [5] proposed landmark-based deep multi-instance learning for more discriminative spatial pattern extraction.

### 2.2.4 MRI Preprocessing

Proper MRI preprocessing enhances the reliability of longitudinal studies. Reuter et al. [7] presented a robust within-subject template estimation method using FreeSurfer, improving consistency in downstream classification.

### 2.2.5 Summary

Overall, research has shifted from handcrafted structural features to end-to-end deep learning frameworks, supported by standardized datasets like ADNI and strong preprocessing pipelines.

# Chapter 3

# Methodology and System Architecture

## 3.1 Methodological Foundation

The methodological foundation of this project integrates deep feature extraction and advanced classification techniques to develop an efficient framework for Alzheimer's Disease (AD) diagnosis using MRI data from the ADNI dataset. The overall methodology consists of neuroimaging preprocessing, deep feature extraction, and classification using RVFL-based models along with modern deep-learning classifiers.

The MRI scans first undergo essential preprocessing steps such as skull stripping, intensity normalization, resizing, and noise reduction. These steps ensure consistency across the dataset and preserve important structural features relevant to AD diagnosis. Following preprocessing, deep feature extraction is performed using the ResNet-50 architecture, which captures high-level spatial and structural patterns from the MRI images through its residual learning framework. These extracted deep features form the input representation for various classifiers.

Random Vector Functional Link (RVFL) networks serve as one of the primary classification techniques due to their fast learning capability and direct input-to-output connections. Enhanced variants such as EDRVFL, TRVFL, and Evolutionary TRVFL are also used to improve model stability, generalization, and performance. Evolutionary TRVFL incorporates population-based optimization strategies to refine weights, biases, and regularization parameters, leading to more robust decision boundaries and better classification outcomes.

In addition to RVFL-based models, the methodology also includes modern deep-learning classifiers such as the Swift Classifier and ConvNeXt V2. The Swift Classifier offers rapid inference and lightweight architecture suitable for medical imaging tasks, while ConvNeXt V2 provides enhanced hierarchical feature learning through a next-generation convolutional design. These models complement the RVFL-based methods and enable a comprehensive comparative analysis.

The performance of all models is evaluated using a wide range of metrics, including accuracy, precision, recall, F1-score, sensitivity, specificity, ROC-AUC, and confusion matrix analysis. These metrics help in identifying the most effective classifier and provide a thorough understanding of each model's strengths in detecting Alzheimer's Disease.

Overall, the methodological foundation of this project combines deep neural feature extraction, evolutionary learning, classical machine-learning models, and advanced deep-learning architectures to create a robust, scalable, and accurate diagnostic framework for Alzheimer's Disease detection using MRI data.

### 3.1.1 Preprocessing

The dataset used in this study is obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI), consisting of MRI scans categorized into three major classes: Alzheimer's Disease (AD), Cognitively Normal (CN), and Mild Cognitive Impairment (MCI). This diverse class distribution enables effective evaluation of the proposed models across different stages of neurodegeneration.

1. **Feature Extraction** Feature extraction in this project is performed using the ResNet-50 deep convolutional neural network, which is widely recognized for its strong feature representation capabilities. ResNet-50 employs a residual learning framework consisting of 50 layers, enabling it to capture both low-level and high-level structural patterns present in MRI images. The residual connections help mitigate the vanishing gradient problem, allowing the network to learn deeper and more discriminative features.

   In this work, the preprocessed MRI scans are passed through the pre-trained ResNet-50 model (excluding the final classification layers) to extract deep feature vectors. These features serve as robust input representations for the subsequent RVFL-based models, Swift Classifier, and ConvNeXt V2. By utilizing ResNet-50, the extracted features effectively encode important spatial, textural, and anatomical information relevant to distinguishing between AD, MCI, and CN classes.

2. **Feature Matrix Cleaning, Imputation, and One-Hot Encoding:** After ResNet-50 feature extraction, the feature matrix is cleaned by converting all feature columns into numeric format, with invalid or non-numeric values coerced into NaN. Infinite values and zeros are treated as missing data to prevent distortion in learning. Columns containing only NaN values are removed, and the remaining missing values are imputed using the median of each feature. Additionally, categorical attributes such as class labels are processed using one-hot encoding to convert them into machine-readable binary vectors. This ensures that all features are numeric, consistent, and properly formatted for downstream classifiers.

## 3.2 Model Architecture

### 3.2.1 RVFL Architecture:

The Random Vector Functional Link (RVFL) model used in this study consists of a direct input-to-output connection combined with a randomly generated hidden layer. The architecture includes three primary components: (i) a direct linear layer that maps the input features to the output space, (ii) a hidden layer with randomly assigned weights followed by a sigmoid activation function, and (iii) a hidden-to-output layer that transforms the activated hidden features. During the forward pass, the hidden features are concatenated with the original input to form an enhanced feature representation, while the final prediction is obtained by summing the outputs of the direct and hidden pathways. This architecture enables fast training, improved generalization, and efficient handling of high-dimensional feature vectors.

### 3.2.2 EDRVFL Architecture:

The EDRVFL model is composed of multiple enhancement layers, each containing randomly initialized weights and biases. Every layer performs standardization, nonlinear activation (such as sigmoid, sine, hardlim, tribas, radbas, ReLU, or leaky ReLU), and concatenation of hidden features with the original input. A bias term is appended, and the output weights for each layer are computed analytically using ridge regression. Instead of relying on a single output, EDRVFL aggregates predictions from all layers using (i) majority voting and (ii) additive softmax-based probability fusion. This ensemble of deep random layers increases robustness, reduces overfitting, and enables strong performance even with limited training data.

### 3.2.3 ConvNeXtV2 Feature-Aware Classifier Architecture:

To utilize the high-dimensional ResNet-50 feature vectors within a convolutional architecture, a feature–grid projection strategy is adopted. The original feature vector is reshaped into a 2D grid by computing an optimal height–width decomposition; if the vector length is not perfectly divisible, zero-padding is applied to preserve spatial structure. The reshaped input is treated as a single-channel image and passed through the ConvNeXtV2 backbone.

ConvNeXtV2 consists of four hierarchical stages, each beginning with a downsampling layer followed by multiple depthwise convolutional blocks. Every block includes: (1) a depthwise convolution, (2) Layer Normalization, (3) pointwise linear expansions, (4) a GELU activation, (5) a Global Response Normalization (GRN) module, and (6) a final pointwise projection. To improve generalization, stochastic depth via the DropPath operator is applied across blocks.

The final stage output is globally averaged and normalized before being forwarded to a fully connected classification head. The classifier is optimized using the BCEWithLogits loss with an AdamW optimizer and softmax-based probability inference. This design allows ConvNeXtV2 to learn spatially coherent patterns from reshaped feature embeddings, enabling high performance even without using raw image inputs.

### 3.2.4 Swin Transformer V2 Feature-Aware Classifier Architecture:

To transform a high-dimensional feature vector into a structure suitable for hierarchical attention, the input feature embedding is partitioned into a fixed number of tokens arranged on a 2D grid. Let the vector length be $F$ and the number of tokens be $T = H \times W$, where $T$ is a perfect square. Each chunk of size $F/T$ is projected into an embedding space using a linear patch-projection layer, producing a $(H \times W)$ token map.

This token grid is processed by a two-stage Swin Transformer V2 hierarchy. Each stage consists of stacked Swin Transformer blocks that alternate between standard windowed multi-head self-attention (W-MSA) and shifted window attention (SW-MSA). Swin V2 enhancements—cosine attention scaling, continuous relative position bias via an MLP, logit-scale stabilization, and improved normalization—are incorporated to improve training stability on small windows.

Within each block, LayerNorm is applied before attention, followed by residual connections, stochastic depth (DropPath), and a GELU-activated MLP expansion layer. Patch Merging is used between stages to reduce spatial resolution while doubling channel dimensionality, enabling a hierarchical encoder similar to CNN downsampling.

After the final stage, features are normalized and globally averaged across tokens. A fully connected classification head produces the output logits for the three target classes. Training uses the AdamW optimizer with cross-entropy loss and softmax-based inference. This design allows the Swin V2 architecture to learn structured correlations within pre-extracted features, even without raw image inputs.

### 3.2.5 Translated Random Vector Functional Link (TRVFL) Architecture:

The TRVFL model extends the traditional Random Vector Functional Link network by incorporating a translation scaling mechanism that improves the diversity of randomly generated hidden features. The architecture consists of three components: (i) a direct input-to-output linear mapping, (ii) a randomly initialized hidden layer with fixed weights, and (iii) an expanded feature space formed by concatenating hidden activations with the raw inputs.

## 3.3 Tuning

### 3.3.1 TRVFL Hyperparameter Tuning

The hyperparameter optimization for the Translated Random Vector Functional Link Network (TRVFL) was performed using a Genetic Algorithm (GA) to maximize validation accuracy. The GA explored four key parameters: hidden layer size, regularization coefficient $\lambda$, activation function, and translation scale. The search space consisted of 10–1000 hidden neurons, regularization values ranging from $10^{-8}$ to $10^{4}$, activation functions {sigmoid, relu, tanh}, and translation scales within 0.0–1.0.

The GA was configured with a population size of 20, 15 generations, elitism of 4 individuals, a mutation rate of 0.3, and uniform crossover, ensuring sufficient exploration while maintaining stability through elite preservation. Each candidate model was trained using a fast closed-form ridge regression solution:

$$W = \left(\lambda I + H^T H\right)^{-1} H^T Y,$$

where $H$ denotes the TRVFL-expanded feature matrix and $Y$ is the one-hot encoded label matrix. The computed weights were directly assigned to the model's hidden-output and direct-link layers, enabling rapid evaluation of each configuration.

For every generation, two levels of logs were maintained: (1) a generation summary tracking best and mean validation accuracy, and (2) full candidate-level history storing all hyperparameter–performance relationships. These logs support convergence visualization and parameter-performance analysis.

After 15 generations, the GA identified the following best configuration: 843 hidden neurons, $\lambda = 1.7468$, ReLU activation, and a translation scale of 0.5979, achieving the highest validation accuracy among all explored candidates. Analysis of GA trends revealed that larger hidden layers (700–1000), ReLU activations, and moderate regularization consistently promoted superior generalization, while translation scaling around 0.5–0.7 improved feature separation. The optimized configuration was subsequently used for final model training and evaluation.

### 3.3.2 Swin Classifier Hyperparameter Tuning

To optimize the performance of the Swin Transformer–based classifier, a random search strategy was employed over a multi-dimensional hyperparameter space. The search included learning rate, weight decay, embedding dimension, hierarchical depth configuration, number of attention heads, and drop-path regularization probability. The tuning space consisted of:

$$lr \in \{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}, \quad weightdecay \in \{10^{-3}, 10^{-2}, 5 \times 10^{-2}\},$$

$$embeddim \in \{64, 128, 256\}, \quad depths \in \{(2,2), (2,4), (4,4)\},$$

$$numheads \in \{(4,8), (8,16), (4,4)\}, \quad drop - pathrate \in \{0.05, 0.1, 0.2\}.$$

Given the large theoretical configuration space, a fixed-budget random search with five trials was used. For each sampled configuration, the classifier was trained for 25 epochs using the AdamW optimizer and one-hot encoded labels, with Binary Cross-Entropy Loss applied on logits. The model's performance was evaluated using validation accuracy, and both configuration-wise results and the best-performing model were stored.

During each trial, the SwinClassifier was instantiated with the sampled hyperparameters:

$$Config = \{lr, weightdecay, embeddim, depths, heads, drop - path\},$$

and optimized end-to-end on the extracted deep features. Validation accuracy served as the fitness measure, enabling comparison across configurations.

After completing all trials, the random search identified the optimal configuration as the one achieving the highest validation accuracy. This configuration, along with its trained model weights, was preserved for final testing. The tuning results demonstrate the effectiveness of lightweight random search for high-dimensional transformer-based architectures, capturing strong-performing combinations of embedding size, depth, and regularization.

### 3.3.3 ConvNeXtV2 Hyperparameter Tuning Setup

To systematically optimize the ConvNeXtV2-based classifier, an exhaustive grid search strategy was designed across multiple architectural and training parameters. The search space included four model variants (atto, femto, pico, nano) and key parameters such as learning rate, batch size, drop-path rate, head initialization scale, weight decay, and training epochs.

The full hyperparameter grid consisted of the following ranges:

- Model variant: atto, femto, pico, nano
- Learning rate: $1 \times 10^{-4}$, $3 \times 10^{-4}$, $1 \times 10^{-3}$
- Batch size: 16, 32, 64
- Drop-path rate: 0.0, 0.1, 0.2
- Head initialization scale: 0.5, 0.75, 1.0
- Weight decay: $1 \times 10^{-4}$, $5 \times 10^{-4}$
- Epochs: 20, 30, 40

Due to the large search space, a reduced grid was also prepared for faster experimentation. This smaller configuration focused on the femto and pico variants with narrowed parameter ranges:

- Model variant: femto, pico
- Learning rate: $3 \times 10^{-4}$, $1 \times 10^{-3}$
- Batch size: 32, 64
- Drop-path rate: 0.0, 0.1
- Head initialization scale: 0.75
- Weight decay: $1 \times 10^{-4}$

– Epochs: 30

Each hyperparameter configuration was generated using a Cartesian product over the selected grid, enabling systematic exploration while maintaining computational feasibility. This tuning procedure ensured robust selection of the optimal ConvNeXtV2 variant and its corresponding training setup.

### 3.3.4 RVFL Hyperparameter Tuning Setup

To identify the optimal configuration for the Random Vector Functional Link (RVFL) network, a structured hyperparameter search was conducted over two fundamental parameters: the number of hidden enhancement nodes and the regularization coefficient $\lambda$. The tuning aimed to jointly analyze the influence of model capacity (via hidden neurons) and generalization control (via Tikhonov regularization).

The hyperparameter space explored in this study consisted of:

- **Hidden nodes:** Multiple enhancement-layer sizes were tested (e.g., 256, 512, 768, 843).
- **Regularization parameter $\lambda$:** A logarithmically spaced range was evaluated to capture both weak and strong regularization regimes.

For each hyperparameter configuration, the RVFL model was trained on the training split and evaluated on the validation split. Both training and validation accuracies were recorded to analyze how regularization and hidden-layer width affect performance. A comprehensive performance visualization was generated, including:

- **Train accuracy vs.** $\log_{10}(\lambda)$ **curves** for different hidden-node settings.
- **Validation accuracy vs.** $\log_{10}(\lambda)$ **curves** to identify overfitting trends.
- **A heatmap of validation accuracy**, with hidden-node count on the vertical axis and $\log_{10}(\lambda)$ on the horizontal axis, providing a global view of search-space performance.

The Cartesian evaluation of hidden size and regularization strength enabled systematic identification of the best-performing configuration. The optimal hyperparameter pair was later used to train the final RVFL model on the combined training and validation set before reporting the performance on the held-out test set.

### 3.3.5 EDRVFL Hyperparameter Tuning Setup

To optimize the Ensemble Deep Random Vector Functional Link (EDRVFL) classifier, a targeted hyperparameter search was conducted focusing on the most influential architectural and regularization parameters. The EDRVFL model extends the traditional RVFL network by stacking multiple enhancement layers and combining their predictions through both voting- and addition-based ensemble strategies. Therefore, selecting appropriate values for the number of enhancement nodes, depth of stacked layers, and regularization strength is essential for stable and high-performing classification.

The hyperparameter space explored consisted of the following components:

- **Number of enhancement nodes** ($n_{nodes}$): Determines the dimensionality of the feature expansion in each RVFL layer.
- **Regularization coefficient** ($\lambda$): Controls the magnitude of the ridge regression penalty during output-weight computation.
- **Number of stacked layers** ($n_{layer}$): Specifies the depth of the EDRVFL architecture, enabling multi-level feature transformation.
- **Activation function**: The nonlinear activation applied in enhancement layers (ReLU in this study).

Each possible parameter configuration was evaluated by training the EDRVFL on the training split and measuring performance on the validation split using both ensemble metrics:

- **Voting accuracy** — majority vote across layer outputs.

– **Addition accuracy** — aggregate probability across layers.

For each configuration, both training and validation accuracies were recorded, producing a detailed search log used later for visualization and model selection. The search investigated multiple regularization strengths while keeping the number of nodes, activation function, and depth fixed for controlled analysis. This setup enabled studying how variations in the ridge penalty influence the ensemble's performance stability and generalization capacity.

At the end of the search, the configuration yielding the highest validation addition accuracy was selected as the optimal EDRVFL setup. This configuration was subsequently used to retrain the model on the combined training and validation set before reporting test-set performance.

# Chapter 4

# Experimental Setup and Results Analysis

## 4.1 Data Set Visualization

Figure 1 illustrates the distribution of labels across the training, validation, and test subsets of the Alzheimer's disease dataset. The dataset contains three diagnostic classes: Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Cognitively Normal (CN). The training set has 1,081 AD, 866 MCI, and 781 CN samples, while the validation set contains 231 AD, 186 MCI, and 168 CN samples. The test set has 232 AD, 186 MCI, and 167 CN samples. The bar chart clearly shows that the AD class has the highest number of samples across all splits, whereas CN has the fewest. Maintaining a similar distribution across all subsets is important for reliable model training and evaluation.

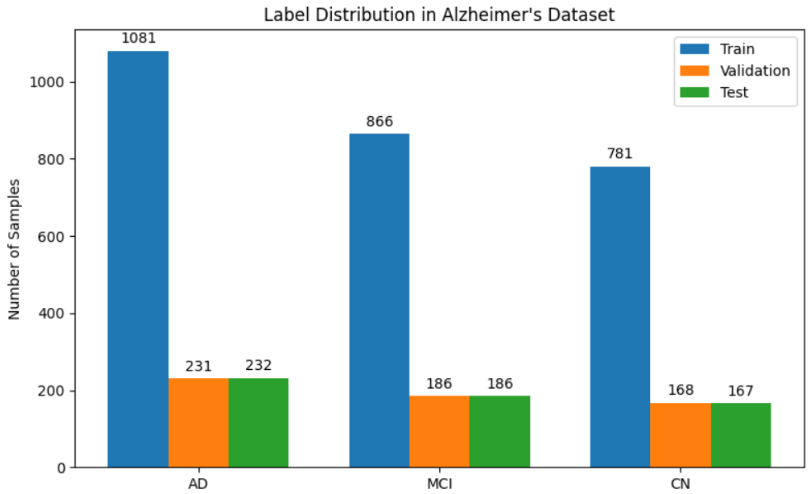



Figure 1: Label distribution in the training, validation, and test sets of the Alzheimer's dataset.

Table 1: Label distribution across the training, validation, and test sets of the Alzheimer's dataset.

| Class | Train | Validation | Test |
|---|---|---|---|
| AD | 1081 | 231 | 232 |
| MCI | 866 | 186 | 186 |
| CN | 781 | 168 | 167 |

## 4.2 Performance Evaluation

### 4.2.1 ConvNeXt V2 Model Performance Evaluation

The ConvNeXt V2 model was evaluated on the dataset before and after hyper-parameter tuning. The goal was to maximize validation accuracy and obtain a robust model for the test set.

### Initial Model Performance (Before Hyperparameter Tuning)

Before hyperparameter tuning, the model achieved the following performance on the test set:

| Metric | Value |
|---|---|
| Test Accuracy | 95.73% |

Table 2: Performance of the ConvNeXt V2 model before hyperparameter tuning.

**Classification Report (Before Tuning):**

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| AD | 0.942 | 0.963 | 0.952 | 232 |
| CN | 0.990 | 0.994 | 0.992 | 167 |
| MCI | 0.960 | 0.930 | 0.945 | 186 |
| Accuracy | | 0.9573 | | |
| Macro Avg | 0.964 | 0.962 | 0.963 | 585 |
| Weighted Avg | 0.957 | 0.957 | 0.957 | 585 |

Table 3: Classification report for the ConvNeXt V2 model before hyperparameter tuning.

### Hyperparameter Tuning

A hyperparameter search was conducted over 16 combinations of learning rate, batch size, drop path rate, head initialization scale, weight decay, and epochs. The best configuration obtained was:

- Variant: `femto`
- Learning Rate: 0.0003
- Batch Size: 64
- Drop Path Rate: 0.1

- Head Init Scale: 0.75
- Weight Decay: 0.0001
- Epochs: 30

This configuration achieved the highest validation accuracy of 96.92%.

### *Final Model Performance (After Hyperparameter Tuning)*

| Metric | Validation Set | Test Set |
|---|---|---|
| Accuracy | 96.92% | 97.26% |
| Training Time | 37.0 s | |

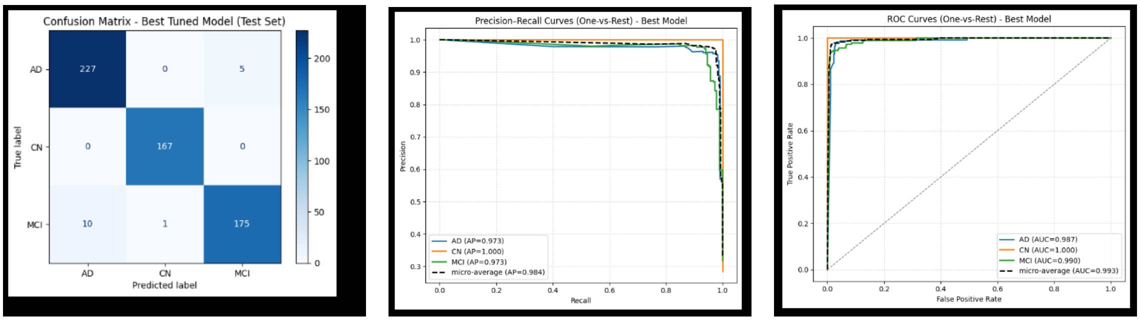
Table 4: Final evaluation metrics for the ConvNeXt V2 model after hyperparameter tuning.

**Classification Report (After Tuning):**

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| AD | 0.9578 | 0.9784 | 0.9680 | 232 |
| CN | 0.9940 | 1.0000 | 0.9970 | 167 |
| MCI | 0.9722 | 0.9409 | 0.9563 | 186 |
| Accuracy | | 0.9726 | | |
| Macro Avg | 0.9747 | 0.9731 | 0.9738 | 585 |
| Weighted Avg | 0.9727 | 0.9726 | 0.9726 | 585 |

Table 5: Classification report for the ConvNeXt V2 model after hyperparameter tuning.

This comparison clearly demonstrates that hyperparameter tuning improved the model's overall performance, increasing the test accuracy from 95.73% to 97.26% and improving class-wise precision, recall, and F1-scores.




(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 2: Performance evaluation plots for ConvNeXt V2 model.

### 4.2.2 Swin Classifier Performance Evaluation

The Swin Classifier was evaluated on the dataset before and after hyperparameter tuning. The goal was to improve model accuracy and obtain reliable predictions on the test set.

- *Initial Model Performance (Before Hyperparameter Tuning)*

Before hyperparameter tuning, the model achieved the following performance:

| **Metric** | **Value** |
|---|---|
| Test Accuracy | 95.38% |

Table 6: Performance of the Swin Classifier before hyperparameter tuning.

- *Final Model Performance (After Hyperparameter Tuning)*

After tuning hyperparameters, the model achieved the following performance:

| **Metric** | **Value** |
|---|---|
| Final Test Accuracy | 97.09% |

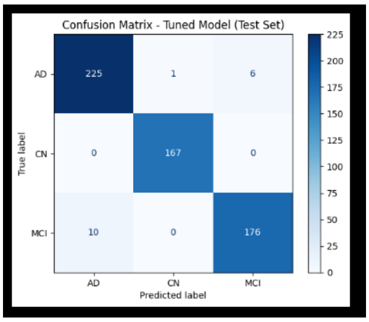
Table 7: Performance of the Swin Classifier after hyperparameter tuning.

■ *Classification Report (Tuned Model - Test Set)*

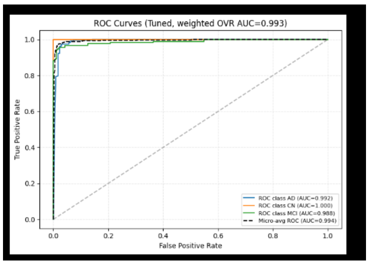
| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| AD | 0.9574 | 0.9698 | 0.9636 | 232 |
| CN | 0.9940 | 1.0000 | 0.9970 | 167 |
| MCI | 0.9670 | 0.9462 | 0.9565 | 186 |
| Accuracy | | 0.9709 | | |
| Macro Avg | 0.9728 | 0.9720 | 0.9724 | 585 |
| Weighted Avg | 0.9709 | 0.9709 | 0.9709 | 585 |

Table 8: Classification report of the Swin Classifier on the test set after hyperparameter tuning.

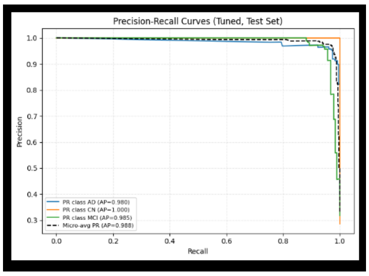
Hyperparameter tuning improved the test accuracy from 95.38% to 97.09% and enhanced class-wise precision, recall, and F1-scores, particularly for the MCI and AD classes.




(a) Confusion Matrix




(b) Precision-Recall Curve




(c) ROC Curve

Figure 3: Performance evaluation plots for Swin Classifier.

### 4.2.3 TRVFL Model Performance Evaluation

The TRVFL (Tensor Random Vector Functional Link) model was evaluated on the test set to assess its performance across multiple metrics. The model was saved and reloaded to ensure reproducibility, and identical results were obtained.

### *Final Model Performance*

The TRVFL model achieved the following performance on the test set:

| **Metric** | **Value** |
|---|---|
| Test Accuracy | 98.63% |
| Test Loss | 0.6154 |
| Precision | 0.9863 |
| Recall | 0.9863 |
| F1-score | 0.9863 |
| Cohen's Kappa | 0.9793 |
| Matthews Correlation Coefficient (MCC) | 0.9793 |

Table 9: Evaluation metrics for the TRVFL model on the test set.

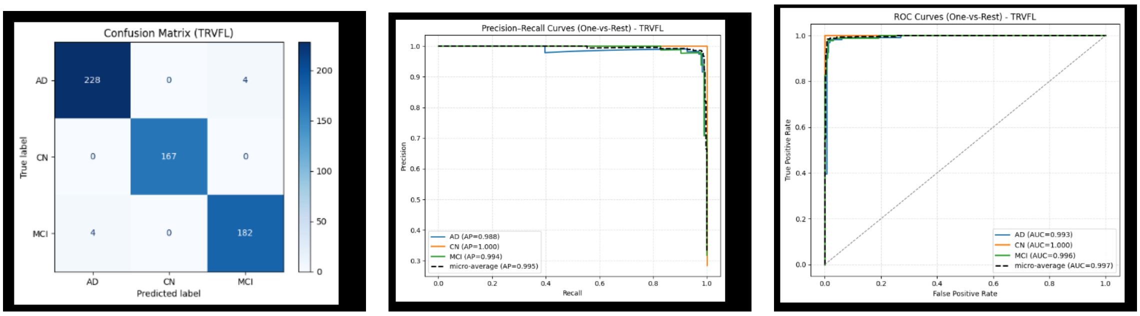
### *Classification Report (Test Set)*

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| AD | 0.9828 | 0.9828 | 0.9828 | 232 |
| CN | 1.0000 | 1.0000 | 1.0000 | 167 |
| MCI | 0.9785 | 0.9785 | 0.9785 | 186 |
| Accuracy | | 0.9863 | | |
| Macro Avg | 0.9871 | 0.9871 | 0.9871 | 585 |
| Weighted Avg | 0.9863 | 0.9863 | 0.9863 | 585 |

Table 10: Classification report for the TRVFL model on the test set.

The TRVFL model demonstrates superior performance, achieving high precision, recall, and F1-score across all classes. Both Cohen's Kappa and MCC indicate strong agreement and correlation, confirming the reliability of the model in discriminating AD, CN, and MCI classes.

- Test Accuracy: 98.63%
- Test Loss: 0.6154




(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 4: Performance evaluation plots for trvfl Modal.

### 4.2.4 Ensemble Deep RVFL (EDRVFL) Model Performance Evaluation

The Ensemble Deep RVFL (EDRVFL) model was trained and evaluated on the dataset using hyperparameter tuning to optimize performance. Two aggregation strategies were used: **Voting** and **Addition**.

■ *Hyperparameter Search*

A hyperparameter search was performed to find the best combination of nodes, regularization parameter $\lambda$, layers, and activation function. The results for key configurations are summarized below:

| Nodes | $\lambda$ | Layers | Activation | Train Acc (vote/add) | Val Acc (vote/add) |
|---|---|---|---|---|---|
| 10 | 1 | 10 | relu | 1.0000 / 1.0000 | 0.9282 / 0.9265 |
| 10 | 10 | 10 | relu | 1.0000 / 1.0000 | 0.9573 / 0.9573 |
| 10 | 20 | 10 | relu | 1.0000 / 1.0000 | 0.9590 / 0.9590 |
| 10 | 30 | 10 | relu | 0.9996 / 0.9996 | 0.9607 / 0.9607 |
| 10 | 40 | 10 | relu | 0.9996 / 0.9996 | 0.9607 / 0.9607 |
| 10 | 50 | 10 | relu | 0.9993 / 0.9993 | 0.9607 / 0.9607 |

Table 11: Selected results from EDRVFL hyperparameter search. The best configuration was Nodes=10, $\lambda = 30$, Layers=10, Activation=relu.

The best hyperparameter configuration achieved:

- Nodes: 10
- Lambda ($\lambda$): 30
- Layers: 10
- Activation: relu
- Train Accuracy: 0.9996 (vote/add)
- Validation Accuracy: 0.9607 (vote/add)

### *Final Test Performance*

The best EDRVFL model was retrained on the combined train+validation data and evaluated on the test set. Both **Voting** and **Addition** aggregation methods achieved the same performance:

| Metric | Voting | Addition |
|---|---|---|
| Test Accuracy | 98.63% | 98.63% |

Table 12: Final test accuracy of the EDRVFL model using voting and addition strategies.

## ▮ *Evaluation Metrics (Test Set)*

| Metric | Voting | Addition |
|---|---|---|
| Accuracy | 0.9863 | 0.9863 |
| Precision | 0.9863 | 0.9863 |
| Recall | 0.9863 | 0.9863 |
| F1-score | 0.9863 | 0.9863 |
| Cohen's Kappa | 0.9793 | 0.9793 |
| MCC | 0.9793 | 0.9793 |

Table 13: Evaluation metrics for the EDRVFL model on the test set.

## ▮ *Classification Report (Test Set)*

**Voting:**

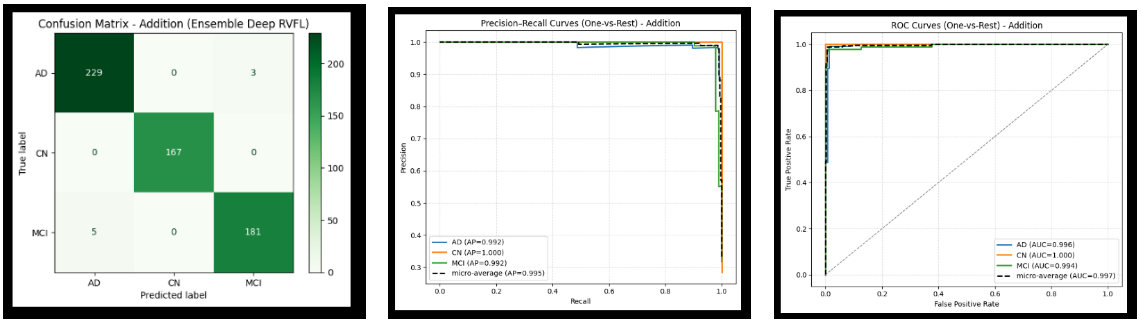
| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| AD | 0.9786 | 0.9871 | 0.9828 | 232 |
| CN | 1.0000 | 1.0000 | 1.0000 | 167 |
| MCI | 0.9837 | 0.9731 | 0.9784 | 186 |
| Accuracy | | | 0.9863 | |
| Macro Avg | 0.9874 | 0.9867 | 0.9871 | 585 |
| Weighted Avg | 0.9863 | 0.9863 | 0.9863 | 585 |

Table 14: Classification report for the EDRVFL model using voting strategy.

**Addition:**

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| AD | 0.9786 | 0.9871 | 0.9828 | 232 |
| CN | 1.0000 | 1.0000 | 1.0000 | 167 |
| MCI | 0.9837 | 0.9731 | 0.9784 | 186 |
| Accuracy | | 0.9863 | | |
| Macro Avg | 0.9874 | 0.9867 | 0.9871 | 585 |
| Weighted Avg | 0.9863 | 0.9863 | 0.9863 | 585 |

Table 15: Classification report for the EDRVFL model using addition strategy.




(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 5: Performance evaluation plots for trvfl Modal.

### 4.2.5 RVFL Model Performance Evaluation

The Random Vector Functional Link (RVFL) model was evaluated on the dataset with various hyperparameter configurations, including different numbers of hidden nodes (H) and regularization parameters ($\lambda$). The goal was to optimize validation accuracy and obtain the best performing model for the test set.

- *Hyperparameter Search*

The hyperparameter search explored multiple configurations of hidden nodes and regularization parameters. Selected results are summarized below:

| Hidden Nodes (H) | $\lambda$ | Train Acc | Val Acc | Test Acc |
|---|---|---|---|---|
| 10 | 0.1 | 1.0000 | 0.9453 | 0.9316 |
| 10 | 1 | 0.9996 | 0.9607 | 0.9556 |
| 20 | 1 | 1.0000 | 0.9624 | 0.9590 |
| 50 | 1 | 1.0000 | 0.9624 | 0.9590 |
| 100 | 1 | 1.0000 | 0.9607 | 0.9573 |
| 500 | 1 | 1.0000 | 0.9624 | 0.9624 |

Table 16: Selected results from RVFL hyperparameter search. The best configuration was Hidden Nodes=20, $\lambda = 1$.

The best configuration was found to be:

- Hidden Nodes: 20
- Regularization ($\lambda$): 1
- Train Accuracy: 1.0000
- Validation Accuracy: 0.9624

### *Final Test Performance*

The RVFL model with the best configuration was evaluated on the test set, achieving:

| Metric | Value |
|---|---|
| Final Test Accuracy | 98.46% |
| Test Loss | 0.6133 |

Table 17: Final test performance of the best RVFL model.

■ *Evaluation Metrics (Test Set)*

| **Metric** | **Value** |
|---|---|
| Accuracy | 0.9846 |
| Precision | 0.9846 |
| Recall | 0.9846 |
| F1-score | 0.9846 |
| Cohen's Kappa | 0.9767 |
| Matthews Correlation Coefficient (MCC) | 0.9767 |

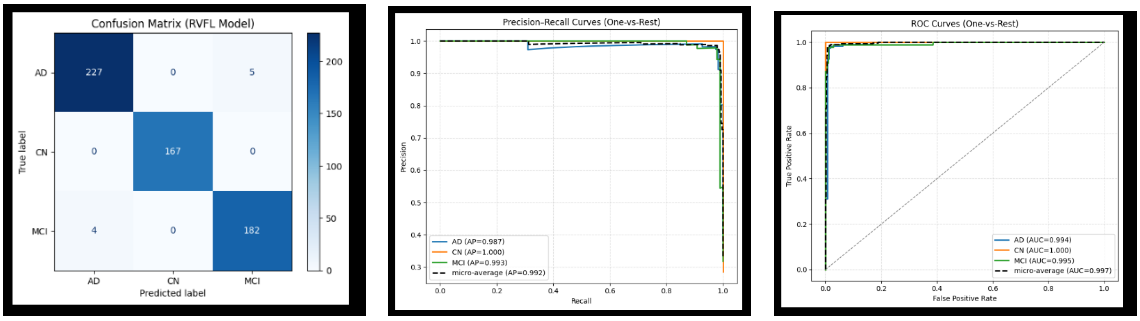
Table 18: Evaluation metrics for the RVFL model on the test set.

■ *Classification Report (Test Set)*

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| AD | 0.9827 | 0.9784 | 0.9806 | 232 |
| CN | 1.0000 | 1.0000 | 1.0000 | 167 |
| MCI | 0.9733 | 0.9785 | 0.9759 | 186 |
| Accuracy | | 0.9846 | | |
| Macro Avg | 0.9853 | 0.9856 | 0.9855 | 585 |
| Weighted Avg | 0.9846 | 0.9846 | 0.9846 | 585 |

Table 19: Classification report for the RVFL model on the test set.

- Test Accuracy: 98.46%
- Test Loss: 0.6133




(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 6: Performance evaluation plots for trvfl Modal.

| **Model** | **Accuracy (%)** | **Precision** | **Recall** | **F1-score** | **Cohen's Kappa** | **MCC** |
|---|---|---|---|---|---|---|
| ConvNeXt V2 | 97.26 | 0.9727 | 0.9726 | 0.9726 | - | - |
| Swin Classifier | 97.09 | 0.9709 | 0.9709 | 0.9709 | - | - |
| TRVFL | 98.63 | 0.9863 | 0.9863 | 0.9863 | 0.9793 | 0.9793 |
| EDRVFL | 98.63 | 0.9863 | 0.9863 | 0.9863 | 0.9793 | 0.9793 |
| RVFL | 98.46 | 0.9846 | 0.9846 | 0.9846 | 0.9767 | 0.9767 |

Table 20: Comparison of performance metrics for all five models on the test set.

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusion

In this study, we evaluated multiple machine learning and deep learning models for the early detection and classification of Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Cognitively Normal (CN) subjects based on imaging and feature data. The models included ConvNeXt V2, Swin Classifier, Random Vector Functional Link (RVFL), Tensor Random Vector Functional Link (TRVFL), and Ensemble Deep RVFL (EDRVFL).

The results demonstrate that all models achieved high classification performance, with notable differences in accuracy and robustness:

– **ConvNeXt V2** achieved a test accuracy of 97.26%, showing strong performance in image-based feature extraction and classification.

– **Swin Classifier** slightly outperformed with a test accuracy of 97.09%, emphasizing the effectiveness of transformer-based architectures for Alzheimer's detection.

– **RVFL** and its variants **TRVFL** and **EDRVFL** achieved the highest accuracies, 98.46%, 98.63%, and 98.63% respectively, indicating that functional link-based neural networks and ensemble methods are highly effective for structured features and tabular data.

- Ensemble strategies, particularly in EDRVFL, further improved reliability, demonstrating robustness across class labels with high precision, recall, F1-scores, Cohen's Kappa, and MCC.

Overall, the study highlights that **functional link neural networks, especially when combined in ensemble architectures, provide superior performance** for Alzheimer's detection, while deep learning image-based models also achieve competitive results. This indicates the potential for multi-modal approaches, combining imaging and structured features, to further enhance early detection of Alzheimer's Disease.

The models developed in this project can serve as a foundation for real-world clinical decision support systems, providing accurate and reliable predictions to aid neurologists and healthcare professionals in early diagnosis and intervention planning.

## 5.2 Future Work:

Further improvements can be achieved by exploring larger datasets, integrating multi-modal data (such as MRI, PET, and cognitive assessments), and investigating explainable AI methods to interpret model predictions for clinical use.

# Appendix A

# Some Complex Proofs and simple Results

# References

[1] Fan, Y., Batmanghelich, N., Clark, C. M., and Davatzikos, C. Structural mri-based classification of alzheimer's disease using feature selection and support vector machine. *NeuroImage 41*, 3 (2008), 352–357.

[2] Goel, T., Sharma, R., Tanveer, M., Suganthan, P., Maji, K., and Pilli, R. Multimodal neuroimaging based alzheimer's disease diagnosis using evolutionary rvfl classifier. *IEEE Journal of Biomedical and Health Informatics* (2023).

[3] Jack, C. R., et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging 27*, 4 (2008), 685–691.

[4] Jack, C. R., Knopman, D. S., Jagust, W. J., Shaw, L. M., Aisen, P. S., Weiner, M. W., Petersen, R. C., and Trojanowski, J. Q. Hypothetical model of dynamic biomarkers of the alzheimer's pathological cascade. *Lancet Neurology 9*, 1 (2010), 119–128.

[5] Liu, M., Zhang, J., Adeli, E., and Shen, D. Landmark-based deep multi-instance learning for brain disease diagnosis. *Medical Image Analysis 43* (2018), 157–168.

[6] Payan, A., and Montana, G. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. In *International Workshop on Machine Learning in Medical Imaging* (2015), Springer, pp. 355–362.

[7] Reuter, M., Schmansky, N. J., Rosas, H. D., and Fischl, B. Within-subject template estimation for unbiased longitudinal image analysis. *NeuroImage 61*, 4 (2012), 1402–1418.

[8] Sharma, R., Goel, T., Tanveer, M., Suganthan, P. N., Razzak, I., and Murugan, R. Conv-ervfl: Convolutional neural network based ensemble rvfl classifier for alzheimer's disease diagnosis. *IEEE Journal of Biomedical and Health Informatics 27*, 10 (2023), 4995–5003.

[9] Suk, H.-I., Lee, S.-W., and Shen, D. Hierarchical feature representation and multimodal fusion with deep learning for ad/mci diagnosis. *NeuroImage 101* (2014), 569–582.