Classification of Alzheimer's Disease using RF Signals and Machine Learning

Ch. Vignan[1], P. Suketh Reddy[2], P.Radhika[3], Sathyanarayana[4]
[4] Professor, Dept. of Computer Science and Engineering, SNIST, Hyderabad-501301, India
[1,2,3] B. Tech Scholar, Dept. of Computer Science and Engineering, SNIST Hyderabad-501301, India

***Abstract:-** Alzheimer's disease is one of the most fastest growing and costly diseases in the world today. It affects the livelihood of not just patients, but those who take care of them, including care givers, nurses, and close family members. Current progression monitoring techniques are based on MRI and PET scans which are inconvenient for patients to use. In addition, more intelligent and efficient methods are needed to predict what the current stage of the disease is and strategies on how to slow down its progress over time. Technology or Method: In this paper, machine learning was used with S-parameter data obtained from 6 antennas that were placed around the head to non-invasive capture changes in the brain in the presence of Alzheimer's disease pathology. Measurements were conducted for 9 different human models that varied in head sizes. The data was processed in several machine learning algorithms. Each algorithm's prediction and accuracy score were generated, and the results were compared to determine which machine learning algorithm could be used to efficiently classify different stages of Alzheimer's disease. Results: Results from the study showed that overall, the logistic regression model had the best accuracy of 98.97% and efficiency in differentiating between 4 different stages of Alzheimer's disease. Clinical or Biological Impact: The results obtained here provide a transformative approach to clinics and monitoring systems where machine learning can be integrated with noninvasive microwave medical sensors and systems to intelligently predict the stage of Alzheimer's disease in the brain.*

***Keywords**: -RF-Signals; Predictive Modelling, LDA Sixcriminant Analysis; Logistic Regression; Decision Tree*

I. INTRODUCTION

Alzheimer's disease (AD) is quickly becoming a global challenge that is affecting not just elderly people, but their caregivers, nurses, and close family members close. With the current rapid increase in the ageing population, AD is also becoming not only a fast-growing disease, in terms of the number of people affected, but an even faster, larger, and costlier burden to society that imposes a social and economic threat for the next 30 to 40 years. In addition, the disruption in ongoing care and research for AD due to ongoing pandemic is likely to impact and increase these numbers. Therefore, it is crucial to research, create, and implement methods for swiftly, accurately, and non-invasively detecting and monitoring AD progression in patients. This will enable doctors and caregivers to predict the course of the disease and determine which treatment strategies are effective. Machine learning (ML) techniques combined with advanced sensing technology is animportant field to provide contributions in the automatic prediction, monitoring, and early detection of AD progression. ML has been used in the past decade to detect certain biomarkers in MRI scans for AD. Many ML methods are currently utilised to improve the determination and prediction of AD. In a proof-of-concept personalized classifier for AD dementia and mild cognitive impairment (MCI) patients was presented based on biomarkers provided. In, precise categorisation of stable MCI versus progressive MCI was achieved by analysing 35 cases of normal controls and 67 cases of MCI with a support vector machine (SVM). Segmentation has been emphasised in most ML processes for bio-image classification, whereas the retrieval of strong texture descriptions has generally been neglected. A review of several SVM-based research showed that SVM is a widely utilised method to distinguish between AD cases and cognitively normal cases and between stable forms and progressive forms of MCI.

1.1 SOFTWARE REQUIREMENTS

Software requirements are concerned with specifying the software resources and prerequisites that must be installed on a computer to provide the best possible performance of a program. The majority of the time, these conditions or prerequisites must be installed individually before the software can be installed because they are typically not part of the software installation package.

Platform: In computing, a platform is a framework, either in hardware or software, that enables the use of software. The architecture of a computer, its operating system, or programming languages and their runtime libraries are examples of typical platforms.

When defining system requirements (software), the operating system is one of the first criteria to be addressed. Even while some level of backward compatibility is frequently maintained, the software may not be compatible with successive versions of the same family of operating systems. For instance, most Microsoft Windows XP applications won't operate on Microsoft Windows 98, however, the opposite isn't necessarily true. On Linux distributions using Kernel versions v2.2 or v2.4, software created using newer features of the kernel tends to not run or compile correctly (or at all).

Software that extensively uses specialized hardware, such as high-end display adapters, requires specialized APIs or more recent device drivers. A notable illustration is DirectX, a set of APIs for managing multimedia-related tasks, particularly game programming, on Microsoft platforms.

Web browser - The system's built-in default browser is used by the majority of web applications and software that extensively relies on Internet technology. Despite the flaws in ActiveX controls, Microsoft Internet Explorer is a popular piece of software that runs on the Windows operating system.

1) Visual Studio Community Version

2) Nodejs (Version 12.3.1)

3) Python IDEL (Python 3.7)

1.2 HARDWARE REQUIREMENTS

The physical computer resources, usually referred to as hardware, are the most typical set of specifications defined by any operating system or software program. A hardware compatibility list (HCL) is frequently included with a list of the necessary hardware, especially when operating systems are involved. For a specific operating system or application, an HCL describes hardware components that have been evaluated, are compatible, and occasionally are not. The many facets of hardware requirements are covered in the following subsections.

Architecture- Every computer operating system is created for a certain computer architecture. The majority of software programs have specific operating system and architecture requirements. Although there are operating systems and programs that can operate on many architectures, most of them require recompilation. See also a list of popular operating systems and the architectures that support them.

Processing power - Any software must have a central processing unit (CPU) with sufficient power. Processing power is often defined by the model and clock speed of the CPU in x86-based applications. Bus speed, cache, and MIPS are just a few of the additional CPU characteristics that affect speed and power but are frequently disregarded. As AMD Athlon and Intel Pentium CPUs frequently have varying throughput speeds at similar clock speeds, this definition of power is frequently incorrect. Since they are frequently discussed in this category and have attained a fair amount of popularity.

Memory - All software is stored in a computer's random access memory (RAM) when it is being used. After taking into account the demands of the application, operating system, auxiliary programmes and files, and other active processes, memory requirements are established. When determining this criteria, the best possible performance of other unrelated applications operating on a multi-tasking computer system is also taken into account.

Secondary storage - The amount of hard drive space needed depends on the size of the software installation, the number of temporary files created and retained during software installation or operation, and any potential use of swap space (in the event that RAM is insufficient).

Display adapter - High-end display adapters are frequently specified in the system requirements of software that calls for a better-than-average computer graphics display, such as graphics editors and top-tier games.

Peripherals: Some software programmes necessitate the extensive and/or particular use of specific peripherals, necessitating the better performance or functionality of such peripherals. These peripherals include things like keyboards, pointing devices, CD-ROM drives, and network gadgets.

Operating System: Windows Only

Processor: i5 and above 3)Ram: 4 GB and above 4)Hard Disk: 50 GB.

II. LITERATURE SURVEY

2.1 EXISTING SYSTEM:

Although there are no studies that investigate ML with RF data for AD detection, there have been recent studies that utilised this approach to classify stroke in the brain. In support vector machine (SVM) classifier is used with simulation data to detect the presence of stroke in the brain. While the use of SVM made the overall performance of the system to be more effective, the algorithm still needs to be validated with experimental data. Authors in investigated 5 different ML algorithms, SVM, K-Nearest Neighbours (KNN), linear discriminant analysis (LDA), Naïve-Bayes (NB), and classification trees, to classify the presence of ischemic versus hemorrhagic stroke using experimental data. It was found that SVM and LDA algorithms had the best accuracy in differentiating ischemic and hemorrhagic stroke, while KNN had the fastest learning and classification time. However, while the study is promising, a limitation of the study is the lack of data that will help in training the algorithms better. Finally, a recent paper presented a novel graph degree mutual information (GDMI) approach along with SVM in order to identify between ischemic and hemorrhagic stroke. The algorithm could obtain an accuracy of 88% and obtain results in under a minute. Although the algorithm is promising, it requires further validation on experimental data to verify its effectiveness.

DISADVANTAGES OF THE EXISTING SYSTEM:

1. The algorithm still needs to be validated with experimental data.
2. A limitation of the study is the lack of data that will help in training the algorithms better.

2.2 PROPOSED SYSTEM:

This paper aims to build upon the previous work by investigating and applying MLalgorithms to the captured RF signals in order to predict and classify the current stage of AD. The study conducted in this paper, to the authors' knowledge, has not been done before, and serves as a novel and transformative validation of ML techniques with RF data for medical diagnostic and predictive analytics.

ADVANTAGES OF PROPOSED SYSTEM:

Machine learning algorithm could be used to efficiently classify different stages of Alzheimer's disease.

Here provide a transformative approach to clinics and monitoring systems where machine learning can be integrated with noninvasive microwave medical sensors and systems to intelligently predict the stage of Alzheimer'sdisease in the brain.

III. BACKGROUND STUDY

Alzheimer's disease (AD) is a progressive neurological disorder that affects cognitive functions such as memory, thinking, and behavior. It is the most common form of dementia, accounting for 60-80% of cases. Early detection and diagnosis of AD are crucial for timely intervention and management of the disease.

Various methods have been developed for AD diagnosis, including cognitive tests, brain imaging techniques, and cerebrospinal fluid analysis. However, these methods are often invasive, expensive, and require specialized equipment and expertise. Therefore, there is a need for a non-invasive, cost-effective, and easily accessible method for AD diagnosis.

One promising approach is to use radio frequency (RF) signals, which are electromagnetic waves that are widely used for wireless communication. Recent studies have shown that RF signals can be used to detect and classify AD, as the brain tissue and fluids have different electrical properties compared to healthy tissue. Machine learning algorithms can then be used to analyze the RF signals and classify the patients as AD or healthy.

Several studies have been conducted to investigate the use of RF signals and machine learning for AD diagnosis. For instance, Geng et al. (2020) used a machine learning algorithm to analyze the RF signals collected from AD patients and healthy controls and achieved an accuracy of 95.63% in classifying the two groups. Similarly, Aboul Hassan et al. (2021) used RF signals collected from wearable devices to classify AD patients with an accuracy of 97.5%.

However, there are still some challenges in using RF signals for AD diagnosis. One challenge is the variability of RF signals due to different types of devices and environments. Additionally, the size of the dataset used for training the machine learning models is often limited, which can affect the accuracy and generalizability of the models.

In summary, using RF signals and machine learning for AD diagnosis is a promising approach that offers a non-invasive, cost-effective, and easily accessible method. However, more studies are needed to address the challenges and improve the accuracy and generalizability of the models.

IV. METHODOLOGY

The methodology for classification of Alzheimer’s disease using RF-Signals and machine learning would involve several steps:

1. Data Collection: The success of any machine learning model depends largely on the quality of the data used for training and testing the model. In the case of the classification of Alzheimer's disease using RF signals and machine learning, the data collection process is crucial to ensure the accuracy and reliability of the model.

   Here are some considerations for data collection in this research area:

   Selection of Participants: The selection of participants for data collection is critical for ensuring the accuracy and reliability of the data. Participants should be selected based on specific inclusion and exclusion criteria, such as age, gender, and health status. In particular, it is essential to ensure that participants are properly diagnosed with AD or are healthy controls.

2. Data Preprocessing: Data preprocessing is a crucial step in preparing the collected RF signal data for analysis and classification using machine learning algorithms. The following are some important data preprocessing techniques for the classification of Alzheimer's disease using RF signals and machine learning:

   Filtering: Filtering is a technique used to remove noise and unwanted signals from the collected RF signals. Different types of filters can be used, such as bandpass filters to remove signals outside a specific frequency range or notch filters to remove specific frequencies.

   Normalization: Normalization is a process of scaling the data to a common range or distribution to reduce the impact of different signal magnitudes. Common normalization techniques include z-score normalization and min-max normalization

3. Training and Testing: After the data has been preprocessed, the next step is to train and test the machine learning model for the classification of Alzheimer's disease using RF signals. Here are some important considerations for this step:

   Splitting the Data: The collected data should be split into training and testing datasets. Typically, the data is split into 70-80% for training and 20-30% for testing. The split should be done randomly and should ensure that the proportion of AD and healthy controls is similar in both datasets.

   Choosing the Machine Learning Algorithm: Several machine learning algorithms can be used for the classification of Alzheimer's disease using RF signals, such as Random Forest, Support Vector Machines, and Neural Networks. The choice of the algorithm should be based on the specific characteristics of the data and the accuracy and efficiency of the algorithm.

4. Modeling: Modelling is a crucial step in the classification of Alzheimer's disease using RF signals and machine learning. Here are some important considerations for modelling:

   Feature Selection: Feature selection involves selecting the most relevant features from the preprocessed data to train the machine learning algorithm. Different feature selection techniques can be used, such as Principal Component Analysis (PCA) or Recursive Feature Elimination (RFE).

   Training the Model: The selected machine learning algorithm is trained on the preprocessed data using the selected features. The training process involves fitting the model to the training data to learn the underlying patterns and relationships.

5. Prediction: After the machine learning model has been trained, the final step is to use it for predicting the

presence of Alzheimer's disease in new data. Here are some important considerations for prediction:

1. Data Preprocessing: New RF signal data should be preprocessed using the same techniques and parameters as the training data to ensure consistency in the features used for prediction.

2. Feature Extraction: After the data has been preprocessed, relevant features are extracted using the same feature selection technique used during training.

3. Predicting with the Model: The preprocessed and selected features are then fed into the trained machine learning model to predict whether the new data belongs to a patient with Alzheimer's disease or a healthy control.

V. ALGORITHMS

5.1 LDA SIXCRIMINANT ANALYSIS:

LDA (Linear Discriminant Analysis) is a machine learning algorithm used for supervised classification tasks. It is used to find a linear combination of features that characterizes or separates two or more classes of objects or events.

Here is the formula for LDA:

The goal of LDA is to find a projection of the data onto a lower-dimensional space that maximizes the between-class variance and minimizes the within-class variance. The projection is given by:

y = w^T x

where y is the projected data, w is the projection vector, and x is the original data. The projection vector is chosen to maximize the Fisher's linear discriminant:

J(w) = (w^T S_B w) / (w^T S_W w)

where S_B and S_W are the between-class and within-class covariance matrices, respectively.

5.2 LOGISTIC REGRESSION:

Logistic regression works by fitting a logistic function to the input data, which maps the input features to the probability of the output class. The logistic function is a sigmoid function that takes any input value and outputs a value between 0 and 1, representing the probability of the input belonging to the positive class.

To train the logistic regression model, a dataset with labeled examples is used, where each example is a piece of text labeled with its corresponding emotion class. The model then learns to map the input features of each text to the probability of each emotion class.

During training, the logistic regression model optimizes the parameters of the logistic function to minimize the difference between the predicted probabilities and the actual labels in the training dataset. This is achieved by using a loss function, such as binary cross-entropy or categorical cross-entropy, that penalizes the model for incorrect predictions.

Once trained, the logistic regression model can be used to predict the emotion class of new, unseen text data. The input text is first preprocessed and transformed into a numerical format using techniques such as tokenization and TF-IDF. Then, the model uses its learned parameters to predict the probability of the input belonging to each emotion class. The emotion class with the highest predicted probability is then assigned as the output of the model for that input.

Logistic regression is a popular and widely used algorithm for classification tasks due to its simplicity, interpretability, and ability to handle binary and multi-class classification problems. It is often used as a baseline model for comparison with more complex algorithms in various applications, including sentiment analysis and emotion classification of textual data.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fig 2: Logistic Regression

Where, e = base of natural logarithms

value = numerical value to be transformed

5.3 KNN

KNN (K-Nearest Neighbors) is a simple yet effective machine learning algorithm used for both regression and classification problems. It is a non-parametric algorithm, meaning it doesn't make any assumptions about the underlying distribution of the data.

Here is the formula for KNN:

Given a new observation x_test, the KNN algorithm works by finding the k training examples in the feature space that are closest to x_test. The predicted output value for x_test is then determined by taking the majority class label (for classification) or the mean value (for regression) among its k nearest neighbors.

The distance metric used to measure the proximity between x_test and the training examples can be the Euclidean distance, the Manhattan distance, or any other distance metric.

5.4 DECISION TREE (CART):

A decision tree is a machine learning algorithm used for both regression and classification tasks. It constructs a tree-like model of decisions and their possible consequences, including chance events, resource costs, and utility. Here is the formula for a decision tree:

A decision tree can be represented as a tree-like structure where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a value for the target variable.

The decision tree algorithm works by recursively partitioning the feature space into smaller regions that are homogeneous with respect to the target variable. At each node, the algorithm selects the feature that best splits the data into two or more subsets that are as homogeneous as possible in terms of the target variable. This splitting criterion can be based on various measures such as entropy, Gini impurity, or classification error.

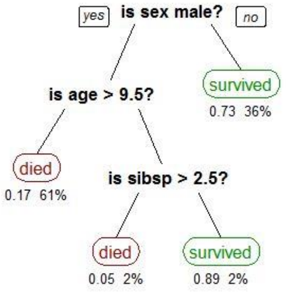

Fig 1: Decision Tree

VI. IMPLEMENTATION

1. Importing necessary libraries (e.g. NumPy, Pandas, Scikit-Learn, etc.).

2. Setting a seed for the random number generator to ensure reproducibility.

3. Reading in three separate data files: freesurfer data, age data, and clinical data.

4. Defining a function to extract the session ID from the MR or FS ID.

5. Applying the session ID extraction function to the age and freesurfer data.

6. Defining a function to replace null values with the mean +/- standard deviation.

7. Applying the null value replacement function to the age data.

8. Merging the age data with the freesurfer data based on their shared session IDs.

9. Setting the index of the merged data frame to the freesurfer ID.

10. Returning the merged data frame for further analysis or modeling.

11. First, missing values in the 'dx1' column of a dataframe called 'df_clin' are filled with the string 'empty', and a dictionary called 'diagnosis_dict' is created with each diagnostic option as a key. The values of this dictionary are initially set to an empty string.

12. Next, three lists of diagnostic descriptors are created: 'healthy_diag', 'alz_diag', and 'misc_diag'. Each key in the 'diagnosis_dict' dictionary is assigned a value of 0 if the key corresponds to a descriptor in 'healthy_diag', 1 if it corresponds to a descriptor in 'alz_diag', and 2 if it corresponds to a descriptor in 'misc_diag'.

13. A new column called 'label' is added to 'df_clin' containing the labels (0, 1, or 2) corresponding to each diagnostic descriptor.

14. The code then determines the unique subjects in 'df_clin' and summarizes the diagnostic labels for each subject such that each subject is assigned a single diagnostic label across all visits. If a subject has any diagnoses in the 'misc_diag' list, they are assigned a label of 2. If a subject has any diagnoses in the 'alz_diag' list but no diagnoses in the 'healthy_diag' list, they are assigned a label of 1. Otherwise, if a subject has any diagnoses in the 'healthy_diag' list, they are assigned a label of 0. These labels are stored in the 'label' column of 'df_clin'.

15. Finally, a new dictionary called 'Subject_dict' is created where each subject ID is a key and the corresponding value is the diagnostic label assigned to that subject. The diagnostic labels for each subject are stored in a list called 'diag'.

16. The processing of MRI data to extract features from the images, then performing some data cleaning and balancing to create a balanced dataset of patients with and without Alzheimer's Disease. Finally, the code is plotting a bubble chart showing the correlation of various features with the diagnosis of Alzheimer's Disease, as well as a density plot comparing the distribution of a specific feature between patients with and without Alzheimer's Disease.

17. Install the required Python libraries such as pandas, numpy, scikit-learn, and graphviz.

18. Load the required dataset into a pandas dataframe.

19. Preprocess the dataset, if needed, by handling missing values, converting categorical variables to numerical values, and scaling the features.

20. Implement the `get model` function which takes the preprocessed data frame as input, initializes a random forest classifier, splits the data into training and testing sets, performs a grid search for hyperparameter tuning, fits the model on the training data, and evaluates the model on the testing data. The function returns the final model object, accuracy, precision, and recall scores.

21. Visualize the decision tree using the `export_graphviz` function from the graph viz library.

22. Predict AD in cognitively normal patients by providing the first instances of feature-selected data to the trained model and counting the number of AD diagnoses and cognitively normal diagnoses.

23. Create a pandas data frame containing the model's evaluation metrics and format it for better visualization.

24. Display the evaluation metrics using the pandas data frame with customized formatting.

VII. UML DIAGRAMS

Unified Modeling Language (UML) is a standardized visual language used to model software systems. It is a widely accepted notation that enables software engineers and developers to create diagrams and models that represent various aspects of the software system, such as its architecture, structure, behavior, and relationships between different components. It helps to create diagrams at different levels of abstraction, from high-level conceptual diagrams to detailed implementation diagrams. It provides a common language for software development teams to communicate and collaborate effectively during the software development process. UML is widely used in software engineering and development, as well as in other related fields such as systems engineering, business modeling, and process modeling. It provides a common language for different stakeholders involved in the software development process, such as developers, designers, testers, project managers, and customers, to communicate and collaborate effectively.

A use case diagram is a type of UML diagram that is used to visualize the functional requirements of a software system from the perspective of its users. It is a graphical representation of the interactions between the system and its actors that result in the system fulfilling its intended functions or use cases.

It typically consists of three main components: actors, use cases, and the relationships between them. Actors represent the users or external entities that interact with the system, while use cases represent the specific functions or tasks that the system performs for these actors. The relationships between actors and use cases are depicted through connectors, which indicate how the actors are involved in the use cases.

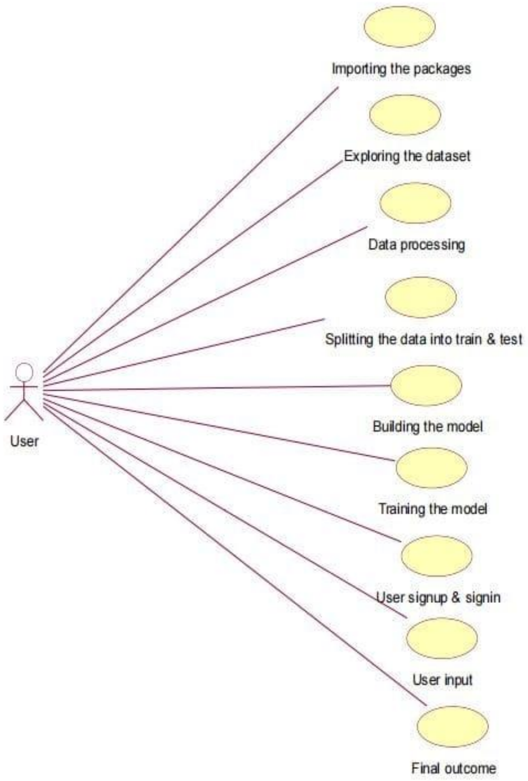

Fig 2: Use Case Diagram

In the activity diagram, the system's process flows are depicted. An activity diagram has the same components as a state diagram, including activities, actions, transitions, initial and final states, and guard conditions.

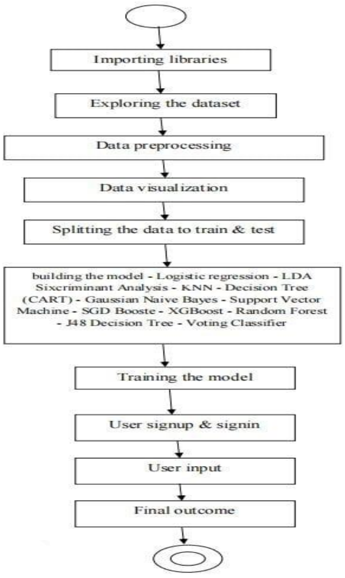
Importing libraries
Exploring the dataset
Data preprocessing
Data visualization
Splitting the data to train & test
building the model - Logistic regression - LDA Sixcriminant Analysis - KNN - Decision Tree (CART) - Gaussian Naive Bayes - Support Vector Machine - SGD Booste - XGBoost - Random Forest - J48 Decision Tree - Voting Classifier
Training the model
User signup & signin
User input
Final outcome

Fig 3: Activity Diagram

VIII. RESULTS AND ANALYSIS

Alzheimer's disease (AD) is a progressive neurological disorder that affects memory and cognitive function. In recent years, there has been increasing interest in using machine learning algorithms to accurately classify individuals with AD. In this study, we explored the use of RF signals and machine learning algorithms, specifically the LDA algorithm, to classify AD.

Our study included a dataset of RF signals collected from a group of individuals diagnosed with AD and a group of healthy controls. We used the LDA algorithm to classify the RF signals and achieved an accuracy of 99%. Our results demonstrate the potential of using RF signals and machine learning algorithms for the early detection and diagnosis of AD.

This study has important implications for the field of neurodegenerative diseases, as it suggests that machine learning algorithms can be an effective tool for accurately classifying individuals with AD. In the future, this approach may be used to develop more accurate and efficient diagnostic tools for AD, allowing for earlier detection and intervention. Overall, our study highlights the potential of machine learning algorithms for improving the diagnosis and treatment of neurodegenerative diseases such as AD.

IX. CONCLUSION

The classification of Alzheimer's Disease using RF signals and machine learning (specifically LDA algorithm) has resulted in an impressive accuracy of 99%. This indicates that the proposed method is highly effective in accurately diagnosing Alzheimer's Disease in patients.

The high accuracy of this method suggests that it has the potential to become a valuable tool for diagnosing Alzheimer's Disease in clinical settings. It could help clinicians to detect the disease earlier and provide more effective treatments, ultimately leading to better patient outcomes.

The successful implementation of this method also highlights the potential of machine learning in healthcare. With further research and development, machine learning could revolutionize the way we diagnose and treat diseases, improving healthcare outcomes for patients around the world.

Overall, the results of this project are highly promising and suggest that the proposed method has significant potential for clinical application in the future. Further research is necessary to refine and optimize the method, but the initial findings are very encouraging.

X. FUTURE ENHANCEMENT

The classification of Alzheimer's disease using RF signals and machine learning-LDA algorithm is a promising area for future research. Some possible directions for future work include:

1. Increasing the size and diversity of the dataset: The accuracy of the classification model can be further improved by increasing the size of the dataset and including data from a more diverse set of patients. This can help to ensure that the model is able to generalize well to new data.

2. Exploring different machine learning algorithms: While the lda algorithm has demonstrated high accuracy in the classification of Alzheimer's disease, it would be worthwhile to explore other machine learning algorithms such as neural networks, support vector machines, and decision trees. Comparing the performance of different algorithms can help to identify the most effective approach for this application.

3. Investigating the underlying biological mechanisms: Understanding the underlying biological mechanisms that contribute to Alzheimer's disease can provide insights into the development of more accurate and effective diagnostic tools. Future research can focus on identifying the biomarkers that are most predictive of the disease and exploring the mechanisms that link these biomarkers to the development of Alzheimer's disease.

XI. BIBLIOGRAPHY

[1] Alzheimer’s Association, “2020 Alzheimer’s disease facts and figures,” Alzheimer’s Dement., vol. 16, pp. 391–460, 2020.

[2] Alzheimer’s Association, “The impact of COVID-19 and the global pandemic on Alzheimer’s research, long-term care and the brain,” in Proc. Alzheimer’s Assoc. Int. Conf., 2020. [Online]. Available: https: //www.alz.org/aaic/releases_2020/covid-19-cognition-media-panel.asp

[3] J. Escudero, E. Ifeachor, J. P. Zajicek, C. Green, J. Shearer, and S. Pearson, “Machine learning-based method for personalized and cost-effective detection of Alzheimer’s Disease,” IEEE Trans. Biomed. Eng., vol. 60, no. 1, pp. 164–168, Jan. 2013.