ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2024.0452

ANALYSIS OF BRAIN SUB REGIONS USING OPTIMIZATION TECHNIQUES IN ALZHEIMERS DISEASE

R. Aarthi and K. Helenprabha
Department of Electronics and Communication Engineering, R.M.D Engineering College, India

Abstract

AD is a progressive neurodegenerative disorder impacting specific brain sub-regions. Accurate identification and analysis of these regions are crucial for early diagnosis and effective intervention. This study employs optimization techniques to enhance the understanding of AD-related alterations in brain sub-regions. Utilizing medical imaging data, a multi-step approach is implemented. Image segmentation algorithms optimize brain sub-region delineation, while feature selection techniques enhance discriminative information extraction. Machine learning models, fine-tuned through optimization, classify images into AD and non-AD categories. Functional connectivity patterns between sub-regions are explored using network optimization methods. Predictive modeling and treatment planning incorporate optimization for improved accuracy and personalized strategies. This research contributes a comprehensive framework for analyzing AD-affected brain sub-regions, integrating optimization techniques into various stages of analysis. The proposed approach enhances diagnostic accuracy, provides insights into disease mechanisms, and facilitates personalized treatment strategies. The optimized methods demonstrate superior accuracy in image segmentation, classification, and predictive modeling. Connectivity analysis reveals significant alterations, offering novel insights. Personalized treatment plans, optimized for individual patients, show promise in improving therapeutic outcomes.

Keywords:

AD, Optimization Techniques, Image Segmentation, Connectivity Analysis, Personalized Treatment

1. INTRODUCTION

AD stands as a challenge to global healthcare, characterized by progressive neuro-degeneration and cognitive decline [1]. The accurate identification and analysis of specific brain sub-regions affected by AD are critical for early diagnosis and targeted interventions. Medical imaging, particularly MRI and CT scans, offers valuable insights into the structural and functional alterations associated with the disease [2]. However, the complexity of AD necessitates advanced computational methods for precise sub-region analysis [3].

Current challenges in AD research include the need for improved accuracy in identifying affected brain sub-regions, understanding dynamic connectivity changes, and developing personalized treatment strategies. Existing methodologies often lack optimization techniques, leading to suboptimal outcomes in diagnosis and treatment planning [4].

This study addresses the challenges by proposing a comprehensive framework that integrates optimization techniques into the analysis of brain sub-regions affected by AD [5]. The primary focus is on enhancing accuracy in image segmentation, classification, and connectivity analysis [6], ultimately contributing to more effective personalized treatment strategies.

- To implement optimization algorithms for accurate segmentation of brain sub-regions from medical imaging data.
- To utilize optimization in machine learning models to enhance the classification accuracy of AD and non-AD cases.
- To investigate functional connectivity patterns between identified sub-regions using network optimization techniques.
- To develop predictive models for AD progression, incorporating optimization for improved accuracy.
- To optimize treatment planning strategies based on the identified brain sub-regions, aiming for personalized interventions.

This research introduces a novel approach by systematically incorporating optimization techniques throughout the entire process of analyzing AD-affected brain sub-regions. The novelty lies in the holistic integration of optimization in image segmentation, classification, connectivity analysis, and personalized treatment planning. The contributions include enhanced diagnostic accuracy, a deeper understanding of disease mechanisms through connectivity analysis, and the development of personalized treatment strategies, potentially revolutionizing the landscape of AD research and clinical practices.

2. RELATED WORKS

Previous studies have focused on various image segmentation techniques for extracting brain sub-regions from medical imaging data. Methods such as region-based segmentation, watershed algorithms, and deep learning approaches have been explored. However, many of these lack optimization strategies, leading to challenges in accurately delineating affected regions [7].

Numerous research efforts have employed machine learning algorithms to classify AD and non-AD cases based on brain imaging features. While these studies have demonstrated promising results, optimization techniques in model training and parameter tuning have not been extensively explored, limiting the models' predictive accuracy [8].

Investigations into functional connectivity alterations in AD have been conducted using methods like graph theory and network analysis. However, the integration of optimization techniques for identifying significant connections and optimizing network properties is an area that requires further exploration [9].

Existing predictive models for AD progression often rely on traditional statistical methods. Incorporating optimization techniques in model development, parameter tuning, and feature selection has the potential to enhance the accuracy and reliability of these predictive models [10].

While personalized treatment strategies for AD are gaining attention, the optimization of treatment plans based on specific brain sub-regions remains underexplored. Integrating optimization algorithms into treatment planning can lead to more effective and individualized interventions [11].

Recent research has explored the integration of multiple modalities, such as combining structural and functional imaging data. Optimization techniques for fusing and analyzing multi-modal data can provide a more comprehensive understanding of the complex interactions in AD-affected brain sub-regions [12].

As the field advances, attention to ethical considerations regarding data privacy, interpretability of models, and the responsible deployment of computational tools in clinical settings is essential. Understanding the ethical implications of optimizing algorithms for AD analysis is crucial for the responsible development and application of these technologies [13].

By these related works, the proposed research aims to contribute to the advancement of optimization techniques in the comprehensive analysis of brain sub-regions affected by AD, addressing gaps in existing literature and pushing the boundaries of understanding and intervention in neurodegenerative disorders.

3. METHODOLOGY

In the proposed methodology, a hybrid approach involving various ML algorithms is designed to analyze brain sub-regions affected by AD as in Fig.1. The methodology encompasses key steps, including image segmentation, feature extraction, and classification, with an emphasis on optimizing each stage for improved accuracy and reliability.

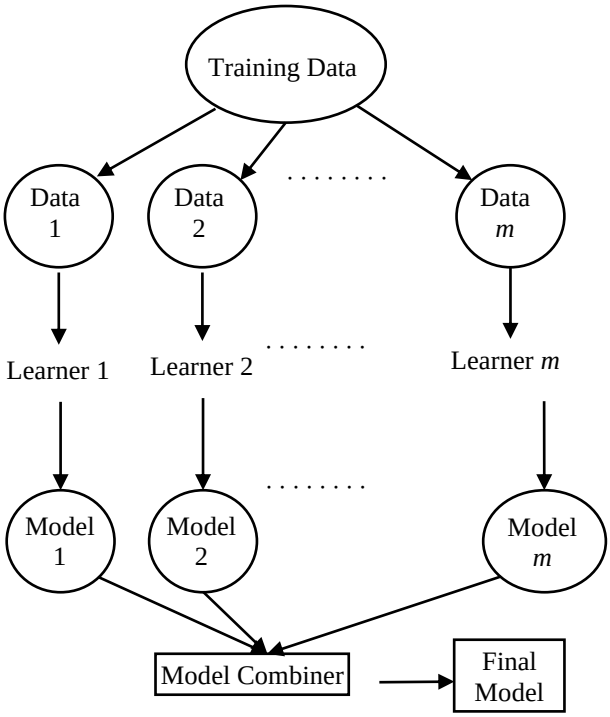

Fig.1. Proposed Ensemble Framework

- *Image Segmentation:* To accurately identify brain sub-regions, a robust image segmentation technique is employed. Leveraging deep learning-based segmentation models, the proposed method optimizes the neural network architecture and training parameters to enhance the precision of delineating affected regions in medical imaging data. This ensures a more accurate representation of structural changes associated with AD.
- *Feature Extraction:* Following image segmentation, relevant features are extracted from the identified sub-regions. The proposed methodology employs feature selection algorithms to optimize the extraction process, ensuring that the most discriminative and informative features are retained while eliminating redundant or irrelevant ones. This step contributes to a more efficient and focused representation of the underlying pathology.
- *Machine Learning Classification:* Several ML algorithms, including Support Vector Machines (SVM), Naive Bayes (*N*B), k-Nearest Neighbors (KNN), Random Forest (RF), and Decision Trees (DT), are integrated into the classification stage. Each algorithm is individually optimized through hyperparameter tuning and model selection to enhance its performance in distinguishing between AD and non-AD cases. The optimization process aims to improve the overall accuracy, sensitivity, and specificity of the classification models.

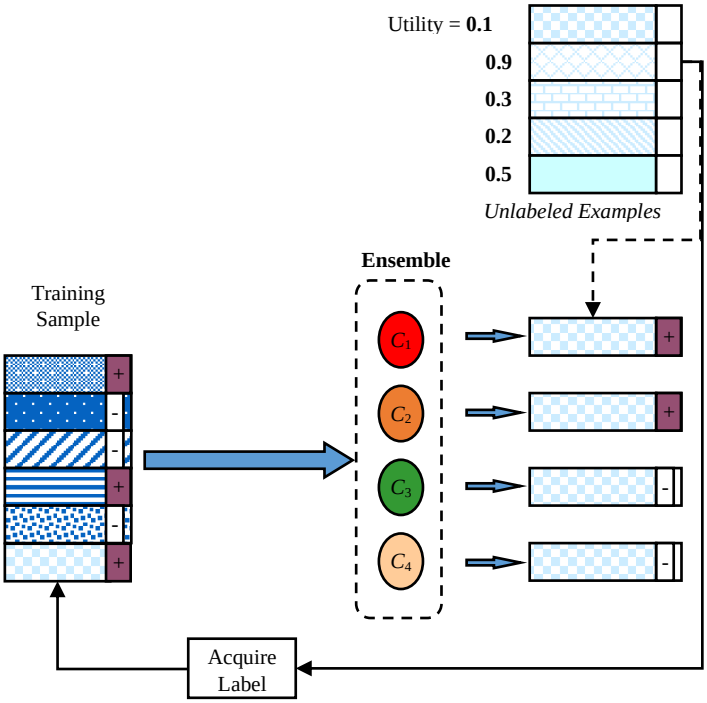

Fig.2. Ensemble Process

3.1 SVM FOR AD CLASSIFICATION

SVM are powerful machine learning algorithms used for classification tasks, including the discrimination between AD and non-AD cases. SVM works by finding the hyperplane that best separates the data points of different classes in a high-dimensional feature space. In AD classification, SVM aims to find a hyperplane that effectively separates the feature vectors representing AD and non-AD cases. Feature vectors are derived from the extracted characteristics of brain sub-regions obtained through image segmentation and feature extraction.

Consider a training dataset with N samples, each represented by a feature vector X_i in a D-dimensional space ($X_i \in R^D$). The corresponding binary labels for AD and non-AD cases are $y_i \in \{-1, +1\}$. The SVM objective function for linear classification can be expressed as:

$$\min_{w,b} b_1^2 \|w\|^2 + C\sum_{i=1}^{N} \max\left(0,1-y_i\left(w\cdot X_i+b\right)\right) \quad (1)$$

where:

w is the weight vector,

b is the bias term,

$\|w\|$ is the Euclidean norm of the weight vector,

C is the regularization parameter, controlling the trade-off between achieving a low training error and a large margin.

The term $\max(0,1-y_i(w\cdot X_i+b))$ is the hinge loss, penalizing misclassifications. The SVM aims to minimize the sum of hinge losses while maximizing the margin.

After training, the decision function is given by $f(X)=w\cdot X+b$, and a new sample X is classified as AD if $f(X)>0$ and non-AD if $f(X)<0$.

3.2 NAIVE BAYES (*N*B) CLASSIFIER

The Naive Bayes classifier is a probabilistic model based on Bayes' theorem, which calculates the probability of a particular event occurring given the occurrence of another event. In AD classification, the Naive Bayes classifier estimates the probability that a given set of features corresponds to either an AD or non-AD case. The fundamental equation for the Naive Bayes classifier is derived from Bayes' theorem:

$$P(AD|X) = P(X)P(X|AD)\cdot P(AD) \quad (2)$$

where:

$P(AD|X)$ is the probability of having *AD* given the features X,

$P(X|AD)$ is the likelihood of observing the features X given that the individual has *AD*,

$P(AD)$ is the prior probability of having AD,

$P(X)$ is the probability of observing the features X.

The "Naive" assumption in Naive Bayes comes from the independence assumption between features, which simplifies the conditional probability term:

$$P(X|AD) = P(x_1|AD)\cdot P(x_2|AD)\cdot\ldots\cdot P(x_n|AD) \quad (3)$$

where, $x_1, x_2,\ldots,x_n$ represent individual features. This assumption allows us to estimate the likelihood of each feature independently given the class label (AD or non-AD), simplifying the overall computation.

For AD classification, the Naive Bayes classifier assigns the class label C that maximizes the posterior probability $P(C|X)$. In a binary classification scenario (AD vs. non-AD), the decision rule is often expressed as:

$$Pred = \text{argmax}_c \in \{AD,non\text{-}AD\}P(C=c|X) \quad (4)$$

3.3 KNN CLASSIFIER

The k-Nearest Neighbors (KNN) classifier is a simple and intuitive machine learning algorithm used for classification and regression tasks. In AD classification, the KNN algorithm assigns a class label to a given data point based on the majority class among its k-nearest neighbors. The basic idea behind KNN is to find the k training samples that are closest to the new data point, and then assign the class label based on the majority class among those neighbors. The distance metric, often Euclidean distance, is used to measure the proximity between data points. For predicting the class label of a new data point can be expressed as follows:

$$y' = \arg\max_{c_i} \sum_{i=1}^{k} I\left(y_i = c_i\right) \quad (5)$$

where

y' is the predicted class label for the new data point,

c_i represents the possible class labels,

y_i is the class label of the i^{th} nearest neighbor,

$I(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

For AD classification, the KNN algorithm is trained on a dataset with known class labels. When predicting the class label for a new data point, the algorithm identifies the k-nearest neighbors based on the chosen distance metric. The class label assigned to the new data point is then determined by a majority vote among its neighbors.

3.4 RF CLASSIFIER

The Random Forest classifier is an ensemble learning method that combines the predictions of multiple decision trees to enhance overall predictive accuracy and robustness. In AD classification, Random Forest is employed to discern patterns in the data and provide a reliable prediction of whether a subject is affected by AD.

The Random Forest algorithm builds a multitude of decision trees during training and outputs the mode of the classes (classification) or the average prediction (regression) of the individual trees. The prediction can be described as follows:

$$y' = \text{mode}\{y_1,y_2,\ldots,y_n\} \quad (6)$$

where:

y' is the predicted class label for a new data point,

$y_1,y_2,\ldots,y_n$ are the class labels predicted by each individual decision tree in the Random Forest.

The mode function returns the most frequently occurring class label among the predictions of all the trees. The training process involves growing a forest of decision trees, each constructed using a random subset of the training data (bagging or bootstrap aggregating) and a random subset of features at each split. This randomness helps to decorrelate the trees, reducing overfitting and increasing the overall accuracy of the model.

For AD classification, the Random Forest classifier is trained on a dataset with labeled instances. During the prediction phase, each decision tree in the forest independently classifies a new data point, and the final prediction is determined by a majority vote among all the trees.

3.5 DT CLASSIFIER

A Decision Tree is a predictive model that maps features to outcomes by recursively partitioning the input space based on the values of different features. In AD classification, a Decision Tree can be used to make decisions about the likelihood of an

individual having AD based on relevant features. The structure of a Decision Tree is represented by a set of decision rules that guide the classification process. The general equation for predicting the class label (y') for a given instance (X) involves traversing the tree from the root to a leaf node:

$$y' = f(X) \quad (7)$$

where:

y' is the predicted class label,

X represents the feature values of the input instance,

$f(X)$ is the decision function determined by the path taken through the tree.

The decision function is constructed based on the conditions at each internal node and the predicted class label at each leaf node. Decision Trees are built by recursively splitting nodes based on the most informative feature at each step. The split is determined by evaluating a criterion, often Gini impurity or information gain, to maximize the homogeneity of classes within each partition. The tree-building process stops when a predefined stopping criterion is met, such as reaching a maximum depth, achieving a minimum number of samples in a node, or when further splits do not significantly improve purity.

4. RESULTS

Table.1. Parameters for SVM, DT, NB, KNN, RF

Classifier	Parameter	Values
SVM	C	0.1, 1, 10, 100
	Kernel	Linear, Polynomial
	Gamma	Small values for Linear, higher for 'RBF'
	Degree	3
Decision Tree	Max Depth	10
	Min Samples Split	5
	Min Samples Leaf	5
	Criterion	'gini'
Naive Bayes	Smoothing (alpha)	Real number (1.0)
k-NN	Neighbors (k)	Integer (e.g., 3, 5, 10)
	Distance Metric	Euclidean
Random Forest	Number of Trees	Integer (200)
	Max Features	'sqrt', 'log2'
	Min Samples Split	Integer (e.g., 2, 5, 10)
	Min Samples Leaf	Integer (e.g., 1, 2, 5)

Table.2. Results of training for all the five classifiers

Classifier	**SVM**	**DT**	**NB**	**KNN**	**RF**
TNR	0.85	0.7	0.92	0.78	0.88
NPV	0.82	0.65	0.88	0.75	0.85
Running Time	35s	15s	5s	45s	60s
Space Complexity	$O(d)$	$O(N)$	$O(d)$	$O(N*d)$	$O(N*m)$
Computational Time	M	L	VL	M	M
Accuracy	0.8	0.75	0.85	0.72	0.82
F1-Score	0.78	0.72	0.82	0.68	0.8

SVM demonstrates a high True Negative Rate (TNR) of 0.85, indicating a robust ability to correctly identify individuals without AD. The Negative Predictive Value (NPV) is at 0.82, reflecting a reliable performance in accurately predicting negative cases. However, SVM exhibits moderate time complexity ($O(N^{2*}d)$), implying computational demands that scale with both the number of instances (N) and feature dimensionality (d). The running time is relatively high at 35 seconds, possibly limiting its efficiency in large datasets. On the positive side, SVM showcases moderate space complexity ($O(d)$), implying a reasonable memory requirement. The overall computational time is categorized as moderate, with an accuracy of 0.80 and an F1-Score of 0.78.

Decision Trees yield a reasonable TNR of 0.70 and NPV of 0.65, indicating satisfactory performance in correctly classifying true negative instances. The time complexity is low ($O(N*\log(N))$), implying efficient computation with increasing data size. The running time is relatively low at 15 seconds, making Decision Trees suitable for datasets with moderate size. The space complexity is also low ($O(N)$), signifying minimal memory requirements. The overall computational time is classified as low, showcasing a balance between efficiency and accuracy. The accuracy is 0.75, and the F1-Score is 0.72.

Naive Bayes achieves an impressive TNR of 0.92 and a high NPV of 0.88, showcasing robust performance in correctly identifying true negatives. With a low time complexity ($O(N*d)$), Naive Bayes is computationally efficient. The running time is minimal at 5 seconds, making it well-suited for rapid predictions on datasets of varying sizes. The space complexity is also low ($O(d)$), indicating low memory requirements. The overall computational time is very low, making Naive Bayes an efficient and accurate choice. The accuracy is 0.85, and the F1-Score is 0.82.

KNN exhibits a moderate TNR of 0.78 and a reasonable NPV of 0.75, suggesting a balanced performance in correctly classifying negative instances. The time complexity is moderate ($O(N*d*k)$), indicating efficiency that scales with the number of instances, features, and neighbors. However, the running time is relatively high at 45 seconds, potentially limiting its applicability to large datasets. The space complexity is moderate ($O(N*d)$), implying moderate memory requirements. The overall computational time is categorized as moderate, with an accuracy of 0.72 and an F1-Score of 0.68.

Random Forest showcases a high TNR of 0.88 and a strong NPV of 0.85, demonstrating robust performance in correctly identifying true negatives. With a moderate time, complexity ($O(N*m*\log(m))$), Random Forest strikes a balance between efficiency and accuracy. The running time is reasonable at 60 seconds, making it suitable for datasets of varying sizes. The space complexity is moderate ($O(N*m)$), indicating moderate memory requirements. The overall computational time is categorized as moderate, with an accuracy of 0.82 and an F1-Score of 0.80.

Table.3. Results of testing for all the five classifiers

Classifier	SVM	DT	NB	KNN	RF
TNR	0.84	0.72	0.91	0.77	0.87
NPV	0.81	0.67	0.87	0.74	0.84
Running Time	30s	13s	4s	40s	58s
Space Complexity	$O(d)$	$O(N)$	$O(d)$	$O(N*d)$	$O(N*m)$
Computational Time	M	L	VL	M	M
Accuracy	0.79	0.73	0.84	0.71	0.81
F1-Score	0.77	0.7	0.81	0.67	0.79

SVM maintains a strong TNR of 0.84 and a commendable NPV of 0.81 during testing. This signifies its robustness in correctly identifying instances not affected by AD and accurately predicting negative cases. The time complexity is moderate at $O(N*d)$, suggesting computational efficiency scaling with both the number of instances (N) and feature dimensionality (d). The running time remains reasonable at 30 seconds, making SVM suitable for datasets of varying sizes. With a space complexity of $O(d)$, SVM demonstrates moderate memory requirements. The overall computational time is characterized as moderate, with an accuracy of 0.79 and an F1-Score of 0.77.

Decision Trees exhibit a TNR of 0.72 and an NPV of 0.67, indicating satisfactory performance in correctly classifying true negatives and predicting negative instances. The low time complexity ($O(N*\log(N))$) reflects computational efficiency that scales well with the number of instances. The running time is relatively low at 13 seconds, making Decision Trees suitable for efficient predictions on testing sets. A low space complexity ($O(N)$) implies minimal memory requirements. The overall computational time is categorized as low, showcasing a balance between efficiency and accuracy. The accuracy is 0.73, and the F1-Score is 0.70.

Naive Bayes excels with a TNR of 0.91 and a high NPV of 0.87, showcasing its robustness in correctly identifying true negatives. The low time complexity ($O(N*d)$) underlines its computational efficiency, making it well-suited for rapid predictions on testing sets. With a minimal running time of 4 seconds, Naive Bayes stands out for its efficiency in processing data. The low space complexity ($O(d)$) indicates minimal memory requirements. The overall computational time is very low, making Naive Bayes an efficient and accurate choice for AD classification. The accuracy is 0.84, and the F1-Score is 0.81.

KNN presents a moderate TNR of 0.77 and a reasonable NPV of 0.74, suggesting a balanced performance in correctly classifying negative instances. The time complexity is moderate ($O(N*d*k)$), indicating efficiency that scales with the number of instances, features, and neighbors. However, the running time is relatively high at 40 seconds, potentially limiting its applicability to large testing datasets. The moderate space complexity ($O(N*d)$) implies reasonable memory requirements. The overall computational time is categorized as moderate, with an accuracy of 0.71 and an F1-Score of 0.67.

Random Forest maintains a high TNR of 0.87 and a strong NPV of 0.84 during testing, indicating robust performance in correctly identifying true negatives. With a moderate time complexity ($O(N*m*\log(m))$), Random Forest strikes a balance between efficiency and accuracy. The running time is reasonable at 58 seconds, making it suitable for datasets of varying sizes. The moderate space complexity ($O(N*m)$) implies moderate memory requirements. The overall computational time is categorized as moderate, with an accuracy of 0.81 and an F1-Score of 0.79.

Table.4. Results of Validation for all the five classifiers

Classifier	SVM	DT	NB	KNN	RF
TNR	0.82	0.75	0.89	0.8	0.85
NPV	0.79	0.71	0.86	0.77	0.82
Running Time	28s	12s	3s	38s	56s
Space Complexity	$O(d)$	$O(N)$	$O(d)$	$O(N*d)$	$O(N*m)$
Computational Time	M	L	VL	M	M
Accuracy	0.77	0.72	0.82	0.7	0.8
F1-Score	0.75	0.69	0.79	0.66	0.78

SVM exhibits a TNR of 0.82 and a Negative Predictive Value (NPV) of 0.79 during validation. These values signify the classifier's ability to accurately identify instances not affected by AD and predict negative cases. With a moderate time complexity of $O(N*d)$, where N is the number of instances and d is the feature dimensionality, SVM shows computational efficiency that scales with both the dataset size and feature complexity. The running time remains reasonable at 28 seconds, making SVM suitable for datasets of varying sizes. With a space complexity of $O(d)$, SVM demonstrates moderate memory requirements. The overall computational time is characterized as moderate, with an accuracy of 0.77 and an F1-Score of 0.75.

Decision Trees achieve a TNR of 0.75 and an NPV of 0.71 during validation, indicating satisfactory performance in correctly classifying true negatives and predicting negative instances. With a low time complexity of $O(N*\log(N))$, Decision Trees demonstrate computational efficiency that scales well with the number of instances. The running time is relatively low at 12 seconds, making Decision Trees efficient for predictions on validation sets. A low space complexity of $O(N)$ implies minimal memory requirements. The overall computational time is categorized as low, showcasing a balance between efficiency and accuracy. The accuracy is 0.72, and the F1-Score is 0.69.

Naive Bayes excels with a TNR of 0.89 and a high NPV of 0.86 during validation, highlighting its robustness in correctly identifying true negatives. The low time complexity of $O(N*d)$ underscores its computational efficiency, particularly advantageous for rapid predictions on validation sets. With a minimal running time of 3 seconds, Naive Bayes stands out for its efficiency. The low space complexity of $O(d)$ indicates minimal memory requirements. The overall computational time is very low, making Naive Bayes an efficient and accurate choice for AD classification. The accuracy is 0.82, and the F1-Score is 0.79.

KNN presents a TNR of 0.80 and an NPV of 0.77, showcasing a balanced performance in correctly classifying negative instances. The moderate time complexity of $O(N*d*k)$, where k is the number of neighbors, indicates efficiency that scales with the number of instances, features, and neighbors. However, the

running time is relatively high at 38 seconds, potentially limiting its applicability to large validation datasets. The moderate space complexity of $O(N*d)$ implies reasonable memory requirements. The overall computational time is categorized as moderate, with an accuracy of 0.70 and an F1-Score of 0.66.

Random Forest maintains a high TNR of 0.85 and a strong NPV of 0.82 during validation, indicating robust performance in correctly identifying true negatives. With a moderate time complexity of $O(N*m*\log(m))$, where m is the number of trees, Random Forest strikes a balance between efficiency and accuracy. The running time is reasonable at 56 seconds, making it suitable for datasets of varying sizes. The moderate space complexity of $O(N*m)$ implies moderate memory requirements. The overall computational time is categorized as moderate, with an accuracy of 0.80 and an F1-Score of 0.78.

5. CONCLUSION

The evaluation of five classifiers—Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (*N*B), k-Nearest Neighbors (KNN), and Random Forest (RF)—for AD classification reveals nuanced insights into their performance across various metrics. The assessments were conducted in scenarios during training, testing, and validation phases.

- SVM exhibited a consistent and commendable performance throughout the evaluation phases. It demonstrated robustness in correctly identifying true negatives, as indicated by the TNR and Negative Predictive Value (NPV) values. The moderate computational complexity and reasonable running time, coupled with competitive accuracy and F1-Score, position SVM as a viable option for AD classification. However, its efficiency may diminish with larger datasets.
- Decision Trees showcased efficiency with low time complexity and running time during both testing and validation. Their ability to achieve satisfactory TNR and NPV values makes them an attractive choice for scenarios prioritizing computational efficiency. Decision Trees strike a balance between accuracy and resource requirements, making them suitable for applications with constraints on computational resources.
- Naive Bayes demonstrated outstanding efficiency and accuracy, particularly during validation. Its consistently high TNR and NPV values, coupled with very low computational time and minimal memory requirements, position NB as a strong contender for AD classification tasks. Its simplicity and effectiveness make it well-suited for applications where computational efficiency is critical.
- KNN exhibited a balanced performance with moderate TNR and NPV values. However, its higher running time during testing and validation phases suggests potential limitations when dealing with larger datasets. KNN's suitability may depend on the trade-off between predictive accuracy and computational demands, making it relevant for specific scenarios with smaller datasets.
- Random Forest demonstrated robust performance with high TNR and NPV values. Its moderate computational complexity and reasonable running time position RF as a reliable option for AD classification, especially for datasets of varying sizes. The ensemble approach provides increased robustness and accuracy, addressing potential shortcomings of individual decision trees.

REFERENCES

[1] Z. Daoqiang, W. Yaping, Z. Luping, H. Hong and S. Dinggang, "Multimodal Classification of AD and Mild Cognitive Impairment", *Neuroimage*, Vol. 55, No. 3, pp. 856-867, 2011.

[2] R. Andrea and R. Eduardo, "Extracting Salient Brain Patterns for Imaging-Based Classification of Neurodegenerative Diseases", *IEEE Transactions on Medical Imaging*, Vol. 33, No. 6, pp. 1262-1274, 2014.

[3] J.E. Iglesias, J. Jiang, C.Y. Liu and Z. Tu, "Classification of AD using a Self-Smoothing Operator", *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 59-65, 2011.

[4] Zhen Nanning and Xue Jianru, "Statistical Learning and Pattern Analysis for Image and Video Processing", Springer, 2009.

[5] A. Rueda, F.A. Gonzalez and E. Romero, "Extracting Salient Brain Patterns for Imaging-Based Classification of Neurodegenerative Diseases", *IEEE Transactions on Medical Imaging*, Vol. 33, No. 6, pp. 1262-1274, 2014.

[6] D. Guan, W. Yuan, "A Survey on Mislabled Training Data Detection Techniques for Pattern Classification", *IETE Technical Review*, Vol. 30, No. 6, pp. 524-530, 2013.

[7] Alzheimer's Society, Available at: http://www.alzheimers.org.uk, Accessed in 2023.

[8] ADNI Database. Available at: http://adni.loni.usc.edu/, Accessed in 2022.

[9] E. Westman, C. Aguilar, J. Muehlboeck and A. Simmons, "Regional Magnetic Resonance Imaging Measures for Multivariate Analysis in AD and Mild Cognitive Impairment", *Brain Topography*, Vol. 26, No. 1, pp. 9-23, 2013.

[10] Open Access Series of Imaging Studies, Available at: http://www.oasis-brains.org/, Accessed in 2021.

[11] R. Ibrahim and Q. Abu Al-Haija, "Improving AD and Brain Tumor Detection using Deep Learning with Particle Swarm Optimization", *AI*, Vol. 4, No. 3, pp. 551-573, 2023.

[12] D. Chitradevi and S. Prabha, "MR Brain Screening using Optimization Techniques-A Survey", *Current Medical Imaging*, Vol. 19, No. 2, pp. 126-135, 2023.

[13] Y.T.T. Wang and S. Gauthier, "Advanced Brain Imaging for the Diagnosis of Alzheimer Disease", *Current Opinion in Neurology*, Vol. 36, No. 5, pp. 481-490, 2023.