DEMENTIA DETECTION USING CNN, NLP AND CARETAKER APPLICATION

A Major project *report was submitted to the* ***JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD*** *in partial fulfilment of the requirements for the award of the degree of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

A. Akshitha (18071A0502)

Ch. Sankranthi (18071A0510)

K. Sravya (18071A0527)

S. Shanmukha Anurag (18071A0551)

Under the Guidance Of

Dr. B.V. Kiranmayee

(Professor, VNR VJIET)

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AN AUTONOMOUS INSTITUTE, NAAC ACCREDITED WITH 'A++' GRADE, NBA ACCREDITED, APPROVED BY AICTE, NEW DELHI, AFFILIATED TO JNTUH)

February 2022

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute, NAAC Accredited With ‘A++’ Grade,

NBA Accredited, Approved by AICTE, New Delhi, Affiliated to JNTUH)

CERTIFICATE

This is to certify that Miss. A. Akshitha (18071A0502), Miss. Ch. Sankranthi (18071A0510), Miss. K. Sravya (18071A0527), and Mr. S. Shanmukha Anurag (18071A0551) have successfully completed their Major Project work at the CSE Department of VNR VJIET, Hyderabad entitled **“DEMENTIA DETECTION USING CNN, NLP AND CARETAKER APPLICATION”** in partial fulfilment of the requirements for the award of B.Tech degree during the academic year 2021-2022.

This work is carried out under my supervision and has not been submitted to any other University/Institute for the award of any degree/diploma.

Dr. Chalumuru Suresh	Dr.S.Nagini
Assistant Professor	Professor & HOD
Department of CSE	Department of CSE
VNR VJIET	VNR VJIET

DECLARATION

This is to certify that the project work entitled “**DEMENTIA DETECTION USING CNN, NLP AND CARETAKER APPLICATION**” submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfilment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide report of the work carried out by us under the guidance and supervision of Dr. B.V. Kiranmayee (Professor), Department of CSE, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

A. Akshitha	**Ch. Sankranthi**	**K. Sravya**	**S. Shanmukha Anurag**
18071A0502	18071A0510	18071A0527	18071A0551
IV B. Tech-CSE	IV B. Tech-CSE	IV B. Tech-CSE	IV B. Tech-CSE
VNR VJIET	VNR VJIET	VNR VJIET	VNR VJIET

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without which it would never have come into existence. To them, we lay the words of gratitude imprinted within us.

We are indebted to our venerable principal **Dr. C. D. Naidu** for this unflinching devotion, which led us to complete this project. The support, encouragement given by him, and his motivation led us to complete this project.

We express our thanks to our internal guide **Dr. B. V. Kiranmayee** and project coordinators **Dr. Ch. Suresh** and **Mrs. V. Baby** for having provided us with a lot of facilities to undertake the project work and guide us to complete the project.

We express our sincere thanks to our faculty of the Department of **Computer Science and Engineering** and the remaining members of our college **VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY** who extended their valuable support in helping us to complete the project in time.

A. Akshitha(18071A0502)

Ch. Sankranthi (18071A0510)

K. Sravya (18071A0527)

S. Shanmukha Anurag (18071A0551)

ABSTRACT

Dementia is the loss of cognitive functioning, i.e., thinking, remembering, and reasoning, to such an extent that it interferes with a person's daily life. Dementia affected people fail to control their emotions and personalities; they may vary depending on different situations. When a person is in the beginning stages of dementia, it only affects his normal functioning. By the time it advances, he must depend on caretakers even for pivotal activities of living. This syndrome is more common as people grow older but it is not a normal part of ageing. Research states that at least some form of dementia might affect around 35% of the population belonging to age 85 or more. Many people live into their 90s and beyond without any signs of dementia. In other words, dementia is a syndrome in which there is deterioration in cognitive function beyond what might be expected from the usual consequences of biological ageing. Like Alzheimer’s disease, dementia might take several forms. Depending on the type, the person’s symptoms may vary. Around 60 to 80 percent of cases are caused due to Alzheimer’s disease. Among the many other conditions that can cause dementia, some are even reversible. Presently, the syndrome stands seventh in the list of major causes of death. It is one of the main reasons behind elder people being disabled thus dependent on caretakers. Currently, more than 55 million people live with dementia worldwide, and there are nearly 10 million new cases every year. A person diagnosed with dementia may struggle to remember recent events, but they can easily recall past events. They tend to forget the names of friends or everyday objects, lose the thread of what they are saying and feel confused even when they are in a familiar environment. Memory problems are common but they can be an early sign of a medical condition such as dementia. Many people notice that their memory becomes less reliable as they get older, but other factors like the side effects of medications may be a reason. It can be caused by many conditions such as primary neurologic, neuropsychiatric, and medical conditions. Other reasons behind the development of dementia include stroke, repeated physical brain injuries, alcohol use, nutritional inadequacies, or explicit infections such as HIV. Various forms of dementia can not be easily differentiated. The syndrome affects people in distinct ways - some impacts may be physical, economic, social, psychological. These consequences influence not only the affected patients but also their families, individual careers and society wholly. Among the many causes of dementia, some can be treated while others can not even be identified. In order to treat them, all the synonyms are to be detected properly. Thus, it is important to detect. Still, obtaining an early diagnosis can help with managing the condition and planning. Currently, dementia does not have any cure or treatment. Antidementia medicines and disease-modifying therapies developed to date have limited efficacy and are primarily labelled for Alzheimer’s disease, though numerous new treatments are being investigated in various stages of clinical trials. There are many applications present which only provide information about dementia but do not help the patients in detecting or helping the person to improve their cognitive responses. The present applications mostly give basic ideas about dementia and its symptoms but don't provide any solution, so providing a proper caretaker application will help patients and make the caretakers life easy.

INDEX

CONTENTS Page No.

CHAPTER 8: IMPLEMENTATION

CHAPTER 9: TESTING

CHAPTER 1

INTRODUCTION

Dementia is the loss of cognitive functioning, i.e., thinking, remembering, and reasoning, to such an extent that it interferes with a person's daily life. Some people with dementia cannot control their emotions and their personalities may change.

Though dementia generally involves memory loss, memory loss has different causes. Having memory loss alone doesn't mean you have dementia, although it's often one of the early signs of the condition.

Dementia ranges in severity from the mildest stage when it is just beginning to affect a person's functioning, to the most severe stage when the person must depend completely on others for basic activities of living.

This syndrome is more common as people grow older but it is not a normal part of ageing. About one-third of all people age 85 or older may have some form of dementia. Many people live into their 90s and beyond without any signs of dementia.

In other words, dementia is a syndrome - usually of a chronic or progressive nature - in which there is deterioration in cognitive function beyond what might be expected from the usual consequences of biological ageing.

There are several different forms of dementia, including Alzheimer's disease. A person's symptoms can vary depending on the type. Alzheimer's disease accounts for 60 to 80 percent of cases. But, there are many other conditions that can cause symptoms of dementia, including some that are reversible.

Alzheimer's disease is the most common cause of a progressive dementia in older adults, but there are a number of other causes of dementia. Depending on the cause, some dementia symptoms might be reversible.

Dementia is currently the seventh leading cause of death among all diseases and one of the major causes of disability and dependency among older people globally. Currently, more than 55 million people live with dementia worldwide, and there are nearly 10 million new cases every year.

A person diagnosed with dementia may struggle to remember recent events, but they can easily recall past events. They tend to forget the names of friends or everyday objects, lose the thread of what they are saying and feel confused even when they are in a familiar environment.

Memory problems are common but they can be an early sign of a medical condition such as dementia. Many people notice that their memory becomes less reliable as they get older, but other factors like the side effects of medications may be a reason.

It can be caused by many conditions such as primary neurologic, neuropsychiatric, and medical conditions. Dementia may also develop after a stroke or in the context of certain infections such as HIV, harmful use of alcohol, repetitive physical injuries to the brain, or nutritional deficiencies. The boundaries between different forms of dementia are indistinct.

Dementia has physical, psychological, social, and economic impacts, not only for people living with dementia, but also for their carers, families, and society at large. Early detection of symptoms is important, as some causes can be treated. However, in many cases, the cause of dementia is unknown and cannot be treated. Still, obtaining an early diagnosis can help with managing the condition and planning.

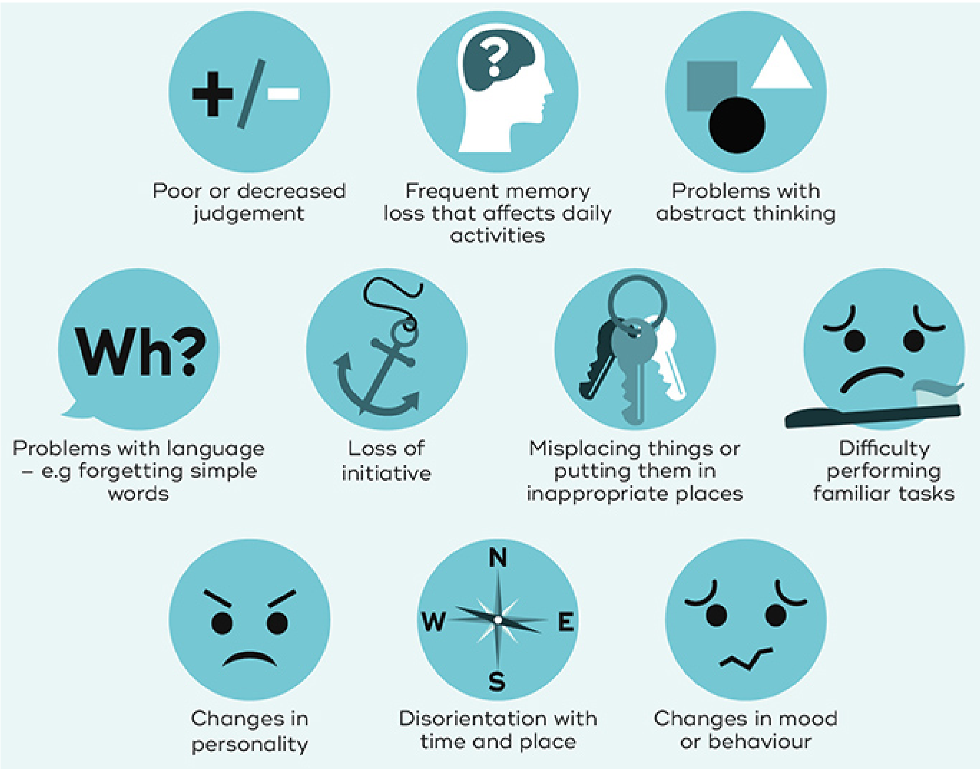


Fig 1.1 Symptoms observed in a dementia affected person

Throughout the world, around 50 million people have dementia, and there are nearly 10 million new cases every year.

There is currently no treatment available to cure dementia. Anti-dementia medicines and disease-modifying therapies developed to date have limited efficacy and are primarily labelled for Alzheimer's disease, though numerous new treatments are being investigated in various stages of clinical trials.

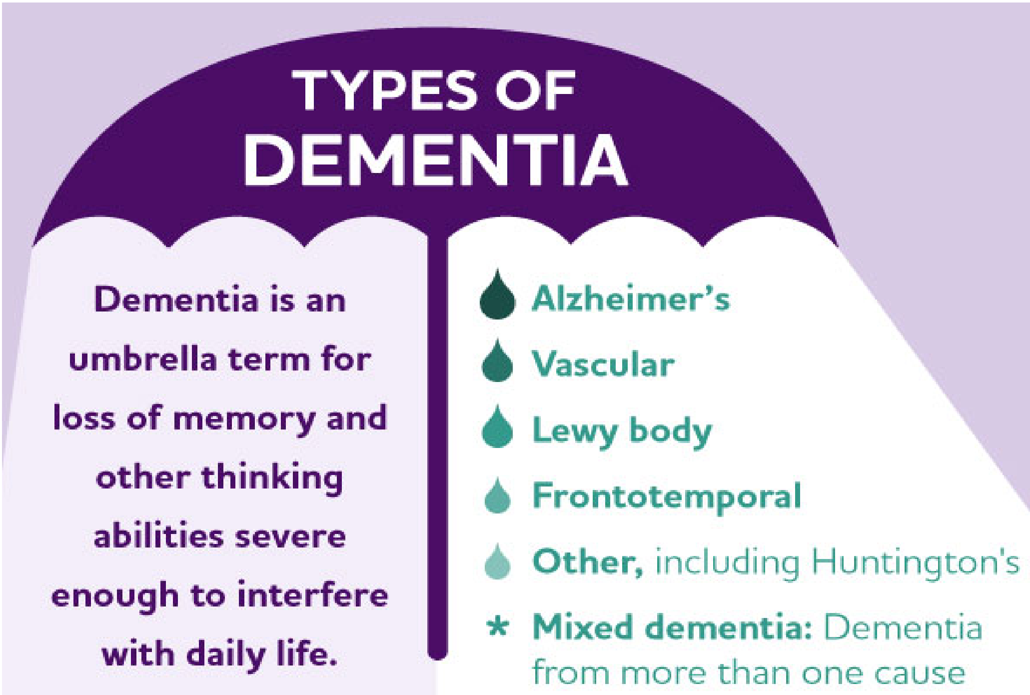


Fig 1.2 The different types of dementia based on the cause of the syndrome

1.1 Need for project

Dementia is a profoundly life-changing condition and reactions to a diagnosis can range from dismay and deep sadness to anger and despair. But for many people, it can also come as a relief. A diagnosis may well provide long-awaited answers for a failing memory, communication problems, and changes in behaviour.

Diagnosing dementia at an early stage opens the door to future care and treatment. It helps people to plan while they are still able to make important decisions on their care and support needs and financial and legal matters. It also helps them and their families to receive practical information, advice, and guidance as they face new challenges.

An early diagnosis – and access to the right services and support – can help people take control of their condition, plan for the future and live well with dementia. It will help to eliminate the possibility of other, potentially treatable, conditions with dementia-like symptoms being responsible for memory, communication, behaviour, and other problems.

It can help people with dementia to have access to relevant information, resources, and support, make the most of their abilities, and potentially benefit from drug and non-drug treatments available. An early diagnosis gives someone the chance to explain to family and friends the changes happening in their life.

It is therefore important for the person with dementia and their family to receive the dementia diagnosis positively, with time made available to answer any questions and for support and reassurance to be provided.

It is also crucial that the family including the demented person is provided access to proper care. This is more likely to lead to the individual feeling more in control and empowered to make decisions. Additionally, drug and non-drug treatment can be more effective the earlier someone is diagnosed.

This syndrome needs to be diagnosed in a possible patient as soon as possible. However, since the cost of healthcare is skyrocketing nowadays, a vast majority of the population may not be willing to get themselves checked regularly.

To avoid the adverse complications caused by the late diagnosis of dementia, a self-sustainable technology-based dementia detector will prove useful. This can be used by anyone at a very reasonable cost while obtaining the most accurate results.

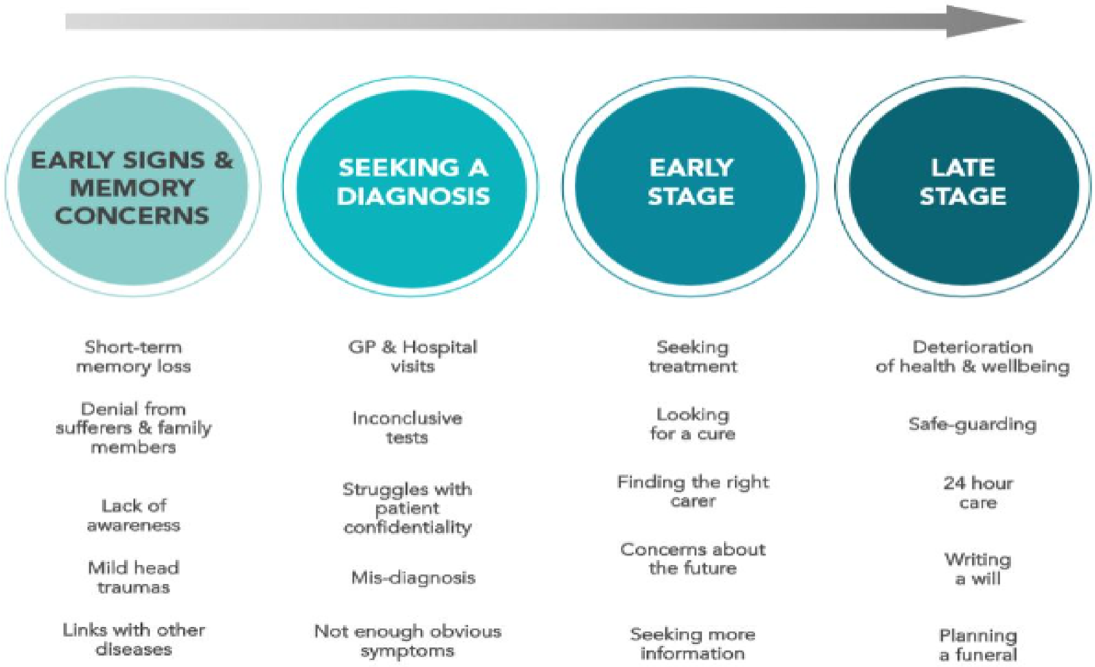


Fig 1.3 Various stages of advancement of dementia

1.2 Employing various methods

Early diagnosis of Alzheimer’s disease plays a pivotal role in patient care and clinical trials. In this study, we try to develop a new approach based on 3D deep convolutional neural networks to accurately differentiate mild Alzheimer’s disease dementia from mild cognitive impairment and cognitively normal individuals using structural MRIs.

For comparison, we built a reference model based on the statistical data of the patients and performed classification using the factors Years of Education, Socio Economic Status, Mini Mental State Exam, Clinical DementiaRating, Estimated Total Intracranial Volume, Normalise Whole Brain Volume and Atlas Scaling Factor.

The deep-learning model is more accurate and significantly faster than the classification model. The model can also be used to forecast progression, re-validate the misinterpreted information and also diagnose if the patient has dementia or not based on the MRI scans and OASIS datasets. An analysis of the features learned by the proposed model shows that it relies on a wide range of regions associated with Alzheimer's disease. These findings suggest that deep neural networks can automatically learn to identify imaging biomarkers that are predictive of Dementia And leverage them to achieve accurate early detection of the disease.

Further analysis of the Mini Mental State Exam using Natural Language Processing is done to further improve the model performance. Additionally, it removes irregularities by the use of biomarkers and analysing speech patterns of the person coming for diagnosis.

Using these wide range of methods, useful features can be included which help the patient suffering from dementia in various ways such as engaging in cognitive exercises, setting alarms and reminders so that the patient does not forget important occasions, and analysis of the application usage to provide assistance in further diagnosis under the review of a clinical doctor, etc. Eventually, they help in improving the memory power of the patient’s brain, interaction with others, carrying out daily life activities and improving the overall standard of demented patient.

CHAPTER 2

LITERATURE SURVEY

Nils D. Forkert et al. [1] developed a method based on machine learning to detect dementia from Neuropsychiatric Symptoms and Neuroimaging Data which uses Baseline neuroimaging, neuropsychiatric, and clinical data from 102 individuals with Normal Cognition and 239 with Mild Cognitive Impairment were extracted from the ADNI database. Neuropsychiatric inventory questionnaire items were transformed to MBI domains using a published algorithm. Diagnosis at the latest follow-up was used as the outcome variable and ground truth classification. The prediction of the follow-up diagnosis was done with the help of a logistic model and IG feature used for decision trees.

Problem Statement: To investigate if baseline mild behavioural impairment (MBI) status used for NPS quantification along with brain morphology features are predictive of follow-up diagnosis, median 40 months later in patients with normal cognition (NC) or MCI.

Methodology: Baseline neuroimaging, neuropsychiatric, and clinical data from 102 individuals with NC and 239 with MCI were extracted from the Alzheimer's Disease Neuroimaging Initiative database. Neuropsychiatric inventory questionnaire items were transformed to MBI domains using a published algorithm. Diagnosis at the latest follow-up was used as the outcome variable and ground truth classification. A logistic model tree classifier combined with information gain feature selection was trained to predict follow-up diagnosis.

Limitations: The study has some limitations inherent with the ADNI database and due to study criteria/analysis plan. By not having a maximum cut-off, results might be confounded as long follow-up time is beneficial in seeing changes for those with NC at baseline. It is also more likely to see MCI-converters. The study is limited by the measures included in the analysis, restricting the model to structural changes in the MRI to follow single image modality led to reduction in accuracy, it is possible to improve the accuracy by using image-based biomarker models such as PET and functional MRI instead of just using structural MRI.

Conclusion: Baseline NPS, categorised for MBI domain and duration, have prognostic utility in addition to brain morphology measures for predicting diagnosis change using ML. MBI total score along with impulse controls and regulations can be used for the prediction of future diagnosis of Alzheimer’s disease.

Future Scope: The situation of the patient or the person to be diagnosed can be changed every 6 months based on the progression or regression of dementia which can be analysed and detected only if there is a continuous track of the person to be diagnosed.

Fubao Zhu. et al. [2] developed a method based on machine learning to help the preliminary diagnosis of normal, mild cognitive impairment (MCI), very mild dementia (VMD), and dementia using an informant-based questionnaire. The model enrolled 5,272 individuals who filled out a 37-item questionnaire to select the features which could be more important than the rest. The top features combined with six classification algorithms were used to develop the diagnostic models.
Results: Information Gain was the most effective among the three feature selection methods. The diagnostic model proposed in this paper provides a powerful tool for clinicians to diagnose the early stages of dementia.
Problem Statement: The reliable diagnosis remains a challenging issue in the early stages of dementia.Using an information based questionnaire the model aimed to develop a new method based on machine learning to help the diagnosis of normal cognition, mild cognitive impairment , very mild dementia, and dementia.
Methodology: The model enrolled 5,272 individuals who filled out a 37- item questionnaire to select the features which could be more important than the rest. The top features combined with six classification algorithms were used to develop the diagnostic models
Conclusion: The diagnostic model proposed in this paper provides a powerful tool for clinicians to diagnose the early stages of dementia.
Future Scope: Further Analysis to be done on sampling techniques and classification algorithms

Shanta Rangaswamy. et al. [3] has presented the implementation of machine learning algorithms to get more accuracy to identify Parkinson disease. The data set referred to is from an online-based machine learning repository also recognized as UCI. Support vector machines, K-nearest neighbour and linear discriminant analysis algorithms are used to calculate the accuracy, recall and confusion matrix. The outcome of this implementation has given the accuracy of 100% for SVM and KNN algorithms and 80% for LDA.
Problem Statement: To investigate the process of diagnosis with more accuracy with different parameters of the disease. The paper presents the implementation of algorithms to identify Parkinson's disease and achieve higher accuracy than the existing models.
Methodology: To use machine learning algorithms such as Support Vector Machines (SVM), K-Nearest Neighbours (KNN) and Linear Discriminant Analysis (LDA).
Limitations: The solution is tested on a dataset with just 196 entries provided that it has an accuracy of 100% with SVM and KNN which might deteriorate with a bigger sample space.
Conclusion: The paper concentrated much more on feature selection rather than detection of

dementia as they got the number of useful features from the whole set to be 5 without damaging the accuracy of the model.

Harshit Parmar. et al. [4] have presented one such synergy of fMRI and deep learning, applying a simplified yet accurate method using a modified 3D convolutional neural network (CNN) to resting state fMRI data for feature extraction and classification of Alzheimer's disease (AD). The CNN model used is built to work with lesser data preprocessing so that it preserves spatial and temporal information on the functional MRI.

Problem Statement: The main purpose of the paper is to utilise the potentials of deep learning techniques for fMRI data for clinical use given that fMRI data is extremely noisy and the inaccuracy involved while thoroughly using the immense quantities of information about the dynamics of the brain.

Methodology: Applying a simplified yet accurate method using a modified 3D convolutional neural network (CNN) to resting state fMRI data for feature extraction and classification of Alzheimer's disease (AD). The CNN is designed in such a way that it uses the fMRI data with much less preprocessing, preserving both spatial and temporal information.

Limitations: Purely relying on the fMRI images would not lead to the accurate diagnosis of dementia however accurate the model can be. This is because of the inaccuracies and noise present in the MRI images.

Conclusion: The CNN can detect and differentiate between the earlier and later stages of MCI and AD and hence, it may have potential clinical applications in both early detection and better diagnosis of Alzheimer's disease.

Future Scope: The model can be integrated with other models with much more useful features which are clinically used to detect and diagnose dementia.

Gloria Castellazzi. et al. [5] provided a method which combines local DTI metrics and GT measures from rs-fMRI data with ML, showing great potential for the automatic classification of AD and VD in patients with mixed clinical assessment. Indeed, multimodal features from MRI could be used to automatically separate AD from VD patients with high accuracy and balanced sensitivity and specificity. Among the pool of ML algorithms available to the user, ANFIS appeared to overcome others in classification performance types (Diffusion Tensor Imaging and Graph theory).

Problem Statement: Tackling the differential diagnosis of Alzheimer disease and vascular dementia fed by MRI-selected features.

Methodology: Improvements in imaging, which represent a pool of qualified methods for exploring data to discover already present unknown patterns. Other studies have shown that the combination of ML with quantitative MRI represents a suitable approach not only to automatically identify dementia diseases, but also to predict the disease progression. A recent study showed that ML combined with volumetric measurements derived from structural MRI represents a useful approach for the differential diagnosis of Alzhiemer's disease and vascular dementia.

Limitations: Usage of MRI is highly noisy and riddled with inaccuracies. Purely relying on the fMRI images would not lead to the accurate diagnosis of dementia however accurate the model can be. This is because of the inaccuracies and noise present in the MRI images.

Conclusion: The encouraging results, we strongly believe that ML coupled to high-resolution MRI will provide a suitable approach to support clinicians in their clinical work, helping them to improve their diagnostic and prognostic accuracy as well as therapy and patient management.

Future Scope: Future works should explore the pattern of features identified here together with clinical and neuropsychological variables and metrics from biological samples to improve the accuracy of the algorithm even further.

Sulantha M. et al. [6] proposed a machine learning-based probabilistic method designed to assess the progression to dementia within 24 months, based on the regional information from a single amyloid positron emission tomography scan. Importantly, the proposed method was designed to overcome the inherent adverse imbalance proportions between stable and progressive mild cognitive impairment individuals within a short observation period. The novel algorithm obtained an accuracy of 84% and an under-receiver operating characteristic curve of 0.91, outperforming the existing algorithms using the same biomarker measures and previous studies using multiple biomarker modalities.

Problem Statement: Amyloid-β protein is the core pathologic feature of Alzheimer's disease, biomarkers of neuronal degeneration are the only ones believed to provide satisfactory predictions of clinical progression within short time frames. Usage of these imaging techniques to predict whether the patient has dementia or not over a time span of 24months.

Methodology: Usage of Machine learning algorithms over the single amyloid positron emission tomography scan like SVM , L1 logistics, L2 logistics Random forest classifier, Random Under Sampling Random Forest (RUSRF).

Limitations: Any Alzheimer's disease study involving in vivo data suffers from the inherent uncertainty of the diagnosis . This is because the diagnosis of Alzheimer's disease can only be confirmed with autopsy data. Therefore, any clinical diagnosis will always be "probable Alzheimer's disease." It is estimated that around 10% of this "probable Alzheimer's disease" diagnosis can be mislabeled "Lewy Body disease" or "Frontotemporal dementia".

Conclusion: A novel algorithm to predict the progression to Alzheimer's disease dementia within MCI individuals based on their Aβ PET measurements. The predictor presented here achieved a high rate of accuracy with an AUC of 0.906 for an independent test set. The novel algorithm overcomes the inherent imbalance of proportions between stable and pMCI seen in a population of MCI individuals, making it ideally suited for a clinical environment as an early diagnostic tool.
Future Scope: Can be used as a tool to calculate the progression or regression of the disease.

Nirjjon. et al. [7] have developed an application mobi-cog, that runs on a mobile device, such as a tablet or a smartphone, and provides an automated and instant dementia screening service. The MOBI-COG App is complete automation of a widely used 3-minute dementia screening test called the Mini-Cog test, which is administered by primary caregivers for quick screening of dementia in the elderly. The test involves a free-hand clock drawing test and asking the patient to remember a set of three words. The MOBI COG App automates all these steps involved in the test.
Problem Statement: Providing the person using the mobile app a mini cognitive test as a pre analysis before performing a clinical analysis irrespective of the difference in demographic information from person to person.
Methodology: The clinic of the primary caregivers provides this test to the patient asking them to recollect 3 words and draw a free hand diagram of a clock with all the digits and provide analysis of the cognitive responses of the patient using machine learning.
Limitations: The application cannot prove to be a standard test for a person with a different demographic information and can only be used as a pre-diagnosis function of a diagnostic test which cannot provide viable results.
Conclusions: This paper presents the design, implementation and evaluation of a mobile application called the MOBI-COG App. The application is an automated mini cog test. This application cannot be used as an alternative to seeing a caregiver.
Future Scope: Model can be used as a part of an application rather than a whole application as in itself. It is recommended to be a part of a clinical diagnosis to provide a fair analysis irrespective of following different doctors and overcoming the restrictions of pen and paper.

Kong. et al [8] have provided a list of iTunes apps that was compiled for the usage of early-stage or mild dementia participants. The method in choosing these apps and determining salient features of the most successful apps was reported. The results will advance the knowledge base on the innovative use of smart technology in clinical settings.

Welsh. et al. [9] designed a mobile application - Ticket to Talk - to support intergenerational interactions by encouraging young people to collect media relevant to individuals with dementia to use in conversations with people with dementia. Evaluation of Ticket to Talk through trials has taken place with groups of older people and two families. We highlight difficulties in using technologies such as this as a conversational tool, the value of digital media in supporting intergenerational interactions, and the potential to positively shape people with dementia's agency in social settings.

Shibata. et al. [10] developed a smartphone-based dementia screening application, VocabChecker, which measures language abilities from a speech narrative via automatic speech recognition (ASR). The model works extensively on 4 language abilities related to dementia i.e. number of tokens, types of tokens, type to token ratio and potential vocab size.It also reported that the use of VocabChecker has distinguished dementia patients from elderly people.

Atee. et al. [11] have described a novel method and system of pain assessment using a combination of technologies: automated facial recognition and analysis (AFRA), smart computing, affective computing, and cloud computing (Internet of Things) for people with advanced dementia.

Problem Statement: To describe a novel method and system of pain assessment using a combination of technologies: automated facial recognition and analysis (AFRA), smart computing, affective computing, and cloud computing (Internet of Things) for people with advanced dementia.

Methodology: Cognification and affective computing were used to conceptualise the system. A computerised clinical system was developed to address the challenging problem of identifying pain in non-verbal patients with dementia. The system is composed of a smart device enabled app (App) linked to a web admin portal The App "PainChek™" uses AFRA(Automated Facial Recognition and Analysis) to identify facial action units indicative of pain presence, and user-fed clinical information to calculate a pain intensity score.

Limitations: The ability of the system to collect a large amount of pain data over a period of time helps in identifying temporal patterns.

Conclusion: PainCheck is a comprehensive and evidence-based pain management system. This novel approach has the potential to transform pain assessment in people who are unable to verbalise because it can be used by clinicians and careers in everyday clinical practice.

Future Scope: The model can be made much cheaper and relate to the application with cloud access to store the huge amount of data and regress the results from it.

Megges et al. [12] presented the importance of including persons with dementia and their primary caregivers in the research and development phase of locating systems to improve user experience in home dementia care.

Problem Statement: The user experience of persons with dementia and their primary caregivers with locating systems is not firmly established.

Methodology: Eighteen dyads used a prototype locating system during 4 weeks. Primary outcomes evaluate ratings of usability, and product functions and features. Secondary outcome measures were caregiver burden, perceived self-efficacy, frequency of use, and willingness to purchase the prototype. Changes in scores between baseline (T1) and end of testing period (T2) were compared by performing independent and dependent samples correlations and descriptive statistics.

Limitations: It has been described that methodological limitations such as relying on proxy evaluations of user experience by having caregivers or professionals answer in place of PwD help contribute to the observed underutilization of locating systems. Furthermore, the importance of adequate knowledge on using locating systems was emphasised. Others examine using GPS technology focused on research questions regarding mobility and cognitive impairment.

Conclusion: The most important conclusion is the high relevance of carrying out randomised controlled trials with larger, more representative samples in a real environment. This recommendation was also stated by other authors. Proposed suitable outcome measures in earlier studies were “time spent searching” and “days until long-term admission” as well as caregivers' well-being and quality of life.

Future Scope: Because technological limitations often inhibit investigating the usefulness of assistive technology, we recommend to focus on the usability, until these kinds of limitations have been overcome. In addition, future studies that include different stakeholders will likely help us gain more insights into how to better address the needs and preferences of PwD and caregivers to improve their user experience with locating systems. As this kind of research addresses many disciplines such as business, design, gerontology, neurology, and psychiatry, focusing on interdisciplinary research is required.

Zorluoglu et al. [13] developed a mobile app to improve pain assessment and management in this vulnerable population. They conducted usability testing of a newly developed iPhone pain assessment application with potential users, in this case as a tool for clinical paramedic practice to improve pain assessment of older adults with cognitive impairment.

Sindi. et al. [14] developed CAIDE Risk Score (mobile application) App based on the CAIDE Dementia Risk Score, involving information on age, educational level, hypertension, hypercholesterolemia, obesity, and physical inactivity. The CAIDE Risk Score App is the first to predict the risk for dementia through a predominant evidence-based tool. The App can encourage users to actively decrease their adjustable risk factors and postpone cognitive impairment and dementia.

Problem Statement: The CAIDE (Cardiovascular Risk Factors, Ageing, and Incidence of

Dementia) Dementia Risk Score is a validated tool to predict late-life dementia risk (20 years later), based on midlife vascular risk factors. The goal was to render this prediction tool widely approachable.

Methodology: The CAIDE Risk Score (mobile application) App was developed based on the CAIDE Dementia Risk Score, involving information on age, educational level, hypertension, hypercholesterolemia, obesity, and physical inactivity.

Limitations: The model probes 20 years into the future by the readings of middle aged health factors which might prove to be a good approach on paper but a bad approach on real life.

Conclusion: The CAIDE Risk Score App is the first to speculate the risk for dementia through a predominant evidence-based tool. The App can encourage users to actively subside their modifiable risk factors and postpone cognitive impairment and dementia.

Future Scope: The use of the App will encourage users to actively decrease their risk factors and has the potential to postpone the onset of dementia, as well as other chronic conditions.

Thorpe. et al. [15] developed a prototype by combining a smartphone, smartwatch, and various applications to offer six support features. This is approved among five end-users (PwD) and their caregivers. supervised usability testing was followed by field testing in a real-world context. Data is accumulated from video recordings, interaction logs, system usability scale questionnaires, logbooks, application usage logs and interviews structured on the unified theory of acceptance and use of technology. The data is explored to evaluate usability, usefulness and user acceptance.

Problem Statement: Smart mobile and wearable technology offer ubiquity and popularity could even benefit user adoption – a great challenge for assistive technology (AT) for PwD that calls for user-centred design (UCD) methods.

Methodology: Research on co-design methods for users with cognitive impairment encourages the involvement of end-users and the use of simple prototypes. Several studies developing AT for dementia document a UCD approach that involves users and includes evaluation of usability and adequacy. Though crucial, these examples apply UCD methods to develop specialised AT equipment rather than employ off-the-shelf solutions..

Limitations: The user testing methods were time-consuming in terms of data collection and analysis, limiting the number of participants. PwD varies substantially regarding symptoms, lifestyle, background and technology literacy etc. – all of which may affect adoption – therefore, it is not possible to make broad generalisations about user adoption of the pervasive AT solution. Instead, our results provide the first impression, showing some promise for user adoption and pinpointing key issues for further study.

Conclusion: This work represents the development and testing of a pervasive AT solution created

by combining off-the-shelf smart technology. These devices offer advantages over specialised AT regarding user adoption, motivating an evaluation of their usability, adequacy and user acceptance.
Future Scope: Finally, implementation in clinical practice is emphasised as an important area of future research, since this is a significant challenge for wearable technology in healthcare. Approaches for healthcare providers to introduce pervasive AT to PwD and incorporate them into care practices are needed to accelerate and support adoption.

Vahia. et al. [16] investigate the feasibility, safety, and utility of tablet devices as novel nonpharmacologic tools in managing older psychiatric inpatients with agitation and dementia. Using tablets as interference for anxiety in severe dementia affected older people and others seems to be safe and feasible. This use is seen as a nonpharmacologic interruption.

Boyd. et al. [17] have focused on the development and usability evaluation of EnCare diagnostics (ECD) and the brain fit plan (BFP) in healthy older adults, cognitively impaired and physically impaired individuals. Calculated on colour selection and customised cognitive stimulation, they have proposed ECD and BFP solutions respectively. In order to assess the utility of the applications, this research was based on two trials. This usability study demonstrated that ECD is highly acceptable in both healthy older adults and those with early stage dementia when given shorter versions to accommodate their diagnosis.

Dethlefs. et al. [18] provided a spoken natural language interface that allows people with dementia to interact with the cognitive stimulation software in the same way as they would interact with a human caregiver. Thus, irrespective of their technical prowess, users can access this beneficial system while enjoying a completely instinctive user interface. A model research which evaluates the practicality of a cognitive stimulation with the help of a plain spoken language confluence was also described in this article. Prototype software was evaluated with 23 users, including healthyelderly people and people with dementia. Feedback was overwhelmingly positive.

Daniel S. et al. [19] have conducted an analysis on ADNI data in order to determine its effectiveness in the process of building three categories of classification models - Cognitively Normal (CN), Mild Cognitive Impairment (MCI), and Dementia (DEM) based on the tuning of three deep learning models. The ReliefF algorithm was used for selecting the most informative features as it proved to be better than the Information Gain technique.
Problem Statement: The purpose of this paper is to conduct an analysis on ADNI data and determine its effectiveness for building classification models to differentiate the categories Cognitively Normal (CN), Mild Cognitive Impairment (MCI), and Dementia (DEM), based on tuning three Deep Learning models: two Multi-Layer Perceptron (MLP1 and MLP2).

Methodology: Routine primary care patient records in the UK have been and are currently used to develop a risk score for the purposes of estimating how at risk an individual may be of developing dementia, by using conventional statistical methods and modern machine learning algorithms. Positron emission tomography (PET) scans and the regional analysis of the protein amyloid-β, have been used by a Random Forest classifier to identify patients with age-related stable MCI and pMCI. In a recent EMIF-AD study, a machine learning methodology based on Extreme Gradient Boosting XGBoost, Random Forest and Deep Learning.
Limitations: Deep Learning requires more sophisticated data pre-processing and feature engineering, as well as much more computational time for training a single model in comparison to statistical implementations.
Conclusion: All the models were able to recognize patterns differentiating the three classes DEM (Dementia), MCI (Minor Cognitive Impairment) and CN (Cognitive Normal), which indicates that each of these machine learning approaches has the capacity to accurately predict AD VD and MCI.
Future Scope: The ConvBLSTM model was slightly less accurate but was explored in view of comparisons with the MLP models, and for future extensions of this work that will take advantage of time-related information.

M. Tanveer et al. [20] have researched on 165 papers from 2005 to 2019 and provide the review using feature extraction and machine learning techniques in order to gain a better understanding of the work done on Alzheimer's. They presented a detailed review on three approaches for Alzheimer's with possible future directions - Support Vector Machines (SVM), Artificial Neural Network (ANN) and Deep Learning (DL).
Problem Statement: Efficient automated techniques are needed for early diagnosis of Alzheimer's. For the classification of Alzheimer's disease, researchers have proposed several state-of-the-art approaches. However, to develop more efficient learning techniques, better understanding of the work done on Alzheimer's is needed. They have provided a detailed review after going through many papers which include machine learning techniques. This experience has helped them in gaining more knowledge regarding Alzheimer's disease.
Methodology: The machine learning techniques are surveyed under three main categories: support vector machine (SVM), artificial neural network (ANN), and deep learning (DL) and ensemble methods. We present a detailed review on these three approaches for Alzheimers with possible future directions.
Conclusion: In this work, papers using three major machine learning techniques - SVM, ANN and DL are analysed for diagnosis of Alzheimer's. Research on other learning techniques like transfer, ensemble, and multi-kernel learning is also discussed.

G. Battineni et al. [21] explored the usage of Support Vector Machine (SVM) in the prediction of dementia and validated its performance through statistical analysis. They achieved optimised results with efficient performance values.

Problem Statement: Using Support Vector Machine (SVM) in the prediction of dementia and validated its performance through statistical analysis.

Methodologies: Support Vector Machine (SVM) in the prediction of dementia and validated its performance through statistical analysis. They achieved optimised results with efficient performance values.

Limitations: Works on multi-class classification but fails with terms of accuracy. Works better only if there is a chance of having binary classification.

Conclusion: Using proper kernel function proved effective in the detection of dementia and avoiding the problem of multi-dimensional data.

Future Scope: The model needs to be further improved in the context of detecting the stage of the dementia with the help of stronger kernel functions and a much larger data set than which is used in the system.

Jo T et al. [22] have identified, reviewed, evaluated and classified deep learning papers on Alzheimer's Disease by algorithm and neuroimaging type and the findings were summarised. The best classification performance was said to be obtained when multimodal neuroimaging and fluid biomarkers were combined. In order for diagnostic classification and prognostic prediction using deep learning to reach readiness for real world clinical applicability, several issues need to be addressed such as transparency and reproducibility.

Problem Statement: The application of deep learning to early detection and automated classification of Alzheimer's disease (AD) has recently gained considerable attention, as rapid progress in neuroimaging techniques has generated large-scale multimodal neuroimaging data. In this project, in order to classify AD in a diagnostic manner, an ordered review of publications which used neuroimaging data and deep learning approaches was performed.

Methodologies: Deep learning approaches, such as convolutional neural network (CNN) or recurrent neural network (RNN), that use neuroimaging data without pre-processing for feature selection have yielded accuracies of up to 96.0% for AD classification.

Limitations: Efficient error functions and gradient computing methods were discussed in these seminal publications, spurred by the demonstrated limitation of the single layer perceptron, which can learn only linearly separable patterns.

Conclusion: Due to the complexity of the deep learning algorithm, which has multiple hidden layers, it is also difficult to determine how those selected features lead to a conclusion. It is also uncertain how the specific features or their subclasses lead to their relative importance.

Future Scope: Future studies ultimately need to replicate key findings from deep learning on entirely independent data sets. This system has been lagging to penetrate neuroimaging data and its associated deep learning technology. However, it is extensively used in genetics and related fields.

Vijay S. et al. [23] worked on extending previous research by basing model estimation on a very large integrated dataset of medical claims and electronic health record (EHR) data. They hypothesised that the use of larger, more complex data, label learning, and sophisticated machine learning estimation methods lead to substantial improvements in predictive algorithms for dementia, that is, raising sensitivity to 25% or more over the 16% result of existing research. They adopted a label learning methodology designed to address the issues identified in previous studies.

Problem Statement: The study objective was to build a machine learning model to predict incident mild cognitive impairment, Alzheimer's Disease, and related dementias from structured data using administrative and electronic health record sources.

Methodologies: A cohort of patients (n = 121,907) and controls (n = 5,307,045) was created for modelling using data within 2 years of the patient's incident diagnosis date. Additional cohorts 3– 8 years removed from index data are used for prediction. Training cohorts were matched on age, gender, index year, and utilisation, and fitted with a gradient boosting machine, lightGBM.

Conclusion: Incident 2-year model quality on a held-out test set had a sensitivity of 47% and area-under-the-curve of 87%. In the 3-year model, the learned labels achieved 24% (71%), which dropped to 15% (72%) in year 8.

Future Scope: Additional progress will require better methods, more patient and family data and better clinical documentation to determine which patients are miss-identified in the source data.

H. Niu et al. [24] have evaluated the prevalence and incidence of Alzheimer’s Disease (AD) in Europe. Novelty is this study’s strength as no meta-analyses were conducted on this topic until then. They observed a slight increase in the prevalence of AD in Europe in the past few years and a decrease in incidence rates. The authors reckon similar trend studies of AD are necessary in other parts of the world.

Problem Statement: The aim of this meta-analysis is to evaluate the prevalence and incidence of AD in Europe.

Methodologies: Literature search was conducted on Medline, Scopus, and CINAHL Complete using the keywords «Alzheimer», «Alzheimer's disease», and «AD» combined with «prevalence», «incidence», and «epidemiology». A Bayesian random effects model with 95% credible intervals was used. The I2 statistic was applied to assess heterogeneity.

Limitations: One of the main limitations of the study is that they did not have access to more databases and were therefore unable to assess more articles.
Conclusion: The results of our meta-analysis allow a better grasp of the impact of this disease in Europe.
Future Scope: Future work would be to expand the scope of the work done and obtain data which is not confined as it would comparatively give biassed results.

So A. et al. [25] proposed a two-layer model inspired by the method used in dementia support centres for the early diagnosis of dementia and using machine learning techniques. Several supervised learning algorithms were used and the performance of each algorithm is compared with every other one. This proposed model claims to reduce the time and economic burden and help simplify the diagnosis method for dementia. This reduction helps to diagnose dementia in early stages in a fast, inexpensive, and reliable way, which improves the current clinical practice.

Molinuevo J. L. et al. [26] have worked on the ALFA project whose aims include identifying early pathophysiological events and developing prevention programs for Alzheimer's Disease. The ALFA project represents a valuable infrastructure that will leverage different studies and trials to prevent AD. The longitudinal ALFA+ cohort will serve to untangle the natural history of the disease and to model the preclinical stages to develop successful trials
Problem Statement: The preclinical phase of Alzheimer's disease (AD) is optimal for identifying early pathophysiological events and developing prevention programs, which are shared aims of the ALFA project, including the ALFA registry and parent cohort and the nested ALFA+ cohort study.
Methodologies: The ALFA parent cohort baseline visit included full cognitive evaluation, lifestyle habits questionnaires, DNA extraction, and MRI. The nested ALFA+ study adds wet and imaging biomarkers for deeper phenotyping.
Conclusion: A total of 2743 participants aged 45 to 74 years were included in the ALFA parent cohort. We show that this cohort, mostly composed of cognitively normal offspring of AD patients, is enriched for AD genetic risk factors.
Future Scope: Throughout the ALFA project, biological and neuroimaging markers present in the AD preclinical phase will be detected, preceding or informing about the presence of brain Aβ deposition. Furthermore, biomarkers used and validated in these studies may constitute endpoints for the construction of large population interventional randomised controlled trials, which will lead to the development of pharmacological and nonpharmaceutical interventions targeting individuals at risk for AD.

Koo B. et al. [27] presented a review of the literature on how mobile platforms, especially smartphones and tablets, are being used for cognitive assessment of older adults along with benefits and opportunities associated with the same. They summarised research findings and categorised them by level of innovation to identify gaps in research and to make suggestions for future study.

Problem Statement: This article presents a review of the literature on how mobile platforms—smartphones and tablets—are being used for cognitive assessment of older adults along with benefits and opportunities associated with using mobile platforms for cognitive assessment.

Methodologies: The procedure started with a systematic search of published literature. We then analysed selected articles for key elements of mobile cognitive assessment and categorised them by level of innovation.

Conclusion: Twenty-nine articles met our inclusion criteria and were categorised into 3 groups as follows:(a) mobile versions of existing articles or computerised neuropsychological tests; (b) new cognitive tests developed specifically for mobile platforms; and (c) the use of new types of data for cognitive assessment. This scoping review confirms the considerable potential of mobile assessment.

Future Scope: Future work should include more diverse participant samples, investigate new sensor and wearable technologies, improve reliability and validity, and incorporate the points of view of people being assessed, carers, and clinicians.

CHAPTER 3
EXISTING SYSTEM

3.1 Detection

In the existing system, it is difficult to identify if a person is suffering from dementia. It can be done with the help of clinical history and by knowing if the person has some genetic disorder. Sometimes, it is also possible that the doctor may not be able to detect the disease. Dementia is characterised by progressive problems with thinking and behaviour that start in middle or old age.

Doctors diagnose dementia based on a careful medical history, a physical examination, laboratory tests, and the characteristic changes in thinking, day-to-day functioning, and behaviour associated with different types of people and symptoms. They perform cognitive and neurological tests, brain scans, psychiatric evaluation, genetic tests, and blood tests among others.

Many machine learning algorithms were developed in the areas of cognitive, neurological tests, and evaluation tests by the use of NLP and classification methods. In addition, brain scans are analysed with the help of neural networks and image analysis. Up until now, the amalgamation of these algorithms like CNN, NLP along with medical data has played a crucial role in providing solutions for the detection of a myriad of dementia syndromes.

CNN analyses visual imagery like brain scans and performs operations to detect dementia. Although it is a reliable technique, CNN would need data in abundance to come to a conclusion that would make the system slower. So, using this algorithm alone won't suffice and provide results as required.

NLP helps in observing the biomarkers in the speech patterns made by the patients for the mini-mental state examination. This method satisfies the ground rules for detecting dementia. It casts away the possibility of patients having a different native language, accent, and educational factors for them to speak the way they do.

SVM methods were used along with kernel functions, and a few other methods like K Nearest Neighbours are used to identify a pattern among people suffering from dementia. SVM came close to separating the data points using a hyperplane, thus solving the problem of multiclass classification but there was a decline in accuracy while predicting the outcome.

Later, KNN was used to overcome the "quantifiability of the stages in dementia" with radial functions, but the following method had issues with the size of the dataset being too small or too large leading to inaccuracy or vastly increasing the time consumed.

3.2. Providing Assistance

Over the years, multiple applications have been released to help demented individuals and their caretakers in assisting themselves or the patients, a few of them are mentioned below.
One such application provides analysis of narrative speech and finds four language ability scores. Another application ensures that the required assistance is delivered to the patient at home through the caregiver where he/she measures the burden perceived self-efficacy, frequency of use, willingness to purchase the prototype, etc. but the drawback of this application would be that the model is not precise and accurate which causes a problem to the patients.

Dementia risk is measured through cardiovascular risk factors, ageing and incidence of dementia where factors like age, educational level, hypertension, physical inactivity are considered. These factors should be improved by providing more information to get precise and reliable predictions.

The user-centred design investigates the adoption of pervasive assistive technology among people with mild dementia but this solution is time-consuming in terms of data collection, analysis and limits the number of participants.

Community-based trials, developing and evaluation of the usability of EnCare diagnostics and Brain Fit Plan application to detect and manage cognitive decline. This should abruptly drop in the users after the first iteration due to a challenging interface that was not accompanied by caretaker support.

All these platforms have incurred challenges where it questions the usability or the methods used. This just concludes to not having a reliable, assistant-providing interface that could tackle all the drawbacks or factors that are missing.

CHAPTER 4
PROPOSED SYSTEM

4.1 Objectives

- Detection of Dementia with the help of Machine Learning algorithms
- Performing data analysis over the obtained results
- Extending proper care and support to the demented patient
- Providing resources to the caretaker of the patient
- Helping in improving the cognitive thinking of the dementia affected patient
- Focusing on accurately detecting dementia and delivering treatment rather than finding a cure.

4.2 Methodologies

The proposed system tries to quantify the aspects of the brain functions with the help of biomarkers, and quantify aspects like understanding, recognising, thinking and behavioural skills of the patients where the individual with dementia faces problems every day.

Few people with dementia are unable to deal with their emotions and their personalities can be changed. From the mildest stage dementia and its effects vary in severity. No cure is available other than treatment.

The main aim of the project is to recognise dementia among various patients at an early stage and also provide proper assistance in terms of technology to help the patient.

A machine learning algorithm can reduce this problem by predicting the disease. An application system can help the patient in improving their thinking ability, reading ability and improving their cognitive responses, reading and thinking abilities and many more which are degraded over the progression of the disease.

The project focuses on building a tool which first helps in predicting whether the given patient has dementia or not and then providing assistance based on the diagnosis provided by the machine learning model.

To detect dementia the machine learning model is built on three different classification systems dependent on different factors of the person up for diagnosis. The first method is a random forest classifier which works on the features like:

- Years of education
- Socio Economic status
- Mini Mental State Examination
- Clinical Dementia Rating
- Estimated Total Intracranial Volume
- Whole Brain Volume
- Atlas Scaling Factor

The Data Classification is an ensemble classification of Random Forest Classification over dataset with features such as age, MMSE score, years of education, socio economic status, clinical dementia rating, estimated total intracranial volume, normalise whole brain volume, atlas scaling factor and 3D CNN classification method over MRI images and usage of NLP LSTM method to identify biomarkers over the answers given for the mini mental state exam tests.

Random Forest Classifier
Dataset : OASIS

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each tree in the random forest spits out a class prediction and the class with the most votes becomes the model's prediction. It utilises ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. It uses bagging and features randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

The reason that the random forest model works so well is that a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The random forest is a classification algorithm consisting of many decision trees. It uses bagging and features randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

In other words, a random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
Random forest algorithms can be used for both classifications and regression tasks. It provides higher accuracy through cross-validation. A random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

3D CNN
Dataset: ADNI

The model proposed by the system uses 3D Convolutional Neural Networks which is nothing but the 3D CNN classifier. The pre-processed data is taken by the classifier. Then it is divided in a way where seventy percent of the data goes as a training dataset used for the model's training and thirty percent of the other data is used as the testing dataset for testing the trained model.

The model uses a cumulative of six neural layers. Among these layers, we use the first layer as the input layer and the next set of four neural layers constitute the hidden layer and ultimately the last layer is taken as the output layer. The model included several modules used for translation of complex images and preserves the spatial and temporal properties of the data. A 90% accuracy can be secured with this proposed system.

Some of the pros provided by the currently proposed system is as follows 1) Doesn't compromise on the temporal and spatial aspects of the data. 2) Mining the data for hidden biomarkers that are crucial for generating more efficient diagnosis results. 3) Deal with imbalanced and varied sized datasets. 4) Faster computation capability with higher accuracy and better performance.

A robust diagnostic of a particular disease should adapt to various datasets, such as, e.g., MRI scans collected by several patient groups, as to diminish discrepancies in data distributions and biases toward specific groups. Deep learning aims to decrease the use of domain expert knowledge in designing and extracting most appropriate discriminative features.

Several popular non-invasive neuroimaging tools, such as structural MRI (sMRI), functional MRI (fMRI), and positron emission tomography (PET) have been investigated for developing such a system. The latter extracts features from the available images, and a classifier is trained to distinguish between different groups of subjects, e.g., AD, mild cognitive impairment (MCI), and normal control (NC) groups.

The sMRI has been recognized as a promising indicator of AD progression. Compared to the known diagnostic systems, the proposed system employs a deep 3D convolutional neural network (3D-CNN) pretrained by 3D Convolutional Autoencoder (3D-CAE) to learn generic discriminative AD features in the lower layers. This captures characteristic AD biomarkers and can be easily adapted to datasets collected in different domains. To increase the specificity of features in upper layers of 3D-CNN, the discriminative loss function is enforced on upper layers (deep supervision).
The proposed deep 3D CNN for learning generic and transferable features across different domains is able to detect and extract the characteristic AD biomarkers in one (source) domain and perform task-specific classification in another (target) domain. The proposed network combines a generic feature-extracting stacked 3D-CAE, pre-trained in the source domain, as lower layers with the upper task-specific fully connected layers, which are fine-tuned in the target domain.

NLP

Dataset: Dementia Bank

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written - referred to as natural language. It is a component of artificial intelligence (AI). It is a discipline that focuses on the interaction between data science and human language and is scaling to lots of industries.

Language variation can act as a proxy that monitors how a patient's cognitive functions have been affected (e.g., issues with word finding and impaired reasoning). The machine learning models can exploit these language patterns and thus help in diagnosing the patient.

Code for LSTM (Long Short Term Memory, which is a recurrent neural network technique) is done by comparing it with Particle Swarm Optimisation Methodology. Long Short Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It is capable of handling the vanishing gradient problem faced by RNN. A recurrent neural network, also known as RNN, is used for persistent memory.

CNN: For each sentence, we apply an embedding and a convolutional layer, followed by a max pooling layer. The convolution features are obtained by applying filters of varying window sizes to each window of words.The result is then passed to a softmax layer that outputs probabilities over two classes.

LSTM-RNN: CNNs are not specialised for capturing long-range sequential correlations. We thus also experimented with an LSTM-RNN model, which consists of an embedding layer followed by an LSTM layer. The final state, containing information from the entire sentence, is fed to a fully-connected layer followed by a softmax layer to obtain the output probabilities.

CNN-LSTM: Observing that both models achieve results comparable to previous best performing approach, and considering that they each have their own complementary strengths, we experimented with a combined architecture, laying an LSTM layer on top of CNN. This CNN layer is identical to the vanilla CNN before the max-pooling layer, and the LSTM layer is identical to the vanilla LSTM-RNN after the embedding layer.

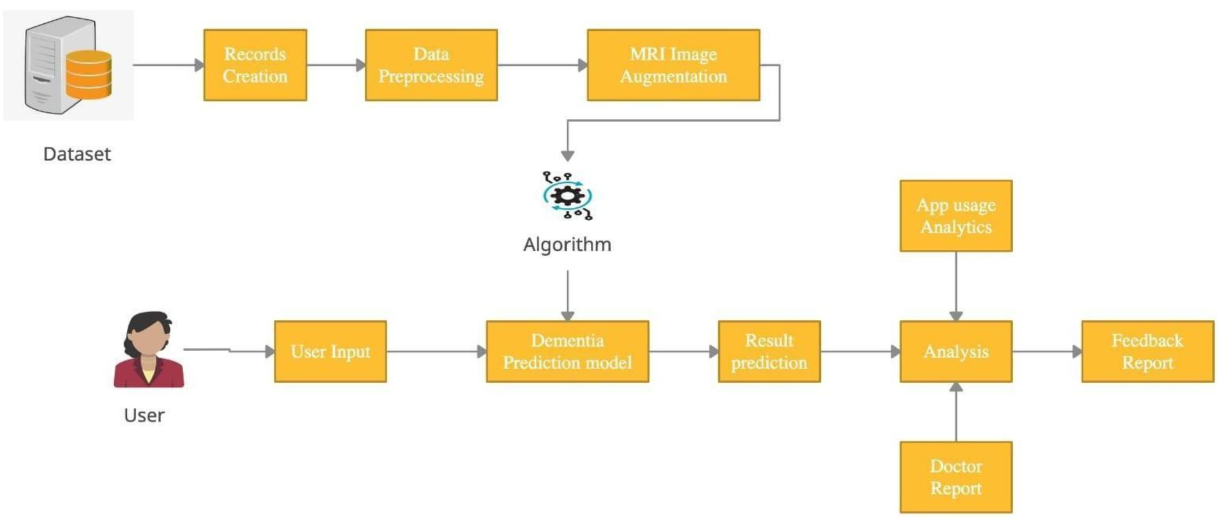


Fig 4.1 Work Flow Diagram

The data obtained from the database is made ready for the analysis which undergoes preprocessing, feature selection and choosing of machine learning algorithms to predict the output. User specific result is predicted based on the user's input which is further analysed along with the models output, doctors report and the report provided by the application that is the analytics obtained while using the interface. Feedback is given to the user combining all these factors.

CHAPTER 5
FEASIBILITY STUDY

Dementia detection and diagnosis is not an easy task. It requires considering many factors. Using various classification methods for this purpose can be a good solution. This would allow a precise detection of dementia in real-time. The accuracy of such a system is very important because the wrongful diagnosis of a symptom not present in the patient could be fatal.

An important issue in the design of the dementia detection system is the right choice of the analysed symptoms of dementia. In a situation where it is not possible to monitor all the potential symptoms of dementia, it should be limited to the symptoms which are most common and easily classifiable to produce accurate results. A few of these are Years of Education, Socio-Economic Skills, Mini-Mental State Examination, Clinical Dementia Rating, Estimated Total IntraCranial Volume, Normalise Whole Brain Volume, Atlas Scaling Factor.

The basis of the dementia detection system is the algorithms responsible for classifying the patient's data based on the dataset provided. They are classified using various classifiers with the help of machine learning concepts like decision trees and random forests in both mathematical and statistical manners. Some methods are very effective in detecting characteristic dementia features but sensitive to outlier data.

The method based on various classification methods is used for processing the input data, classification methods are used for identification and classification of pattern data, and therefore they are also used in classifying the patient data and classes. While decision trees are the most commonly used methods for classification problems, ensemble methods were also used to improve the performance of the system.

Common methods used to detect dementia are analysing the brain MRI images of the patient, and the further examination of the Mini-Mental State Examination results using NLP or tokenization. Differential methods determine the difference between the subsequent image frames.

Additionally, to improve dementia detection, the outlier data should be removed. Also, the removal of unwanted features should take place to avoid overfitting. When designing a detection and diagnosis system one should choose algorithms that are resistant to interference. Interference occurrence may disturb the processing and the analysis of the data, which may lead to misinterpretation by the system.

The next procedure is to register and analyse the movement of the classified features. For example, if the person without dementia is diagnosed as demented then there is a risk of using unnecessary medication and in the opposite scenario where the patient with dementia is diagnosed as non-demented then leaving the patient without proper care might worsen shortly and that Behavioral too.

Feasibility study includes making a decision if a project is executable. Generally, such studies precede technical development and project implementation. This study assesses the potential of a project to achieve success, hence one of the important factors is perceived objectivity in authority of study for the lending institutions and potential investors. With a rapid advancement of science and technology, the development and enhancement of hardware and software have substantially lowered developers' and users' costs. Educational information, learning environments, and communication techniques have also changed dramatically. Based on machine learning features, this application serves better for better understanding of symptoms, causes & detection of Dementia which is difficult to do in the regular study methods.

Organisational Feasibility

Organisational feasibility emphasises on how well the objective of organisation is supported by the proposed information system and its respective plan for the proposed system. The web application will serve its best for the organisational feasibility goals. The structured results displayed in the application help the clinicians and researchers of hospitals to better understand the diagnosis of ADNI. The access to the application is quite quick and indeed is useful for the hospitals to utilise the application instantly for the diagnosis based on the results and the severity levels. Based on the output and suggestions generated in the result, the necessary treatment or therapies can be started accordingly.

Economic Feasibility

Economic Feasibility helps in assessing the cost, viability, and the benefits which are in association with the projects in advance to the financial resources being allocated. This valuation basically involves the benefit or cost project analysis. 11 The application is so designed that it requires minimal cost and eliminates costs as there is a minimal need for manual work. The technologies used, help in understanding the user without any investment. As the machine will be trained it reduces the cost that is required for deploying the man power and also eradicates the problem of time consumption. The project is financially feasible because it can be accessed on any kind of web browser and can also be used without any payment/subscription to the web application. The software used for developing the application is economically feasible as well.

Technical Feasibility

The software Visual Studio Code, Flask framework are needed to be installed in the system to build this application to design the 3d models. Technical feasibility is the software and hardware requirements evaluation of the proposed system. The technology involved in this project is Machine Learning. The language that is used to implement the concepts of Machine Learning is Python Programming and the tool that is used to execute the Python code is Jupyter Notebook (IPython notebook). A minimum of 8GB ram is required to run all this software smoothly. Thus, this application is technically feasible.

Operational Feasibility

The application involves parameters which are design dependent such as sustainability , usability, supportability, disposability, reliability, affordability, maintainability and others. It minimises the drawbacks of the current system by building an application that automatically resolves the user queries and helps to analyse the user data by taking various biomarkers into consideration. The interface of the application is self-understood and on submission of patients` clinical data a result of the analysis is displayed and in addition we also get therapy suggestions and comparison of the results with other ML-DL algorithms.

CHAPTER 6

SYSTEM ANALYSIS AND ARCHITECTURE

6.1 Introduction

Software requirements can be categorised as functional or non-functional needs which are required to be implemented in the system. Functional can be considered as a service that the user is provided with whereas non-functional can be considered as the efficiency that needs to be provided, it can be based on performance.

6.1.1 Document Purpose

Software requirements are needed to be documented as it acts as the roadmap for the product that is being built and gives an idea on the features that are being included in the project and helps in keeping the track of implementation of all the functional and nonfunctional requirements. It can also be used as the foundation for the agreement or the agreement mentions the functionality of the software. It ensures the consistency of the project by acting as the guide. It saves time and money for implementing the software. It helps in ensuring the stability of the software and also provides or leaves out less chances of fail conditions for the software.

6.1.2 Definitions

Normal Cognition : The term normal cognition means that the patient is doing well and doesnt have to be further diagnosed for any kind of disease related to cognitive impairment

Mild Cognitive Impairment: The term is the stage in between expected cognitive decline of normal ageing and the more serious decline of dementia.

Alzheimer's Disease: The disease is a progressive disorder that causes the brain to shrink and brain cells to die.

Vascular Dementia: Vascular Dementia is a decline in thinking skills caused by the conditions that block or reduce blood flow to various regions of the brain, depriving them of functioning properly

3D Convolutional Network: 3D convolutional network is a type of convolution which takes a series of frames that are 2D data or consider a volume of dataset that is 3D as an input with kernel slides in 3 dimensions. 3 Dimensional Convolutions applies filters to the dataset with the size as 3 and the filter moves in 3 directions to calculate the feature map.

Resampling: Resampling is a method in which small samples are repeatedly picked from the original dataset. Statistics and metrics are calculated on each drawn sample to obtain detailed information about the data and these metrics can be considered as the performance of the model.

Random Forest: Random forest is a machine learning model under the category of supervised learning which is used for both regression and classification. It develops decision trees on different samples and collects the majority votes from all the trees developed and outputs the majority one. It works on the principles of the decision tree.

SVM: Support Vector Machine or abbreviated as SVM is a machine learning model categorised under supervised learning. The aim of SVM is to develop a hyper plane in space that distinctly classifies the data points. It can be applied to both regression and classification types of problems.

Naive Bayes: Naïve Bayes classifier uses a mathematical theorem called Bayes' Theorem. It is categorised as a supervised learning algorithm. It predicts based on the probability of the objects considered in the dataset.

6.1.3 Requirement Analysis

Requirement Analysis can also be considered as the requirement engineering. It is defined as the process of determining the end-user requirements on the existing software or the new software. It involves the constant communication between the developers and the end-users which makes the build process as per the need of the users. It reduces the miscommunication chances and reduces the risk of going out of the track from the requirements mentioned by the end-users.

It is also referred to as requirement gathering or requirement capturing in the field of software engineering. Bridging the different requirements and designs of a system can be considered as a software engineering task.

The phase of analysing requirements is important to determine the success rate of the project. The requirements should pose the characteristics of being able to track, test, and measure. It acts as a communication bridge between the developers and end-users to put down their requirements and developers can be right about the requirements from users. This can also act as a proof at the end for any dissatisfaction from end-users. So, it can also be used as a proof.

As part of the requirement phase, we gathered all the requirements and documented them as a Software Requirement Specification. This is documented in such a way that both the technical and non-technical team can understand. Every change that is made in the development process is documented and follows a protocol for the approval process. Requirement Analysis helps in understanding the effort needed from hardware and software perspective.

The aim of the requirement analysis phase is to develop software that meets the customer needs with high quality and to be able to finish in a given budget. Software Requirements helps in improving the standards of the project with the involvement of the specifications required by stakeholders.

Functional requirements include –
The application should be in a state to accept the data from the user as input and process the data accordingly.
The user should be in a state to provide input data to software and the data provided should be in understandable format by the application.
After processing, the application should provide the required prediction result i.e. prediction of ADNI using the details provided to the application by the user.

Non-Functional requirements include –
Scalability: The application should be able to perform well with the larger datasets.
Economic Feasibility: The application should eliminate the costs as there would be minimal need for manual work.
Usability: The application's interface should be easy (user-friendly) to understand and use for end user.

Software requirements include:
• Python3
• Jupyter notebook
• Flask

6.2 System Architecture

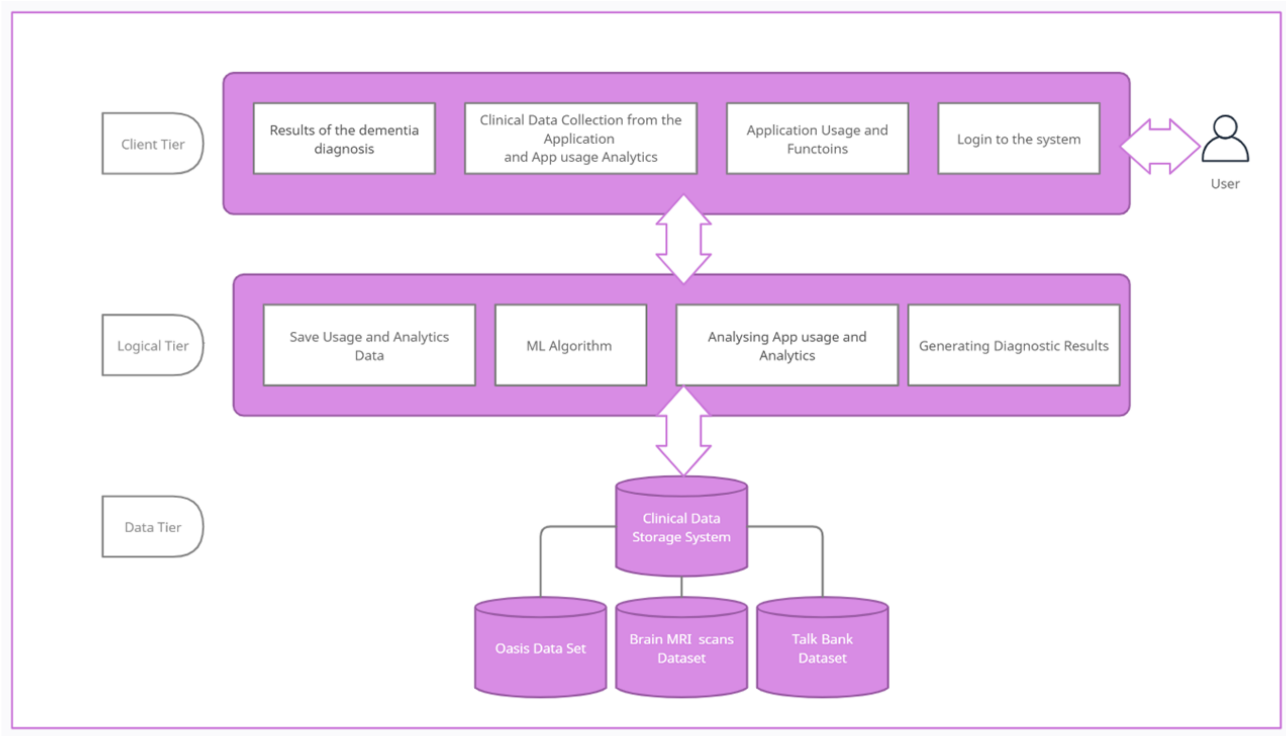


Fig 6.1 Detailed architecture of the proposed system

System Architecture describes the components that are involved in the system and the relation between each component. It helps in understanding how each component interacts with each other in order to complete the functionality. It serves as the blueprint of the system.

It can be considered as the structured solution to cover-up all the operational and technical requirements while optimising the non-functional requirements like the quality, efficiency and performance. A system architecture includes developed sub-systems and several system components that coordinate and execute together in order to successfully run the whole system.

The system architecture can be taken as a conceptual model that defines the structure, behaviour and views of the system. An architecture description is a formal description and representation of a system that is organised in a way that allows for reasoning about the system's structures and behaviours. This System architecture concentrates on the interfaces that exist between the components or the subsystems of the system internally and its environment that is external to it.

There are 3 tiers in the system architecture that is proposed for this project. First tier is the User Tier.

In the user tier the user can view the form under as a part of the clinical data collection from the user. The user provides the data with respect to the scores of dementia, the data with respect to the Dementia test reports and finally fills up the demographic information of the user. Clicking the predict option redirects to the next tier that is the second tier which is called the logical tier.

In the logical tier the data undergoes pre-processing techniques. Then the #D CNN deep learning model is trained. After training, the diagnosis is generated. The logic for training the 3D CNN learning model using the data from the data tier and prediction of diagnosis by the model with the data provided by the user runs in this tier.

The third tier is the data tier. In this tier the deep learning model's data provided by the user runs in this tier.

The third tier is the data tier. In this tier the deep learning model's data is present from which the model is extracted and displayed on the user tier. Data tier is all about the ADNI, OASIS and Dementia Bank datasets, and the target data on the prediction of Dementia.

6.3 Functional Requirements

Functional Requirements can be defined as the description of the services that are to be provided by the software. It describes the functionality of the software. It helps in tracking the system if it's providing all the functionalities up to the expectations of the end-users or not. The errors which are caught in the functional requirements are easier and cheaper to fix. It helps in building a strong base from the initial phases of the project.

- The application should be capable of accepting the data from the user and process it accordingly.
- The user should possess and hence be able to provide the data as required by the application.
- The application should provide the required classification results i.e; detection of Dementia on the basis of the user's input data.
- The results should be displayed on the screen and the user must be able to see them.
- The application should be able to deal with data sets that show imbalance.

6.4 System Analysis

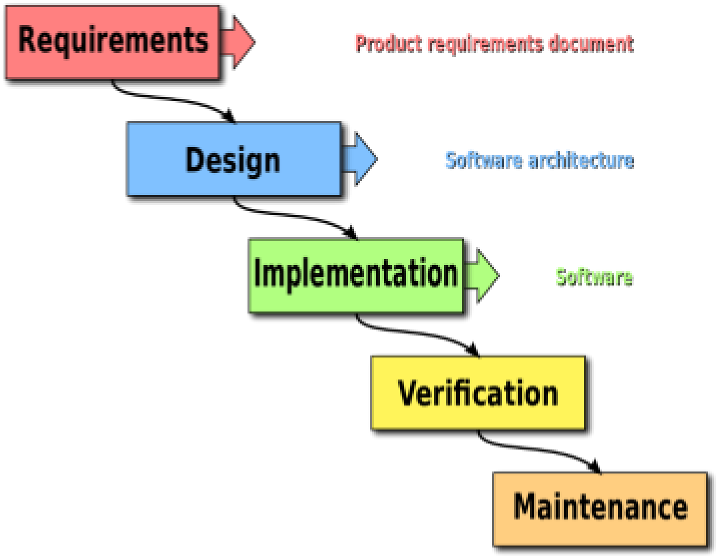


Fig 6.2 Steps involved in the process of analysis of the system

As the development process of the application, we followed the Waterfall model throughout the development. It was the first introduced process model. It is a basic model out of all the existing process models and is extremely easy to apply and understand.

It is also known as a linear sequential life cycle model. In this model, all the phases are implemented in a sequential manner where the phases do not overlap each other. Before each phase begins its previous phase must be completed. It was the first SDLC approach implemented in software development.

The model development is linear and in sequential process. Only after the completion of the phases will we be able to move to the next phase which ensures the strong base line development as the completion of previous phases provides the strong base for the next phase.

The steps in the waterfall model are explained below:

Requirement Analysis: In this phase, all the focus is on the requirements needed to develop software. It can be functional or non-functional. To start with the development of the project, the software engineer must understand the functions required for user interface development and so on. After analysing the entire requirement, these should be documented and presented to the user for

validation from their end. This builds a strong base and covers most of the end user requirements.
Design: After the completion of the requirement analysis phase, the next phase is designing. All the requirements which are gathered are changed in representational format which acts as a blueprint of the software. The design must be capable of meeting the requirements mentioned in the previous stage. This design acts as a blueprint for the development. All the design details must be documented for the next phases.
Implementation: According to the design developed in the previous phase, a coding process is used in implementing the design. Coding uses a programming language and builds the functionality mentioned in the design. In this phase the programmer put the design into technical implementation.
Verification: After implementation we are required to test the implementation part to check if the software is working according to our desired results or not. All the software functions must be tested by testers with various methods available to ensure that the software is error-free and the results of the function should be according to the design of the function.
Maintenance: Software developed should work on a long run and need maintenance. When it is implemented in a production environment it may still have minor errors which might have been skipped previously. Whenever there is any shift in the operating system further development and support is needed. So maintenance can be a continuous or noncontinuous process.
Useful factors: Waterfall model is basic and is easy to implement. Moreover in this model, all the requirement analysis is done at once which provides a broader idea for implementation. Many issues can be resolved in the requirement phase and designing phase.

6.5 Non Functional Requirements

6.5.1 Performance Requirements

Performance Requirements are the requirements which are related to the efficiency of the software that are being expected by the end user. Performance Requirements include the responsiveness of the software, throughput, and storage capacity. It can be considered as the set of requirements which mention how the software should perform under certain circumstances.

6.5.2 Design Constraints

The application needs to be lightweight enough that it can run with minimal delay on a standard web browser. Information provided by the app about the content may be limited to what is available on the internet.

6.5.3 Software System Attributes

Attribute can be considered as the specification that defines a functionality of the object or a file. It can be considered as a metadata for any instance and considered as a property of the property. These are more focused on the quality factors as quality plays a major role in software development. There can be several attributes like performance, security, reliability, availability etc. All these attributes are prioritised and are focused in the software development.

Adaptability

- Users must be able to choose any web browser and the software should support and resize according to each browser for better user interface.
- Users must be able to comfortably use the website.
- Learning curve should be low from an interface perspective.

Reliability

- The application will not crash or hang except as the result of operating system error.
- It should be portable supporting all the hardware changes.

Accuracy:

- The application should be able to perform high accuracy and less bias.

Scalability:

- The application should be able to perform with larger datasets provided as input.

6.6 Software Requirement Specification

Software Requirement Specification, represented as SRS, acts as a specification for a software product and its functionality performance in a specific environment. SRS is used in defining the needs and expectations of the users.

Characteristics of SRS:

- Correctness
- Completeness

- Consistency
- Unambiguousness
- Verifiability

Jupyter Notebook

Jupyter Notebooks is an interactive environment with high computational capabilities, in which we can perform the combination of code execution, execute code to get graphs, perform mathematical operations. The Jupyter Notebook is also known as IPython Notebook. The Ipython Notebook consists of two essential elements:

- A web application
- Notebook documents

Web Application: A browser – based utility for interactive functionality of the documents which combine mathematics and computations to produce the desired results.

Notebook documents: Notebook represents the content that is visible in the web browser when we run the application which includes both the inputs and outputs of the computations and the mathematics logic involved in the documents

Plotting

Displaying plots is a significant feature provided by Jupyter Notebook which is generated using the output that is resulted from the code cells after running them. The Notebook supports matplotlib, a library that is used for the process of plotting

The backend provided by Ipython is called the inlinebackend. This feature is supported by only the Ipython QtConsole and IPython Notebook.

Using this backend feature, the output generated by the commands issued for plotting will be shown inline within the book, directly under the code cell that generated it. The final plots will be stored in the notebook.

Importing .py files

.py format python files will be imported as a notebook with the same base name, but an .ipynb extension format file is located in the notebook directory. The notebook created will have jisu one cell, which will contain all the code in the .py file. Later manually the cells can be divided by using the Edit | Split Cell menu option, or to the Ctrl-m shortcut on the keyboard.

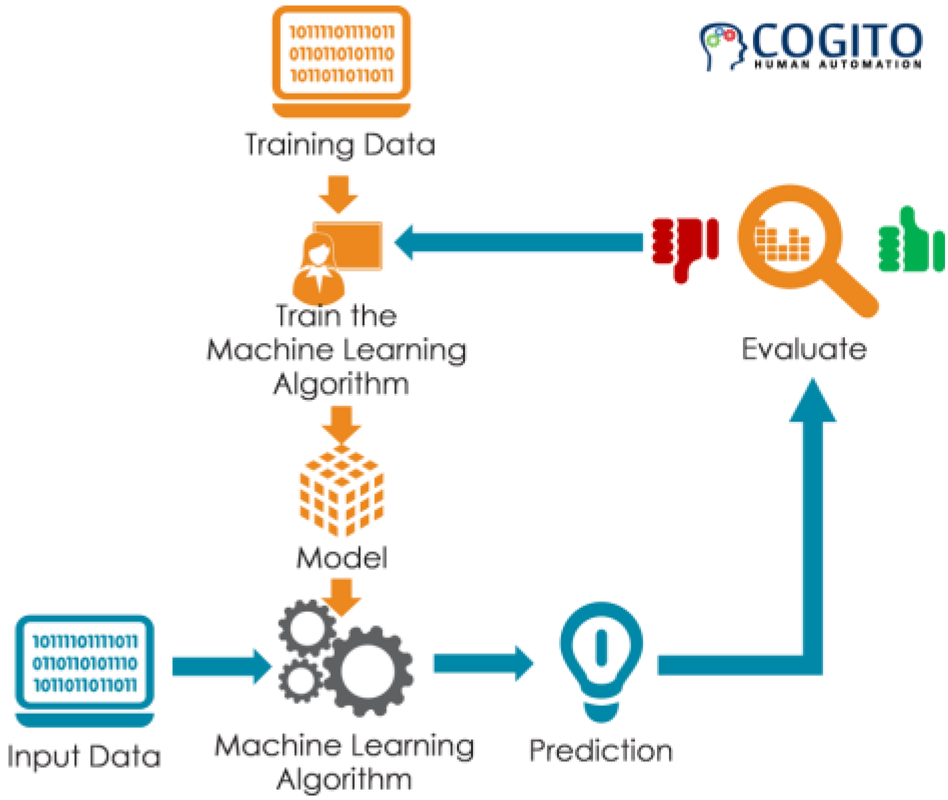


Fig 6.3 Work flow of a basic machine learning model

6.7 Software Requirements

Software Requirements can be considered as the description of the features and functionalities of the end product. Requirements help in understanding the expectations of the end-user from the product. It can be assumed as the process of gathering the software requirements from the client, analysing and documenting them. We collected data from a set of users like what all the functionalities they expect from a product and documented to fulfil their requirements in the development process to develop a quality software which satisfies the end user.

- Softwares : Flask
- Operating System : Windows family
- Technology : Machine Learning
- IDE : Visual Studio Code
- Compatibility : Web Browser

6.8 Hardware Requirements

- Web Browser
- Minimum 8GB Ram Laptop
- Internet Connection

CHAPTER 7

SOFTWARE DESIGN

7.1 UML Diagrams

The Device Architecture Manual describes the application requirements, operating state, application and subsystem functionality, documents and repository setup, input locations, yield types, human-machine interfaces, management reasoning, and external interfaces. The Unified Modelling Language (UML) assists software developers in expressing an analysis model through documents that contain a plethora of syntactic and semantic instructions. A UML context is defined as five distinct viewpoints that present the system from a particularly different point of view.

The elements are the same as the components that can be connected among various conductors to form a whole UML picture, also known as the diagram. As an outcome, understanding the various diagrams is critical in order to put on the knowledge in real world systems. Creating pictures or diagrams of a complex system is the best way to know it. These diagrams have a higher impact on our conception. When looked around, we notice that diagrams aren't a new concept, but are widely used in different forms in various industries.

User Model View

The perspective refers to the structure from clients' perspective. The exam's depiction depicts a situation of utilisation from the perspective of end-clients. The user-view offers a perception into the system from the user's point of view, in which the users' system functionality is modelled in terms and what is expected by the user from the system.

Structural Model View

This layout represents the details and functionality of the device. The static structures are mapped by this software design. activity diagrams, sequence diagrams, and state machine diagrams are all included in this view.

Behavioural Model View

It refers to the social dynamics as framework components, delineating the assortment cooperation between various auxiliary components depicted in the client model and basic model view. Diagrams of UML Behavioural portray the elements of a system which are time dependent and convey the system's dynamic concepts and how they relate to one another. interaction diagrams, and activity diagrams, state–chart diagrams and Use case diagrams are the different types of behavioural diagrams.

Implementation Model View

The essential and actions as frame pieces are discussed in this when they are to be manufactured. The Package diagram is a UML diagram used for representing the developmental view. This is also referred to as the implementation view which labels components of a system using the UML Component diagram.

Environmental Model View

Systemic and functional component of the world where the program is to be introduced was expressed within this. The diagram in the environmental view explains the software model's after-deployment behaviour. This diagram typically explains user interactions and the effects of software on the system. The following diagrams are included in the environmental model: Diagram of deployment.

UML is specifically built through 2 distinct domains, which are:

- Demonstration of UML Analysis, emphasising the auxiliary-model and client model perspectives on framework.
- UML configuration displays that focus upon conducting demonstrations, displaying usage and natural model perspectives.

7.1.1 Definition

UML is a general-purpose visual modelling language that is used to specify, visualise, construct, and document the artefacts of the software system.

7.1.2 UML is a language

It will provide vocabulary and rules for communications and function on conceptual and physical representation. So, it is a modelling language.

7.1.3 UML Specifying

Specifying means building models that are precise, unambiguous, and complete. In particular, the UML addresses the specification of all the important analysis, design, and implementation decisions that must be made in developing and displaying a software-intensive system.

7.1.4 UML Visualisation

The UML includes both graphical and textual representation. It makes it easy to visualise the system and for better understanding.

7.1.5 UML Constructing

UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

7.1.6 Building blocks of UML

The vocabulary of the UML encompasses 3 kinds of building blocks

- Things
- Relationships
- Diagrams

7.1.6.1 Things

Things are the data abstractions that are first-class citizens in a model. Things are of 4 types: Structural Things, Behavioural Things, Grouping Things, Annotational Things.

7.1.6.2 Relationships

Relationships tie things together. Relationships in the UML are Dependency, Association, Generalisation, Specialisation.

7.1.6.3 UML Diagrams

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). There are two types of diagrams, they are:

1. Structural Diagrams

2. Behavioural Diagrams

7.1.6.3.1 Structural Diagrams

The UML‘s four structural diagrams exist to visualise, specify, construct and document the static aspects of a system. I can View the static parts of a system using one of the following diagrams. Structural diagrams consist of Class Diagrams, Object diagrams, Component Diagrams, and Deployment Diagrams.

7.1.6.3.2 Behavioural Diagrams

The UML’s five behavioural diagrams are used to visualise, specify, construct, and document the dynamic aspects of a system. The UML’s behavioural diagrams are roughly organised around the major ways which can model the dynamics of a system. Behavioural diagrams consist of Use case Diagram, Sequence Diagram, Collaboration Diagram, Statechart Diagram, and Activity Diagram.

7.2 UML DIAGRAMS

7.2.1 Use Case Diagram

The drive of the use-case diagram is to apprehend the dynamic aspect of the system. Though, the description is very broad for describing purpose, since the purpose of the other 4 diagrams: activity, sequence, State chart & collaboration) is the same. Investigation is done with a specific purpose that sets it apart from other 4 illustrations.

Use case diagrams are made use of for collecting system's requirements, with external & internal influences. Most of the requirements are designed ones. As an outcome, when analysing a system for gathering the use-cases, functionalities, actors are identified.

When a preliminary task is done, use-case diagrams are shaped for displaying the outside view.

In short, use-case diagrams assist with the following functions:

- Used for system's requirements collection
- Used for getting the bird's-eye sight of a system.
- Regulate both internal & external factors which are influencing the structure.
- Display interaction of requirements as the actors.

Use case diagrams considered requirement analysis for higher levels of the system. While a system's requirements are analysed, in use cases functionalities are captured. Use cases can be defined as system-functionalities written in an organised manner. Actors are the second thing that is relevant to use cases. Actors are the objects which interact with the system.

Actors can be external applications, internal applications or human users. When creating a use case diagram, it should be having these items in mind. .

- As a use case, functionalities will be signified.
- Actors
- Relationships between actors and use-cases

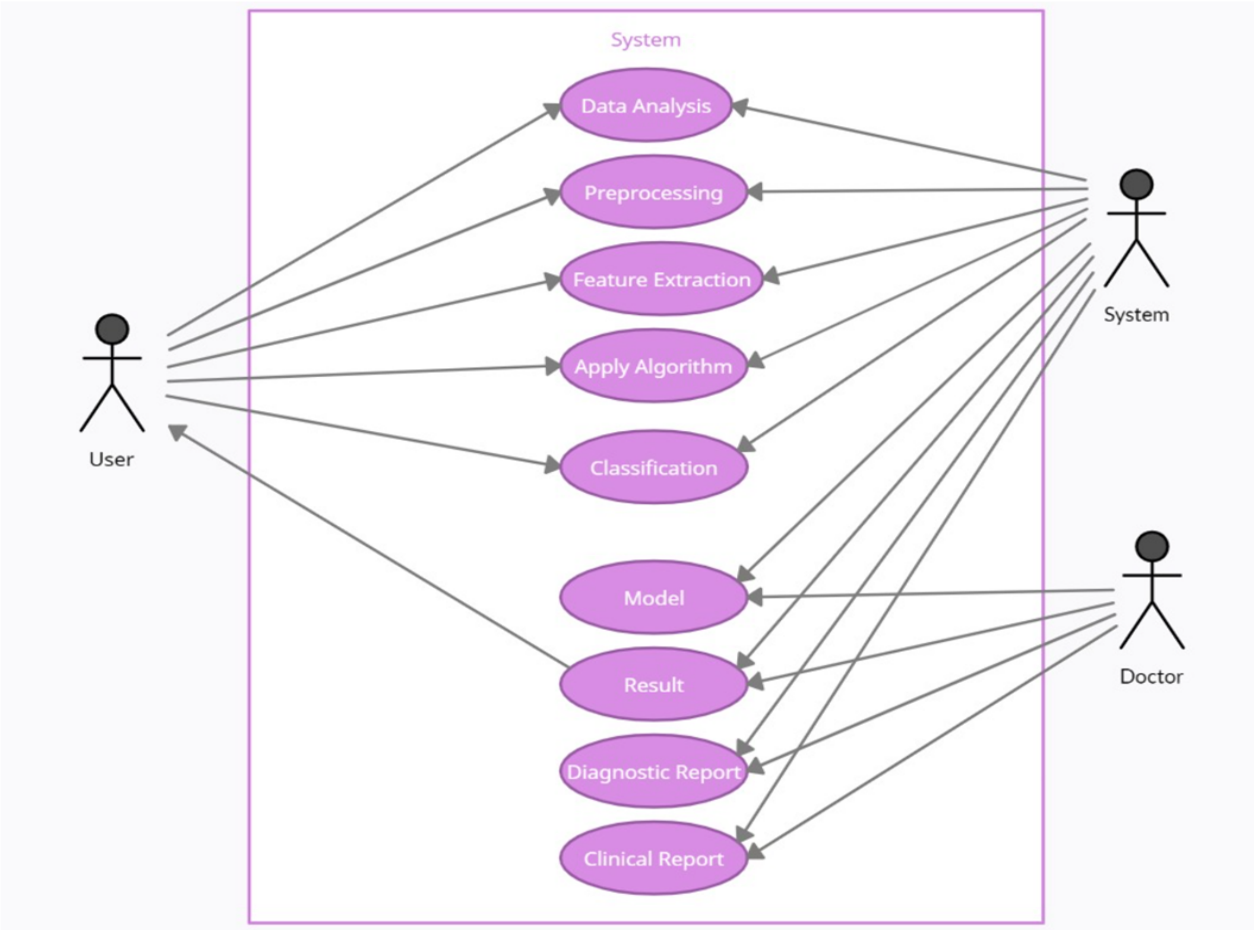


Fig 7.1 Use Case Diagram of the system

Actors

- User
- Admin
- Doctor

Use Cases

Use cases are written instructions on how website users complete their tasks. It describes how the system behaves while responding to requests from the user's perspective. Each use case is represented by a series of simple steps which start with the user's goal and end when that goal is achieved.

Graphical Representation

Use-cases are represented by an oval shape. A better detailed description of the use-case might be as follows:

- The systems pattern behaviour
- A series of related transactions performed by the system as well as an actor
- Sending something useful to actor

Use cases help you understand system requirements and communicate with top users and domain experts. Test the system. The best way to find a use case is to look at the actors and define what the system can do.

Flow of events

A sequence of times can be thought of as a collection of interactions (or opportunities) carried out by the system. They provide daily point-by-point details, published in terms of what the framework can do rather than whether the framework performs the task.

- How and when an employment case starts and ends.
- Interactions between the actor and use-case and the actor
- Information required by the employment case
- Employment case's standard order of events
- An exceptional or different flow.

Construction of Use Case

The framework's behaviour is depicted graphically in outlines of use cases(use cases). The graphs deliver a higher level view of how framework is made use of as seen through the eyes of an untouchable (actor). The utilisation case graph can depict all or some of a framework's work instances.

A use-case diagram may include the following elements:

- "Things" outside the framework - actors
- Use Cases (framework Limits Recognize what the framework should do)

Relationships in use cases

Active-relationships are commonly used in use case diagrams and are also known as behavioural relationships. Communication, inclusion, extension, and generalisation are the 4 basic behavioural relationships types.

1) Communicates

The communicated behavioural relationships connect actors to use cases. Remember that the purpose of the use case is to provide some benefit to the actors in the system. Therefore, it is important to document these interactions between actors and use cases. Lines without arrows connect actors to use cases.

2) Includes

This relationship is also known as a uses relationship that describes a state in which the use case contains behaviour which is shared by multiple use cases. To put it another way, in the other use cases common use cases are included. This relationship indicates a dotted-arrow pointing to use case that is common

3) Extends

The Extends relationships represent situations where a use case behaves to allow a new use case for handling variations/exceptions from this basic use-case. Exceptions to the basic use case are handled by another use case. Arrows connect advanced and basic use cases.

4) Generalises

Generalised relationship denotes that a single thing is more common than other. This connection could exist between two use cases or two actors. The arrow directs to the general-"thing," which is more general than another "thing" in UML.

In this application Use Case diagram,

Actors:

User

- Dementia API
- Dementia dataset
- CNN model to detect Dementia

Use Cases

- Collecting clinical Dementia Data
- Classify Dementia
- Calculate Dementia results
- Display Results

Connections

- A request is sent by the system, requesting user’s data for Dementia classification.
- User communicates with the website to fill up the form fields.
- The pre-processing of data is performed on the ADNI and Dementia Bank dataset.
- The machine learning model to detect Dementia will classify Dementia and calculate the Dementia results.
- The results are displayed to the user using the Dementia API

7.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram for explaining how a set of objects interact with one another (and in which order). Business people and Software developers use the diagrams for understanding the requirements of new documents and systems existing processes. Event scenarios and event diagrams are aliases for sequence diagrams.

Companies and other organisations can benefit from using sequence diagrams as a reference. Create the following sequence diagram

- Describes the particulars of the UML use-case.
- Model the logic of complex procedures, functions, or operations.
- Find out how components and objects interrelate to complete the process.
- Plan & understand specific features of current/future scenarios

The following scenarios lend themselves well to the use of a sequence diagram:

Usage scenarios are expressions that describe how to use the system efficiently. This ensures that the logic of all possible system usage scenarios is executed.

Method logic:

Just as you can use UML sequence diagrams to explore the logic of use cases, you can also use UML sequence diagrams to explore the logic of any function, procedure, or complex process. Sequence diagrams are a great way to represent service logic when you think of a service as a high-level method used by different clients.

Vision sequence diagram:

Any Vision-sequence diagram which is created can be uploaded into Lucid chart. Lucid chartis a great Microsoft Visio alternative that imports .vdx and supports.vsd file imports. Almost all the images on this site's UML section were created with Lucid chart

Object:

An object has a state, a lead, and a personality. The structure and direction of objects that are, for all intents and purposes, indistinguishable are depicted in the fundamental class. In the diagram, each object represents a specific instance of the class. Order case is an object that’s not named

Message:

Message is an exchange of information between 2 articles which causes an event to occur. The transmission of information from the source point of control convergence to the objective point of control convergence is done by a message.

Link:

An existing association between two objects, including class, implying that there is an association between their opposing classes. If an object associates with itself, use the image's hover adjustment.

Lifeline:

As it moves down to show the passage of time. This dashed line represents the events that occur sequentially on the object during the recorded process. Lifelines can start with an engraved rectangle or an actor's symbol.

Actor:

Shows the entities that interact with the system or that are outside the system.

Message Symbols

Synchronous message

A solid line with a solid arrow is used to represent this. This icon is used when the sender needs to wait for a reply to the message before proceeding. The chart should contain both the response and the call.

Asynchronous message

A lined arrowhead with a solid head is used to represent this. Asynchronous messages do not necessitate a response before the sender can proceed. The diagram should only include the call

Delete message

A solid arrowhead with a solid line is trailed by an X. The message has the effect of destroying an object.

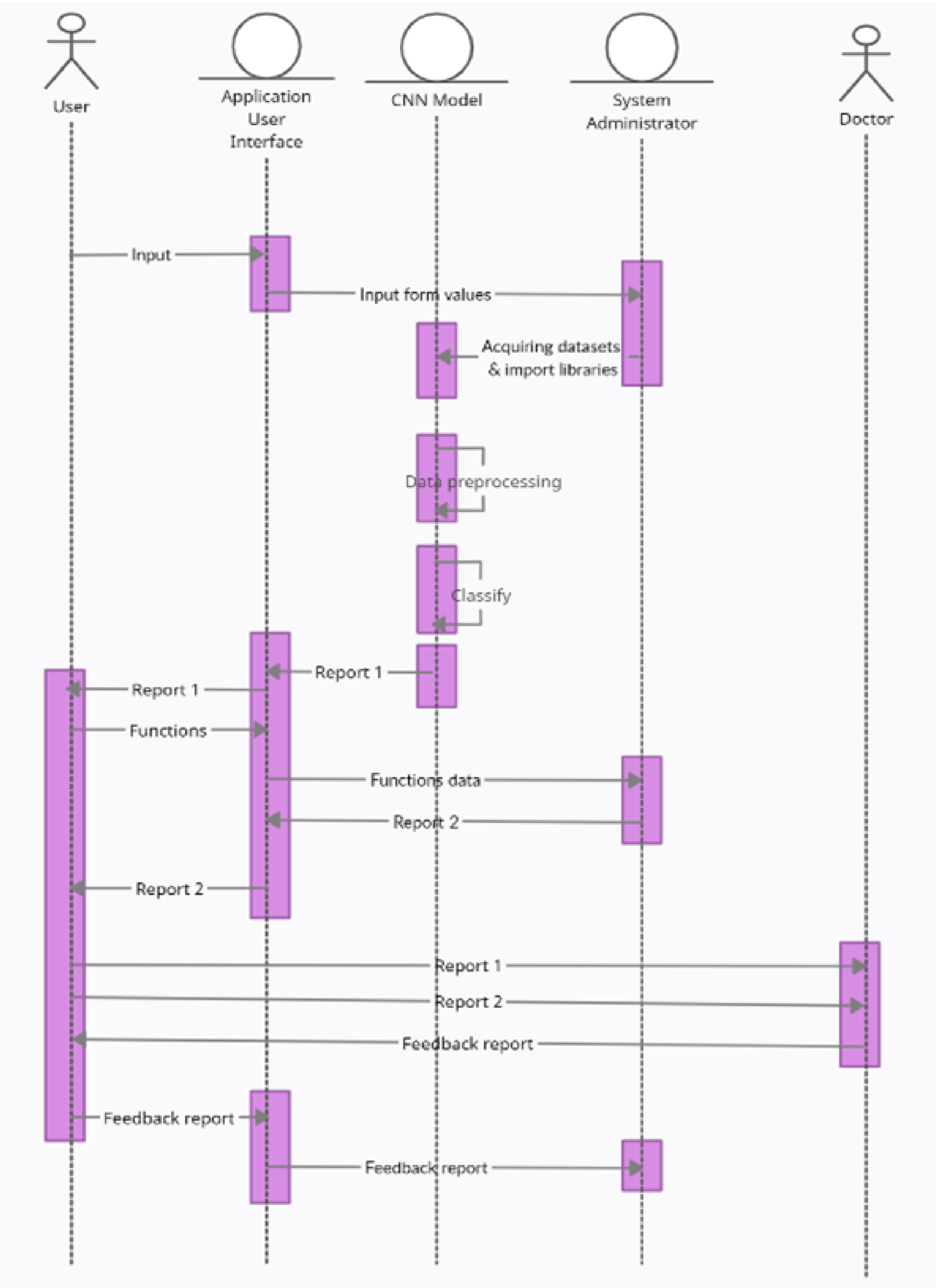


Fig 7.2 Sequence diagram of the system

In the above sequence diagram, the lifelines are:

- User
- Dementia API
- 3D-CNN Model
- Dementia Dataset

The sequence starts from the user, where the user provides the Dementia clinical data. The Dementia API collects this data and sends this data to the 3D-CNN model. The Dementia dataset is pre-processed by removing noise and splitting the dataset into testing and training sets. Modelling and testing of ML classifiers is done and the Dementia results are predicted. Results regarding Dementia are exhibited to an user via Dementia API. Sequence starts with the user moving through the Dementia API, 3D-CNN model, Dementia dataset and returns to the user.

7.2.3 Class Diagram

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualise, describe, document various different aspects of the system, and also construct executable software code.It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

Class diagrams are often made of use when modelling object-oriented systems as they are the only UML diagrams which can be mapped directly to an object oriented language. The class diagram describes the operations and attributes of the class, as well as the constraints. It is imposed on the system. A collection of classes visual representation is associations, collaborations, class diagram, interfaces, and constraints. Also called a structural diagram.

The aim of the class diagram is for representing a static-view of your application. Class diagrams are often used during construction as those are the only diagrams that can be directly mapped to an object oriented language. UML diagrams, such as sequence diagrams and activity diagrams, can only represent the sequence flow of an application. However, the class diagram is a little different. This is the most widely used UML diagram in the programming community.

The purpose of the class diagram is to represent a static view of your application. Class diagrams are often used during construction because they are the only diagrams that can be mapped directly to an object-oriented language. UML diagrams, such as activity diagrams and sequence diagrams, can only represent the sequence flow of an application. However, the class diagram is a little different. This is the most widely used UML diagram in the programming community.

The class diagram’s purpose is briefed as below:

- Design and analysis of an application's static view.
- Describe the system responsibilities.
- The foundation for deployment and component diagrams.
- Both forward and reverse engineering are used.

Class diagrams take many properties to reflect when drawing, but the diagram is viewed in the top level in this case. The class diagram is a graphical representation of a static view of a system which represents various aspects of application. Entire system is represented by the collection of the class diagrams.

- The following points should be considered while creating the class diagram:
- The name of a class diagram that is meaningful to describe a system aspect.
- The element & its relationship may get identified ahead of time.
- Each class responsibility: methods and attributes would be clearly defined.
- The characteristics number for every class might be specified, as unnecessary properties may complicate the diagram.
- While describing the specific aspect of the diagram, use notes. Developer/coder should understand the drawing at the end.
- In conclusion, beforehand the diagram is to be drawn on plain paper before creating a final version and to make sure proper accuracy is maintained, it should be reworked as many times as it can be done.

Aggregation or a-part-of relationship:

It refers to situations in which a classification is made up of various class segments. A classification made up of different classes does not function as a whole. It's extremely difficult to keep going. This relationship's fundamental characteristics are transitivity and hostility to evenness.

Below are the questions whose answers determine the distinction between partial and full relationships :

- Does the class part have a place in the problem area?
- Is the part class subject to the framework's responsibilities?
- Is the part class capable of catching a serious single worth? (If not, simply incorporate it as a class trait).
- Does this provide valuable deliberation for dealing with the issue?

Assembly:

It is assembled from that part and has the fact of physical assembly.

Container:

The physical entire envelope, on the other hand, is not made up of physical parts.

Member of the collection:

The theoretical entire is made up of physical or applied parts. Empty precious stones speak to the container and collection, but strong jewels speak to the creation.

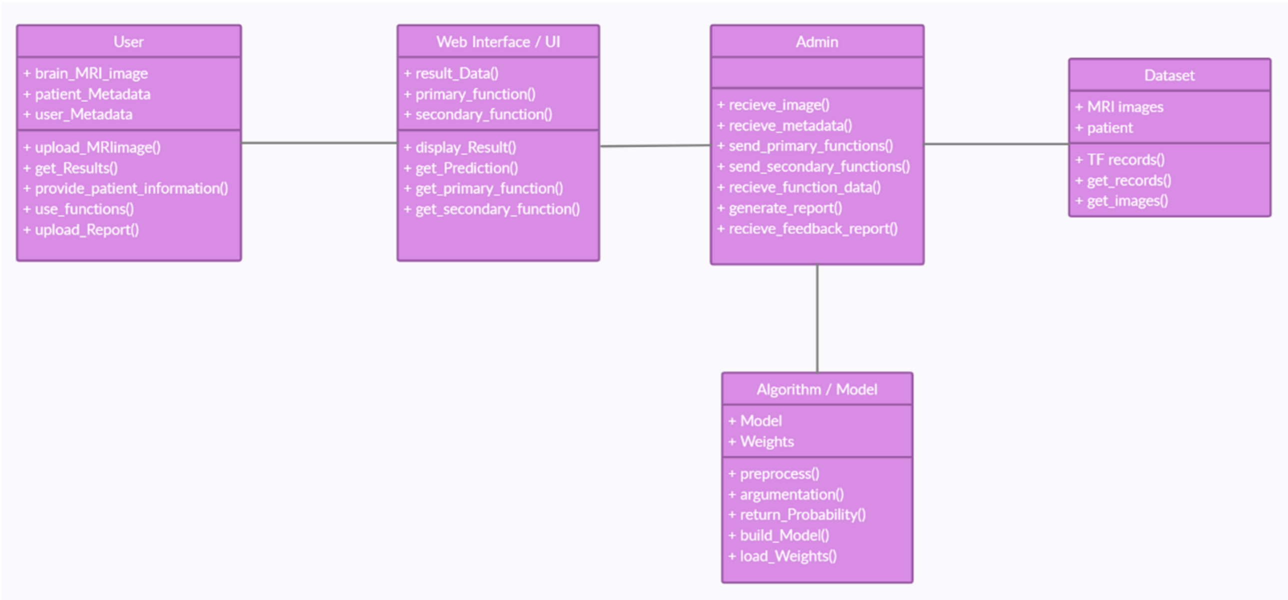


Fig 7.3 Class diagram of the system

The class diagram shows that the application is connected to the user class, Dementia API class and Dementia Dataset directly. User class has the result attribute which is a string and ClinicalData attribute which is a list of input fields and the function GetResultData(ClinicalData). We have the Dementia API class with the attribute ClinicalData and SendDataForTest(ClinicalData) function to receive the data from the User class. The Dementia API had ClinicalData, PreProcessedTrainingData and PreProcessedTestingData as attributes. It has the function CreateModel(PreProcessedTrainingData, PreProcessedTestingData) to trigger the build for the 3D CNN model and Prediction(ClinicalData) to call for the prediction of Dementia diagnosis.

CHAPTER 8

IMPLEMENTATION

The implementation of this project has been divided into three modules - namely:

- Classification of the collected data - The information required for the model to process and generate the classification of Dementia condition is obtained from the input data given by the users via the interface of the system. Now the collected information is augmented to enhance the images by reducing the resolution differences, removing distortions and transposing the image's projection. Then the data is passed through 3 different machine learning models namely Random Forest Classifier, 3D CNN model and NLP model which are primarily responsible for generating the target classes.

- Usable features for the demented patients - Based on the severity of the demented patient the features to be used varies as the purpose of these features vary from person to person.

- Detailed analysis of the data - The final diagnosis results are generated and displayed to the user. A comparative analysis and recommendation results are also presented to the user.

8.1 Data Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

A real-world data generally contains noises, missing values, and maybe in an unusable format that cannot be directly used for machine learning models. Data preprocessing is required for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

There are mainly two ways to handle missing data, which are:

(i) By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give an accurate output.

(ii) By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it in the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

```
df.isna().sum()
```

```
Subject ID     0
MRI ID         0
Group          0
Visit          0
MR Delay       0
M/F            0
Hand           0
Age            0
EDUC           0
SES           19
MMSE           2
CDR            0
eTIV           0
nWBV           0
ASF            0
dtype: int64
```

Fig 8.1 Identifying empty fields in raw data

```
sum(df.duplicated())
```

```
0
```

Fig 8.2 Summation of all duplicate values in raw data

```
df["SES"].fillna(df["SES"].median(), inplace=True)
df["MMSE"].fillna(df["MMSE"].mean(), inplace=True)
```

```
sns.set_style("whitegrid")
ex_df = df.loc[df['Visit'] == 1]
sns.countplot(x='Group', data=ex_df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f505a496d50>
```

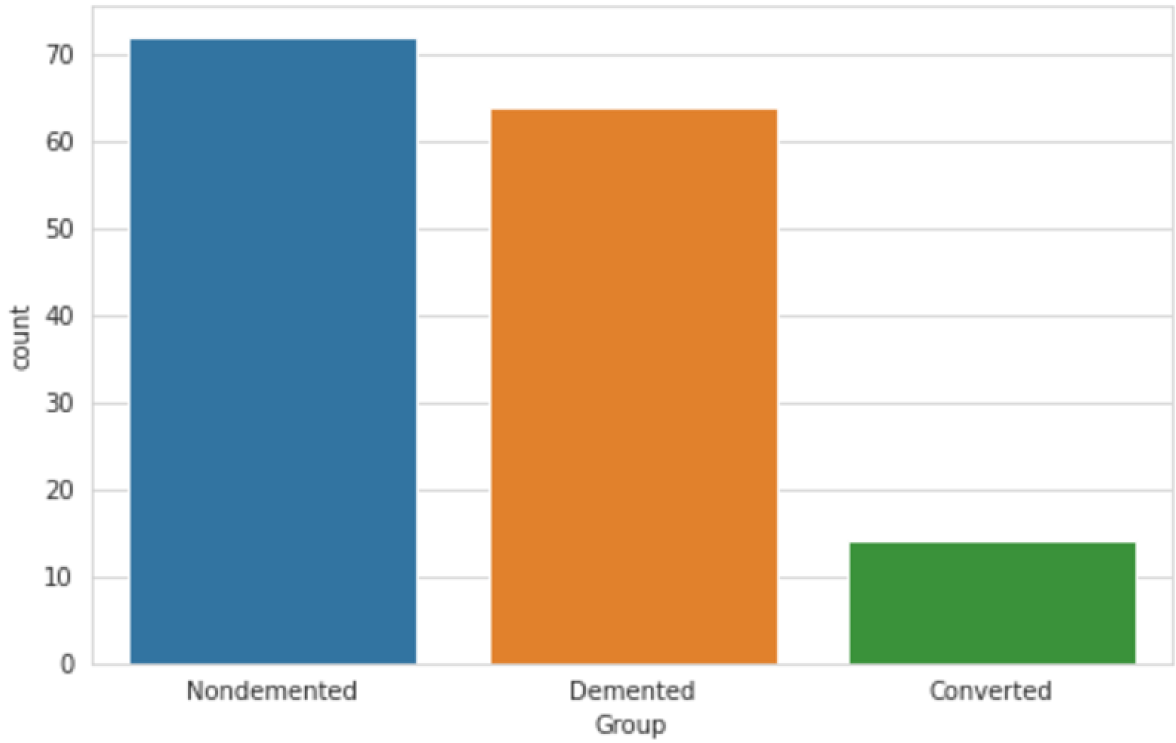


Fig 8.3 Bar graph representing how many people have dementia

```
ex_df['Group'] = ex_df['Group'].replace(['Converted'], ['Demented'])
df['Group'] = df['Group'].replace(['Converted'], ['Demented'])
sns.countplot(x='Group', data=ex_df)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guid
  """Entry point for launching an IPython kernel.
<matplotlib.axes._subplots.AxesSubplot at 0x7f505a4a0d50>
```

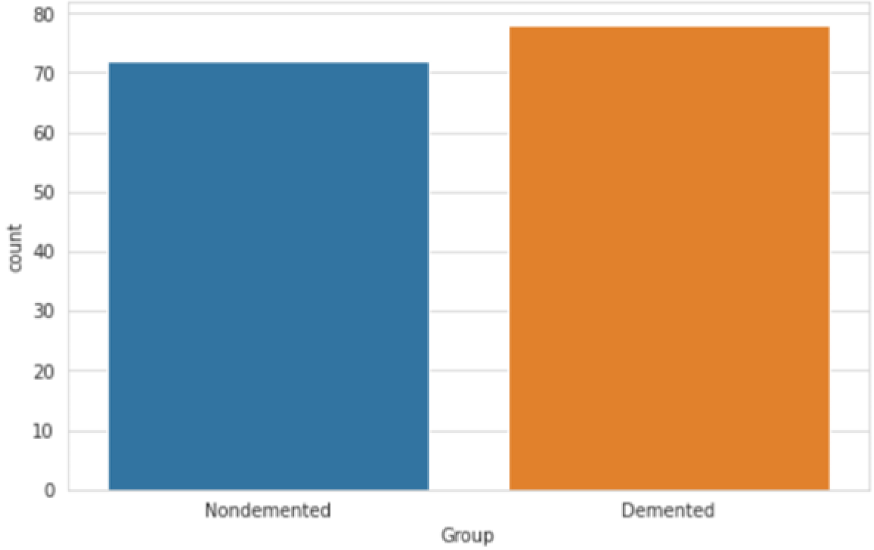


Fig 8.4 Bar graph after converting Converted group into Demented

```
def bar_chart(feature):
    Demented = ex_df[ex_df['Group']=='Demented'][feature].value_counts()
    Nondemented = ex_df[ex_df['Group']=='Nondemented'][feature].value_counts()
    df_bar = pd.DataFrame([Demented,Nondemented])
    df_bar.index = ['Demented','Nondemented']
    df_bar.plot(kind='bar',stacked=True, figsize=(8,5))
    print(df_bar)

bar_chart('M/F')
plt.xlabel('Group',fontsize=13)
plt.xticks(rotation=0,fontsize=12)
plt.ylabel('Number of patients',fontsize=13)
plt.legend()
plt.title('Gender and Demented rate',fontsize=14)
```

Fig 8.5 Code to group the dementia patients based on gender

```
             M   F
Demented    40  38
Nondemented 22  50
Text(0.5, 1.0, 'Gender and Demented rate')
```

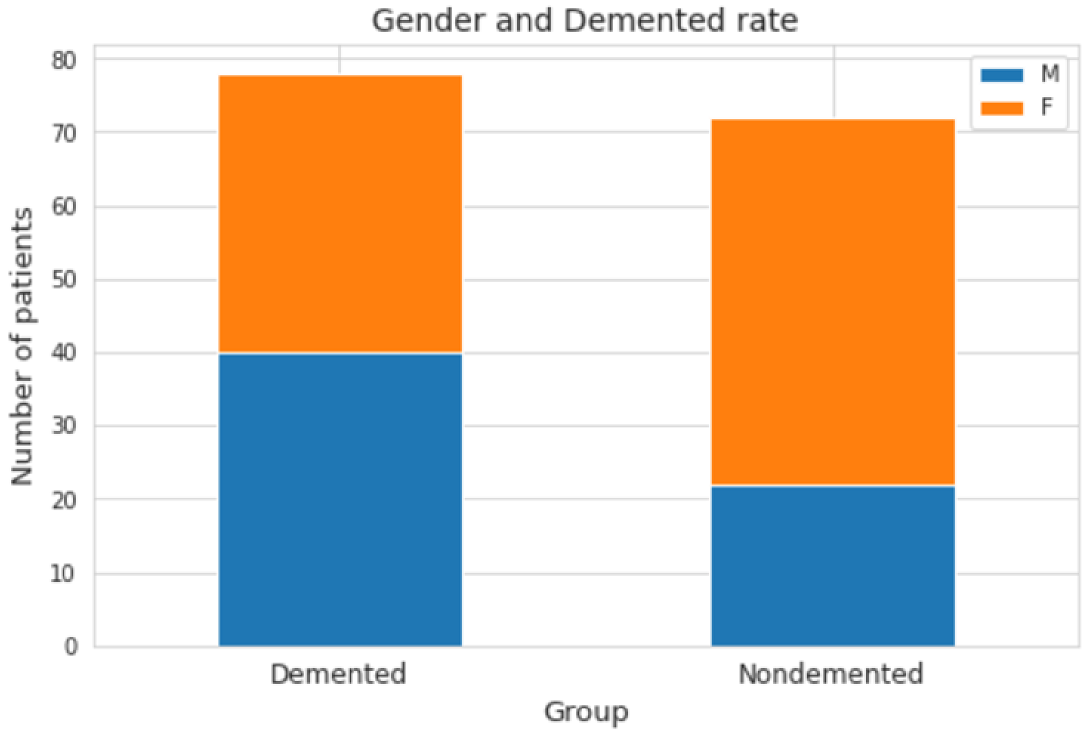


Fig 8.6 Bar graph representing the grouping of dementia patients based on gender

```
plt.figure(figsize=(10,5))
sns.violinplot(x='M/F', y='CDR', data=df)
plt.title('Violin plots of CDR by Gender',fontsize=14)
plt.xlabel('Gender',fontsize=13)
plt.ylabel('CDR',fontsize=13)
plt.show()
```

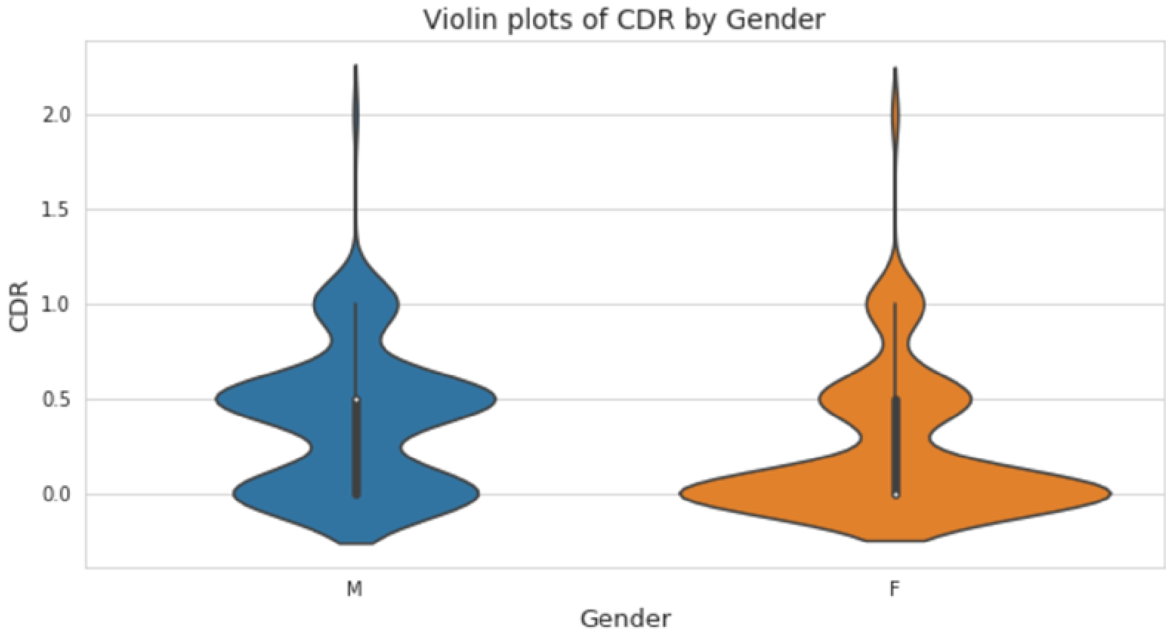


Fig 8.7 Violin plots of CDR by gender

```
ax = sns.countplot(x='Age', data=ex_df)
ax.figure.set_size_inches(18.5, 5)
```

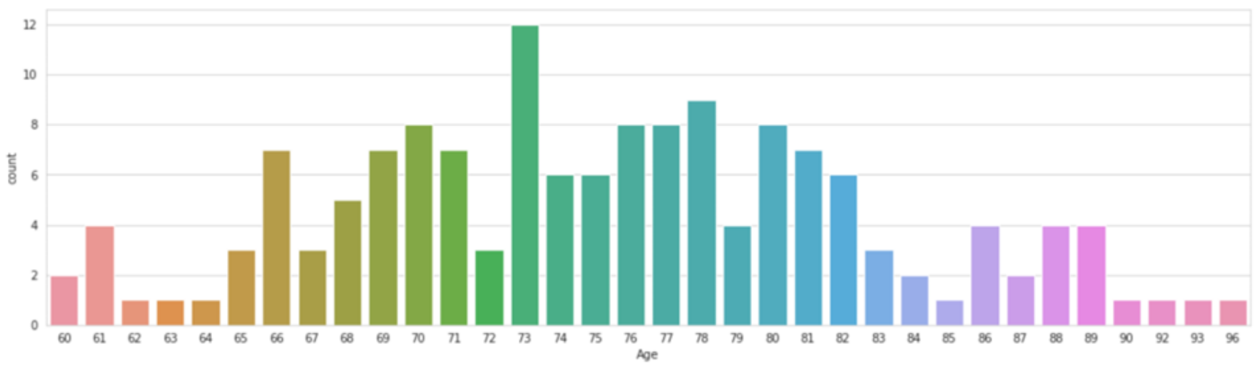


Fig 8.8 Bar graph grouping the dementia patients based on age

```
plt.figure(figsize=(10,5))
sns.violinplot(x='CDR', y='Age', data=df)
plt.title('Violin plot of Age by CDR',fontsize=14)
plt.xlabel('CDR',fontsize=13)
plt.ylabel('Age',fontsize=13)
plt.show()
```

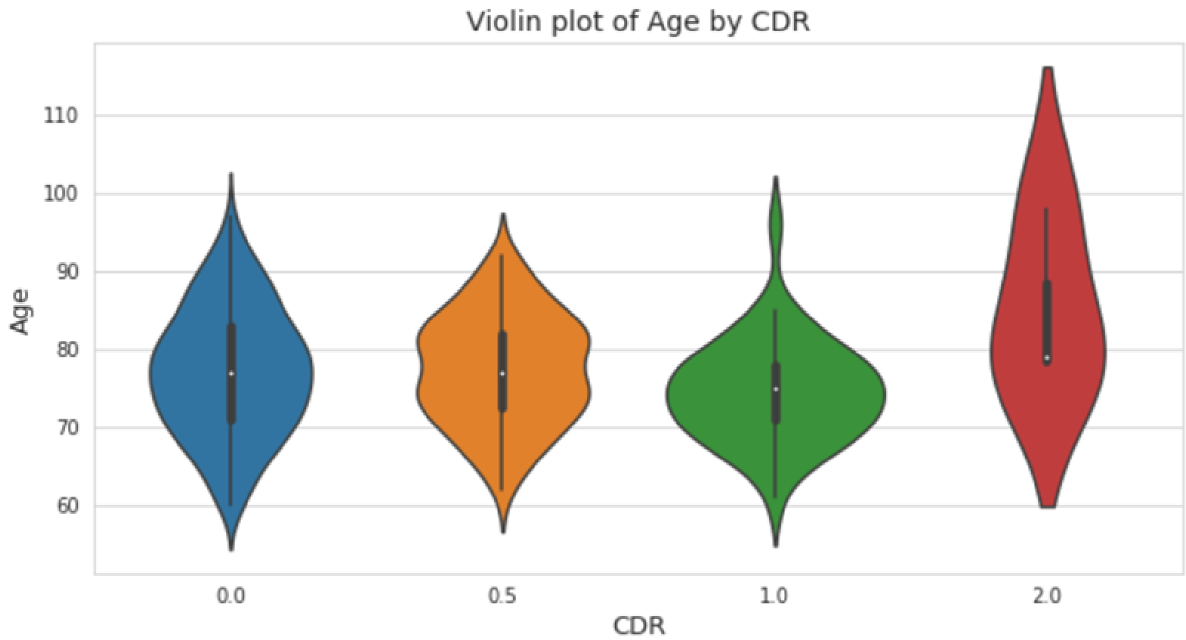


Fig 8.9 Violin plot of age by CDR

```
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'Age',shade= True)
facet.set(xlim=(0, df['Age'].max()))
facet.add_legend()
plt.xlim(50,100)
```

```
(50.0, 100.0)
```

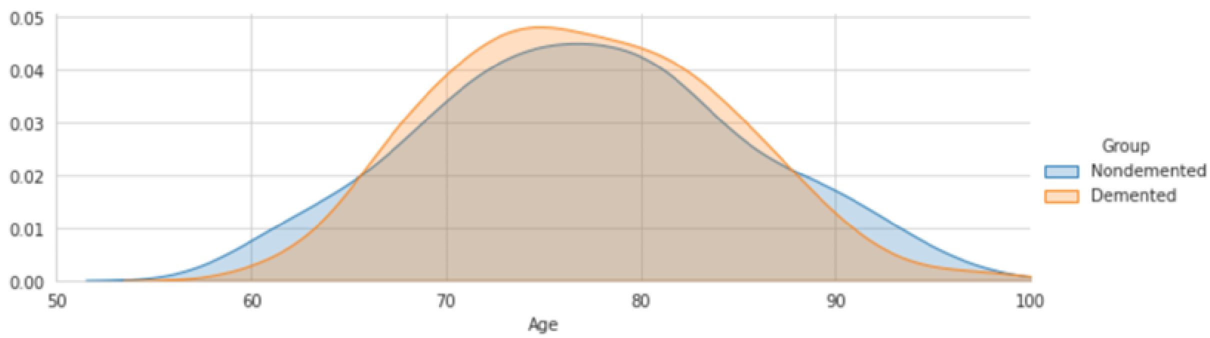


Fig 8.10 Facet grid chart representing patient groups based on age

```
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'MMSE',shade= True)
facet.set(xlim=(0, df['MMSE'].max()))
facet.add_legend()
plt.xlim(15.30)
```

(15.3, 30.0)

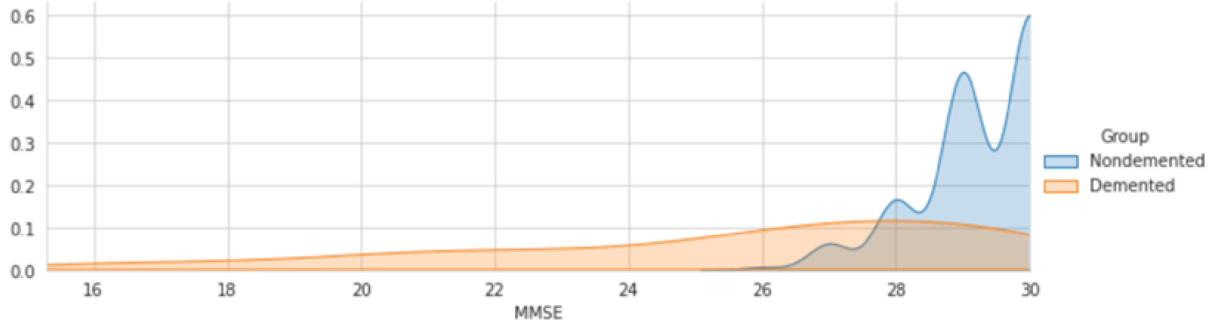


Fig 8.11 Facet grid chart representing patient groups based on MMSE

```
def outliers_iqr(ys):
    quartile_1, quartile_3 = np.percentile(ys, [25, 75])
    iqr = quartile_3 - quartile_1
    lower_bound = quartile_1 - (iqr * 1.5)
    upper_bound = quartile_3 + (iqr * 1.5)
    return np.where((ys > upper_bound) | (ys < lower_bound))

list_atributes = ['MR Delay','EDUC', "SES", "MMSE", 'eTIV', "nWBV", "ASF"]
print("Outliers: \n")
for item in list_atributes:
    print(item,': ',outliers_iqr(df[item]))
```

```
Outliers:

MR Delay :  (array([ 32,  71,  75, 153, 159, 160, 265, 369]),)
EDUC :  (array([107, 108, 109]),)
SES :  (array([136, 137, 138, 161, 162, 179, 180]),)
MMSE :  (array([  4,  25,  26,  43,  44,  51,  52,  60,  88,  89,  90,  93,  94,
        97,  98,  99, 100, 101, 105, 106, 138, 162, 172, 173, 184, 185,
       186, 222, 225, 226, 231, 232, 234, 251, 299, 300, 316, 317, 328,
       332, 360, 366]),)
eTIV :  (array([0, 1]),)
nWBV :  (array([], dtype=int64),)
ASF :  (array([282]),)
```

Fig 8.12 Code to find the outliers in each of the columns

```python
from pylab import rcParams
rcParams['figure.figsize'] = 8,5
cols = ['Age','MR Delay', 'EDUC', 'SES', 'MMSE', 'CDR','eTIV','nWBV','ASF']
x=df.fillna('')
sns_plot = sns.pairplot(x[cols])
```

Fig 8.13 Code to plot all the feature values in a pair-wise manner

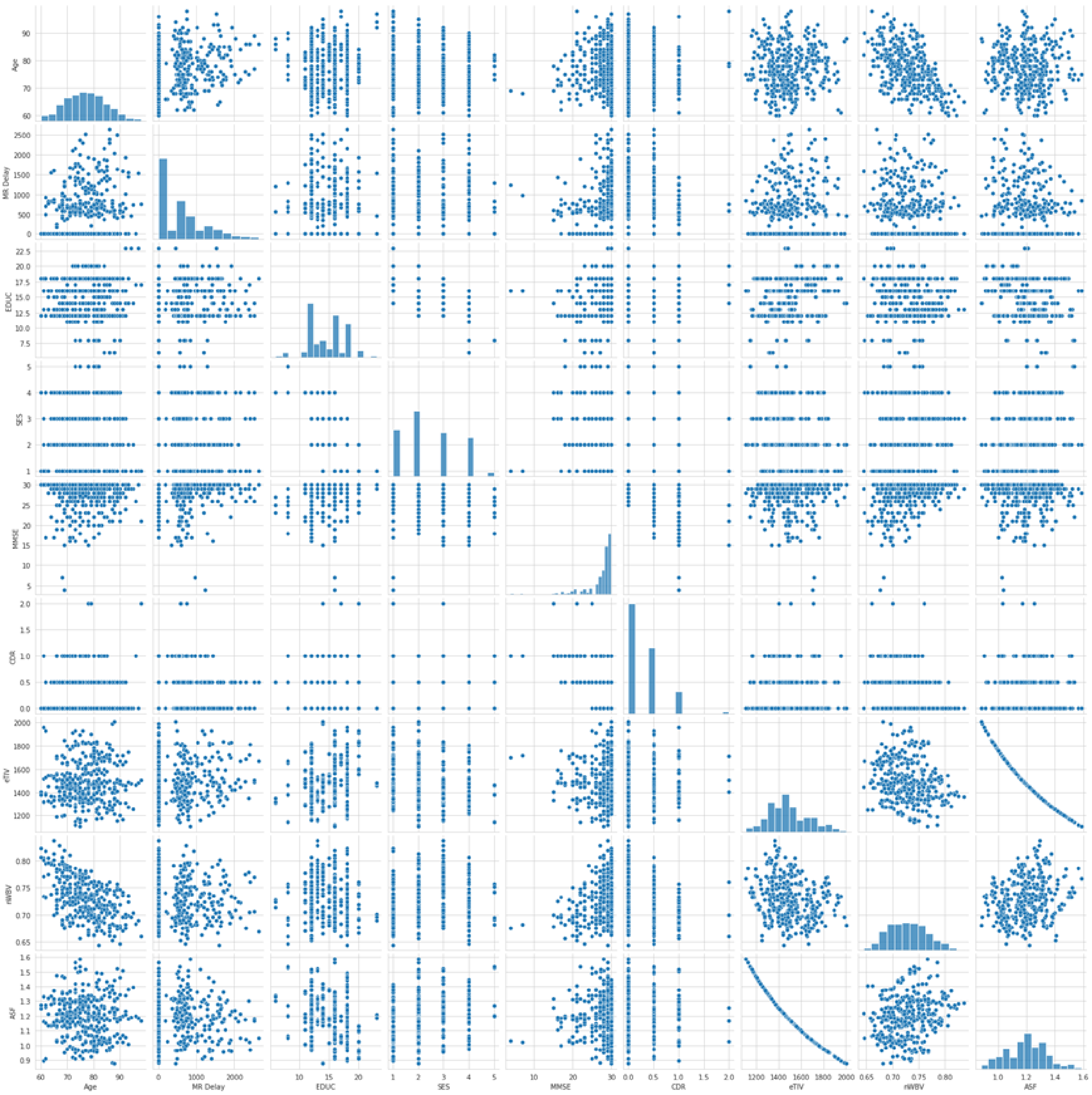


Fig 8.14 Graphs representing all the column values plotted in a pair-wise manner

```
fig, axes = plt.subplots(2,3,figsize = (16,6))
fig.suptitle("Box Plot",fontsize=14)
sns.set_style("whitegrid")
sns.boxplot(data=df['SES'], orient="v",width=0.4, palette="colorblind",ax = axes[0][0]);
sns.boxplot(data=df['EDUC'], orient="v",width=0.4, palette="colorblind",ax = axes[0][1]);
sns.boxplot(data=df['MMSE'], orient="v",width=0.4, palette="colorblind",ax = axes[0][2]);
sns.boxplot(data=df['CDR'], orient="v",width=0.4, palette="colorblind",ax = axes[1][0]);
sns.boxplot(data=df['eTIV'], orient="v",width=0.4, palette="colorblind",ax = axes[1][1]);
sns.boxplot(data=df['ASF'], orient="v",width=0.4, palette="colorblind",ax = axes[1][2]);
```

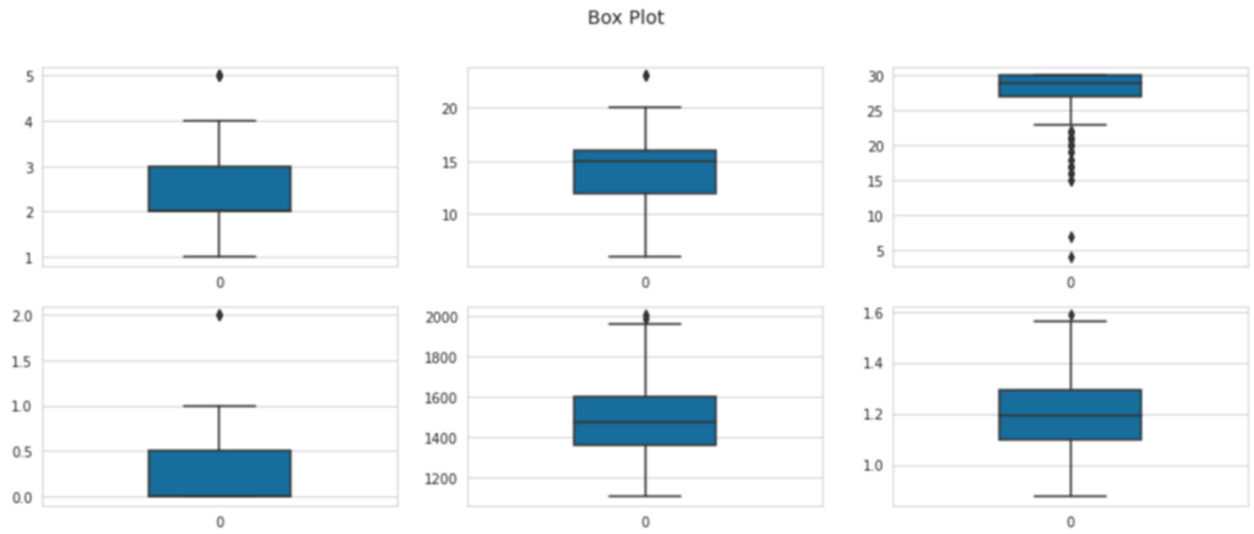


Fig 8.15 Box plots showing the IQR (Interquartile range)

```
group_map = {"Demented": 1, "Nondemented": 0}

df['Group'] = df['Group'].map(group_map)
df['M/F'] = df['M/F'].replace(['F','M'], [0,1])
```

Fig 8.16 Code displaying the conversion of character data into numeric values

```
def plot_correlation_map( df ):
    corr = df.corr()
    _ , ax = plt.subplots( figsize =( 12 , 10 ) )
    cmap = sns.diverging_palette( 240 , 10 , as_cmap = True )
    _ = sns.heatmap(corr,cmap = cmap,square=True, cbar_kws={ 'shrink' : .9 }, ax=ax, annot = True, annot_kws = { 'fontsize' : 12 })
```

```
plot_correlation_map(df)
```

Fig 8.17 Code to plot the correlation map of the data

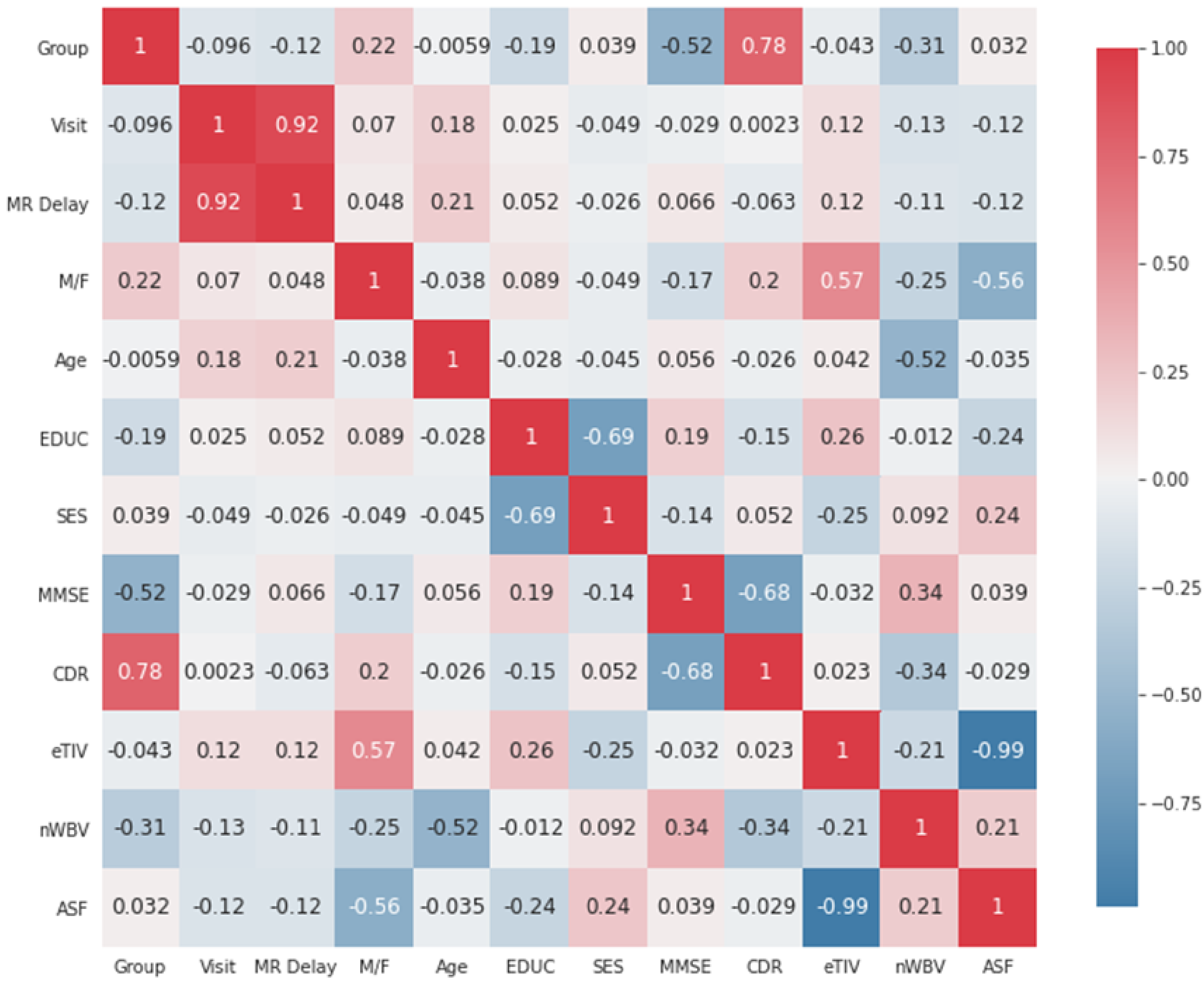


Fig 8.18 The correlation map of the data

```
from sklearn.preprocessing import LabelEncoder
for column in df.columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
```

```
from sklearn.model_selection import train_test_split

feature_col_names = ["M/F", "Age", "EDUC", "SES", "MMSE", "eTIV", "nWBV", "ASF"]
predicted_class_names = ['Group']

X = df[feature_col_names].values
y = df[predicted_class_names].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)
```

Fig 8.19 Code to represent columns encoded into numeric values

```
from sklearn import metrics
def plot_confusion_metrix(y_test,model_test):
    cm = metrics.confusion_matrix(y_test, model_test)
    plt.figure(1)
    plt.clf()
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Wistia)
    classNames = ['Nondemented','Demented']
    plt.title('Confusion Matrix')
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    tick_marks = np.arange(len(classNames))
    plt.xticks(tick_marks, classNames)
    plt.yticks(tick_marks, classNames)
    s = [['TN','FP'], ['FN', 'TP']]
    for i in range(2):
        for j in range(2):
            plt.text(j,i, str(s[i][j])+" = "+str(cm[i][j]))
    plt.show()
```

Fig 8.20 Code to plot the confusion matrix on the data

```
from sklearn.metrics import roc_curve, auc
def report_performance(model):

    model_test = model.predict(X_test)

    print("\n\nConfusion Matrix:")
    print("{0}".format(metrics.confusion_matrix(y_test, model_test)))
    print("\n\nClassification Report: ")
    print(metrics.classification_report(y_test, model_test))
    plot_confusion_metrix(y_test, model_test)

def roc_curves(model):
    predictions_test = model.predict(X_test)
    fpr, tpr, thresholds = roc_curve(predictions_test,y_test)
    roc_auc = auc(fpr, tpr)

    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' % roc_auc)
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic')
    plt.legend(loc="lower right")
    plt.show()
```

Fig 8.21 Code to report the performance of the model

8.2 Classification of the collected data

The Data Classification is an ensemble classification of Random Forest Classification over dataset with features such as age, MMSE score, years of education, socio economic status, clinical dementia rating, estimated total intracranial volume, normalise whole brain volume, atlas scaling factor and 3D CNN classification method over MRI images and usage of NLP LSTM method to identify biomarkers over the answers given for the mini mental state exam tests.

8.2.1 Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each tree in the random forest spits out a class prediction and the class with the most votes becomes the model's prediction. It utilises ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. It uses bagging and features randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

The reason that the random forest model works so well is that a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

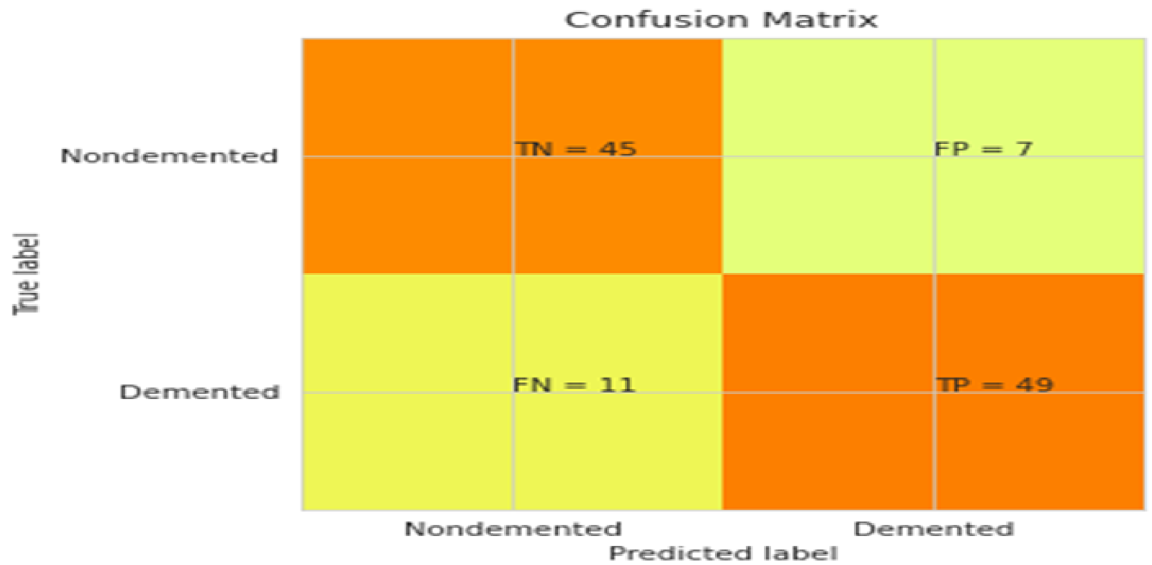
The random forest is a classification algorithm consisting of many decision trees. It uses bagging and features randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

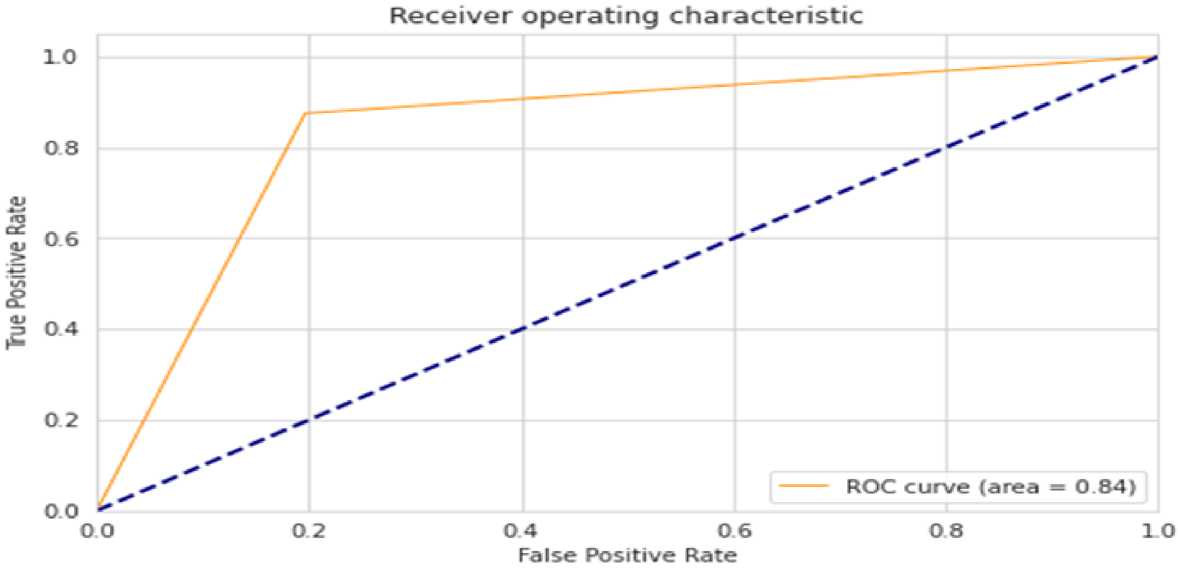
In other words, a random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
Random forest algorithms can be used for both classifications and regression tasks. It provides higher accuracy through cross-validation. A random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

RandomForestClassifier

```
[ ]  rfc=RandomForestClassifier(random_state=42)

     param_grid = {
         'n_estimators': [200],
         'max_features': ['auto'],
         'max_depth' : [4,5,6,7,8],
         'criterion' :['gini']
     }

     CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv= 5,scoring = 'roc_auc')
     CV_rfc.fit(X_train, y_train.ravel())
     print("Best parameters set found on development set:")
     print(CV_rfc.best_params_)
     report_performance(CV_rfc)
     roc_curves(CV_rfc)
     accuracy(CV_rfc)
```

```
Best parameters set found on development set:
{'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'n_estimators': 200}


Confusion Matrix:
[[45  7]
 [11 49]]


Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.87      0.83        52
           1       0.88      0.82      0.84        60

    accuracy                           0.84       112
   macro avg       0.84      0.84      0.84       112
weighted avg       0.84      0.84      0.84       112
```





8.2.2 3D CNN Classifier

The model proposed by the system uses 3D Convolutional Neural Networks which is nothing but the 3D CNN classifier. The pre-processed data is taken by the classifier. Then it is divided in a way where seventy percent of the data goes as a training dataset used for the model's training and thirty percent of the other data is used as the testing dataset for testing the trained model.

The model uses a cumulative of six neural layers. Among these layers, we use the first layer as the input layer and the next set of four neural layers constitute the hidden layer and ultimately the last layer is taken as the output layer. The model included several modules used for translation of complex images and preserves the spatial and temporal properties of the data. A 90% accuracy can be secured with this proposed system.

Some of the pros provided by the currently proposed system is as follows 1) Doesn't compromise on the temporal and spatial aspects of the data. 2) Mining the data for hidden biomarkers that are crucial for generating more efficient diagnosis results. 3) Deal with imbalanced and varied sized datasets. 4) Faster computation capability with higher accuracy and better performance.

A robust diagnostic of a particular disease should adapt to various datasets, such as, e.g., MRI scans collected by several patient groups, as to diminish discrepancies in data distributions and biases toward specific groups. Deep learning aims to decrease the use of domain expert knowledge in designing and extracting most appropriate discriminative features.

Several popular non-invasive neuroimaging tools, such as structural MRI (sMRI), functional MRI (fMRI), and positron emission tomography (PET) have been investigated for developing such a system. The latter extracts features from the available images, and a classifier is trained to distinguish between different groups of subjects, e.g., AD, mild cognitive impairment (MCI), and normal control (NC) groups.

The sMRI has been recognized as a promising indicator of AD progression. Compared to the known diagnostic systems, the proposed system employs a deep 3D convolutional neural network (3D-CNN) pretrained by 3D Convolutional Autoencoder (3D-CAE) to learn generic discriminative AD features in the lower layers. This captures characteristic AD biomarkers and can be easily adapted to datasets collected in different domains. To increase the specificity of features in upper layers of 3D-CNN, the discriminative loss function is enforced on upper layers (deep supervision).
The proposed deep 3D CNN for learning generic and transferable features across different domains is able to detect and extract the characteristic AD biomarkers in one (source) domain and perform task-specific classification in another (target) domain. The proposed network combines a generic feature-extracting stacked 3D-CAE, pre-trained in the source domain, as lower layers with the upper task-specific fully connected layers, which are fine-tuned in the target domain.

To overcome the feature extraction limitations of the conventional approaches, the 3D-CAE learns and automatically extracts discriminative AD features capturing anatomical variations due to the AD. The pre-trained convolutional filters of the 3D-CAE are further adapted to another domain dataset, e.g., to the ADNI after pre-training on the CADDementia. Then the entire 3D-CNN is built by stacking the pre-trained 3DCAE encoding layers followed by the fully connected layers, which are fine-tuned to boost the task-specific classification performance by employing deep supervision. Such models are called Deeply Supervised Adaptable 3D CNN.

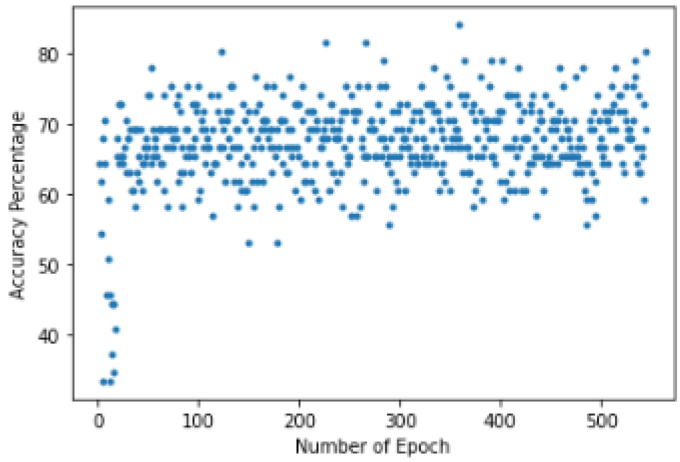
```python
def conv3d(x, W):
    conv = tf.nn.conv3d(x, W, strides=[1,1,1,1,1], padding='SAME')
    conv = tf.nn.dropout(conv, 0.5)
    return conv

def maxpool3d(x):
    #                        size of window         movement of window as you slide about
    return tf.nn.max_pool3d(x, ksize=[1,2,2,2,1], strides=[1,2,2,2,1], padding='SAME')
```

```python
def convolutional_neural_network(x):
    #                # 5 x 5 x 5 patches, 1 channel, 32 features to compute.
    weights = {'W_conv1':tf.Variable(tf.random_normal([3,3,3,1,32])),
               #       5 x 5 x 5 patches, 32 channels, 64 features to compute.
               'W_conv2':tf.Variable(tf.random_normal([3,3,3,32,64])),
               #                                  64 features
               'W_fc':tf.Variable(tf.random_normal([248768,1024])),
               'out':tf.Variable(tf.random_normal([1024, n_classes]))}

    biases = {'b_conv1':tf.Variable(tf.random_normal([32])),
               'b_conv2':tf.Variable(tf.random_normal([64])),
               'b_fc':tf.Variable(tf.random_normal([1024])),
               'out':tf.Variable(tf.random_normal([n_classes]))}

    #                            image X      image Y        image Z
    x = tf.reshape(x, shape=[-1, IMG_SIZE_PX_X, IMG_SIZE_PX_Y, SLICE_COUNT, 1])

    conv1 = tf.nn.relu(conv3d(x, weights['W_conv1']) + biases['b_conv1'])
    conv1 = maxpool3d(conv1)


    conv2 = tf.nn.relu(conv3d(conv1, weights['W_conv2']) + biases['b_conv2'])
    conv2 = maxpool3d(conv2)

    fc = tf.reshape(conv2,[-1, 248768])
    fc = tf.nn.relu(tf.matmul(fc, weights['W_fc'])+biases['b_fc'])
    fc = tf.nn.dropout(fc, keep_rate)

    output = tf.matmul(fc, weights['out'])+biases['out']

    return output
```

```python
from sklearn.model_selection import train_test_split

def train_neural_network(x):
    prediction = convolutional_neural_network(x)
    cost = tf.reduce_mean( tf.nn.softmax_cross_entropy_with_logits(logits=prediction, labels=y) )
    optimizer = tf.train.AdamOptimizer(learning_rate=1e-3).minimize(cost)

    file = open("output.txt", "w");

    hm_epochs = 1000
    with tf.Session() as sess:
        sess.run(tf.initialize_all_variables())

        successful_runs = 0
        total_runs = 0

        for epoch in range(hm_epochs):
            epoch_loss = 0

            train_data, validation_data = train_test_split(much_data, train_size=0.8)

            for data in train_data:
                total_runs += 1
                try:
                    X = data[0]
                    Y = data[1]
                    _, c = sess.run([optimizer, cost], feed_dict={x: X, y: Y})
                    epoch_loss += c
                    successful_runs += 1
                except Exception as e:
                    pass
                    #print(str(e))
```

```python
            print('Epoch', epoch+1, 'completed out of',hm_epochs,'loss:',epoch_loss)
            file.write('Epoch'+ str(epoch+1)+ 'completed out of'+str(hm_epochs)+'loss:'+str(epoch_loss))

            correct = tf.equal(tf.argmax(prediction, 1), tf.argmax(y, 1))
            accuracy = tf.reduce_mean(tf.cast(correct, 'float'))

            print('Accuracy:',accuracy.eval({x:[i[0] for i in validation_data], y:[i[1] for i in validation_data]}))
            file.write('Accuracy:',accuracy.eval({x:[i[0] for i in validation_data], y:[i[1] for i in validation_data]}))

        print('Done. Finishing accuracy:')
        print('Accuracy:',accuracy.eval({x:[i[0] for i in validation_data], y:[i[1] for i in validation_data]}))

        print('fitment percent:',successful_runs/total_runs)

        file.write('Done. Finishing accuracy:')
        file.write('Accuracy:',accuracy.eval({x:[i[0] for i in validation_data], y:[i[1] for i in validation_data]}))

        file.write('fitment percent:',successful_runs/total_runs)
```

```
import matplotlib.pyplot as plt

plt.plot(x_epoch,accuracy,'.')
#plt.axis([0, 600, 0, 80000])
plt.ylabel("Accuracy Percentage")
plt.xlabel("Number of Epoch")
plt.show()
```



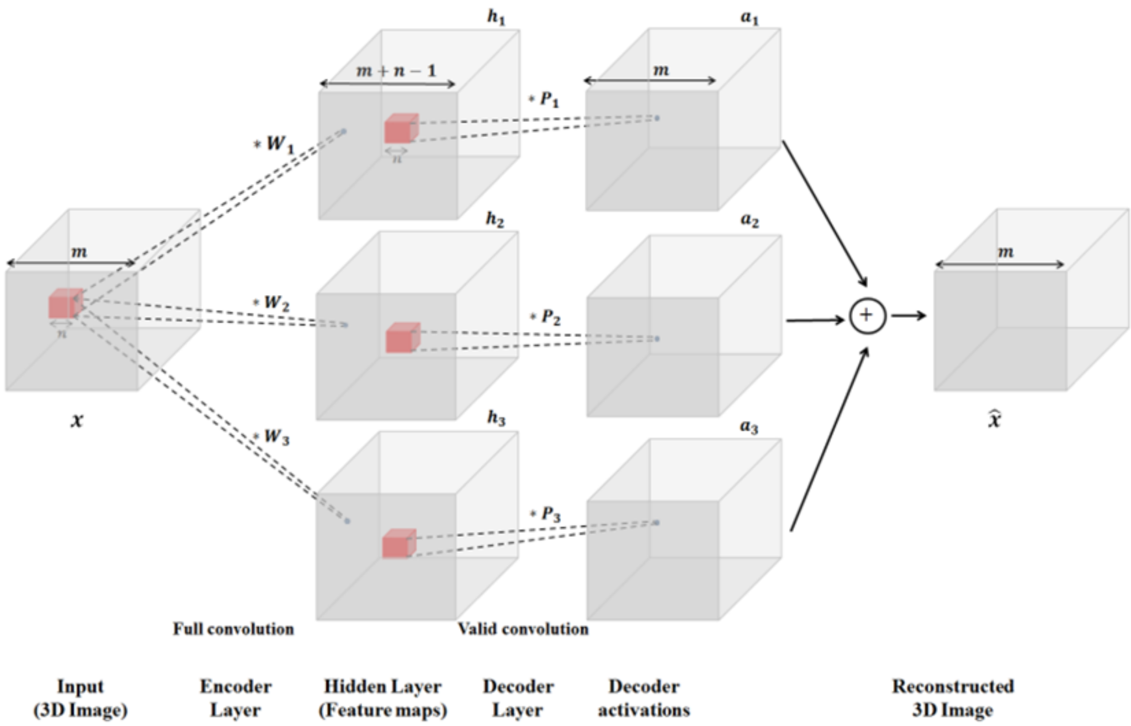


Fig 8.22 Detailed representation of all the layers involved in 3D CNN

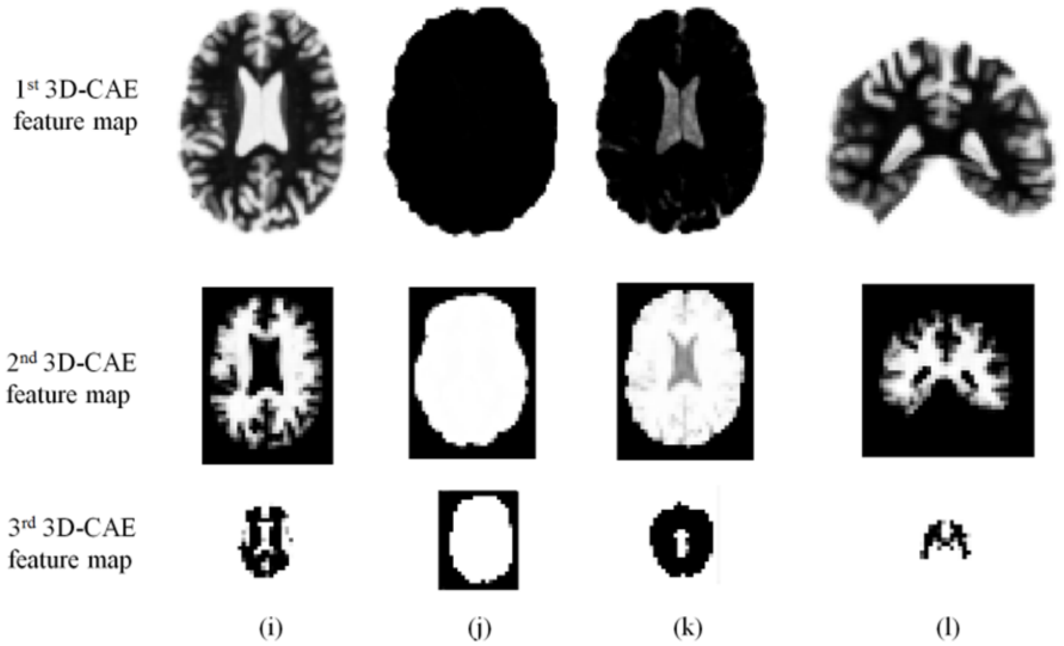
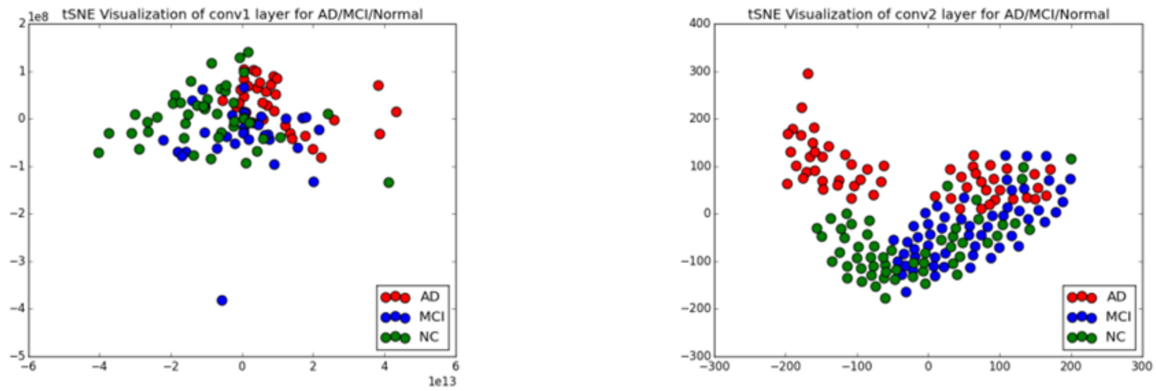


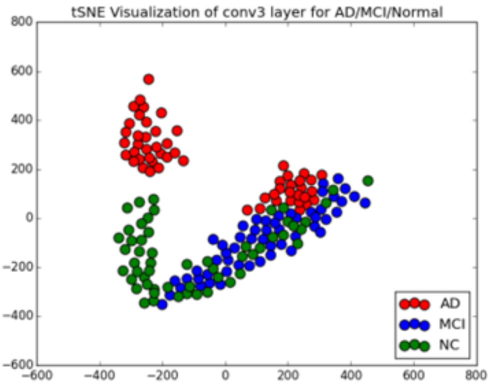
Fig 8.23 Feature maps of various 3D convolutional autoencoders.



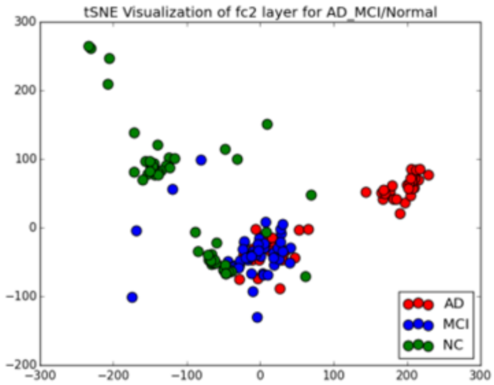
(a) conv1 hidden

(b) conv2 hidden

Fig 8.24 Visualisation of the first two hidden layers (convoluted layer1 and convoluted layer2) for AD.

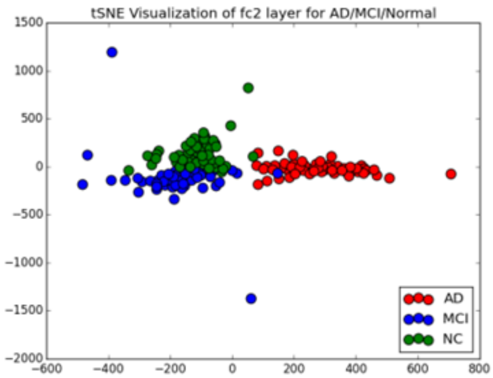


(c) conv3 hidden

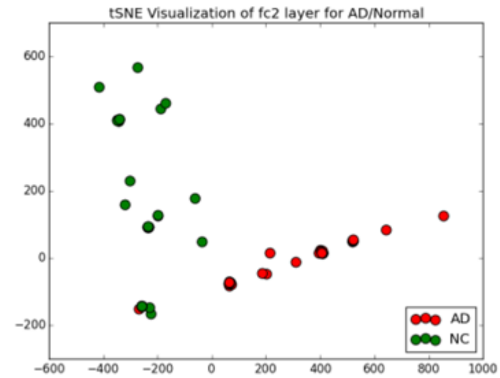


(d) fc2 hidden - AD/MCI/NC

Fig 8.25 Visualisation of the hidden convoluted layer3 and full convolution (fully convoluted layer2) layers for AD.

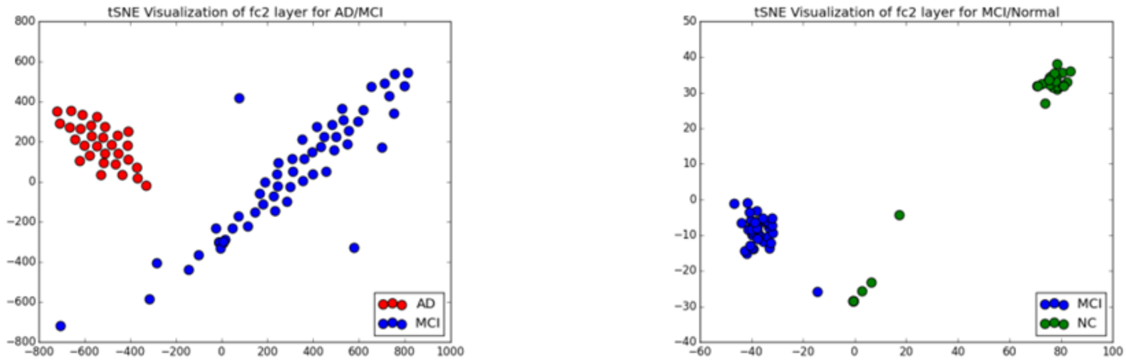


(e) fc2 hidden - AD+MCI/NC



(f) fc2 hidden - AD/NC

Fig 8.26 Visualisation of fully convoluted layer2 hidden layers for different purposes.



(g) fc2 hidden - AD/MCI

(h) fc2 hidden - MCI/NC

Fig 8.27 tSNE Visualisation of fully convoluted layer2 hidden layers for different purposes.

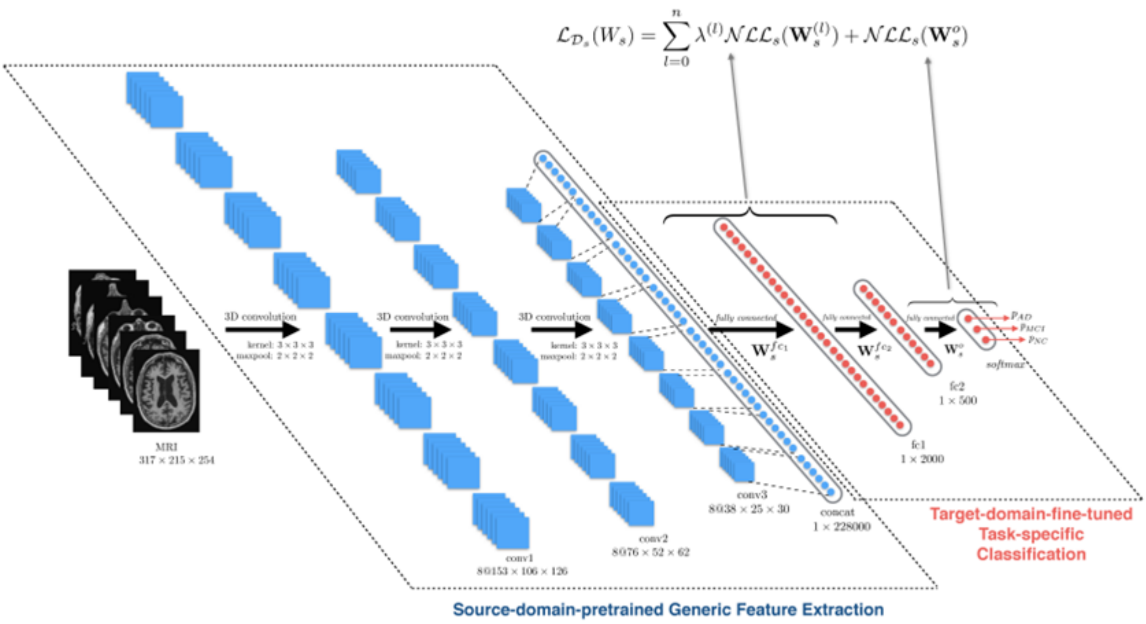


Fig 8.28 Architecture of the 3D CNN classification of MRI images for reference.

8.2.3 NLP

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written - referred to as natural language. It is a component of artificial intelligence (AI). It is a discipline that focuses on the interaction between data science and human language and is scaling to lots of industries.

Language variation can act as a proxy that monitors how a patient's cognitive functions have been affected (e.g., issues with word finding and impaired reasoning). The machine learning models can exploit these language patterns and thus help in diagnosing the patient.

Code for LSTM (Long Short Term Memory, which is a recurrent neural network technique) is done by comparing it with Particle Swarm Optimisation Methodology. Long Short Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It is capable of handling the vanishing gradient problem faced by RNN. A recurrent neural network, also known as RNN, is used for persistent memory.

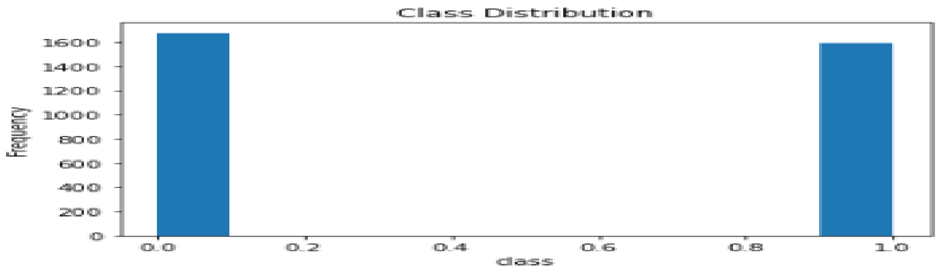
CNN: For each sentence, we apply an embedding and a convolutional layer, followed by a max pooling layer. The convolution features are obtained by applying filters of varying window sizes to each window of words.The result is then passed to a softmax layer that outputs probabilities over two classes.

LSTM-RNN: CNNs are not specialised for capturing long-range sequential correlations. We thus also experimented with an LSTM-RNN model, which consists of an embedding layer followed by an LSTM layer. The final state, containing information from the entire sentence, is fed to a fully-connected layer followed by a softmax layer to obtain the output probabilities.

CNN-LSTM: Observing that both models achieve results comparable to previous best performing approach, and considering that they each have their own complementary strengths, we experimented with a combined architecture, laying an LSTM layer on top of CNN. This CNN layer is identical to the vanilla CNN before the max-pooling layer, and the LSTM layer is identical to the vanilla LSTM-RNN after the embedding layer.

```
counts, bins = np.histogram(df['AD'])
plt.hist(bins[:-1], bins, weights=counts)
plt.title("Class Distribution")
plt.xlabel('class')
plt.ylabel('Frequency')
print("Class 0 samples: ",len(df[df['AD']==0]))
print("Class 1 samples: ",len(df[df['AD']==1]))
plt.show()
```

```
Class 0 samples:  1676
Class 1 samples:  1596
```



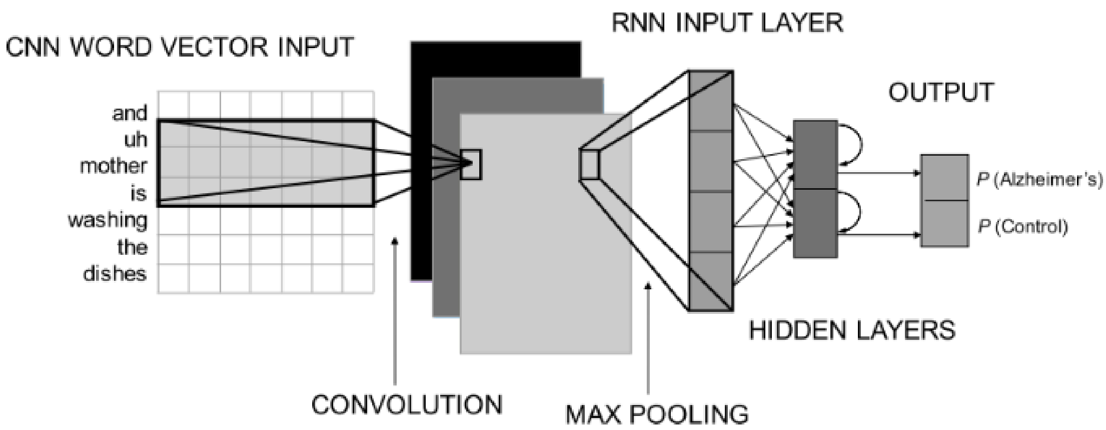


Fig 8.29 Architecture of Natural Language Processor and its working

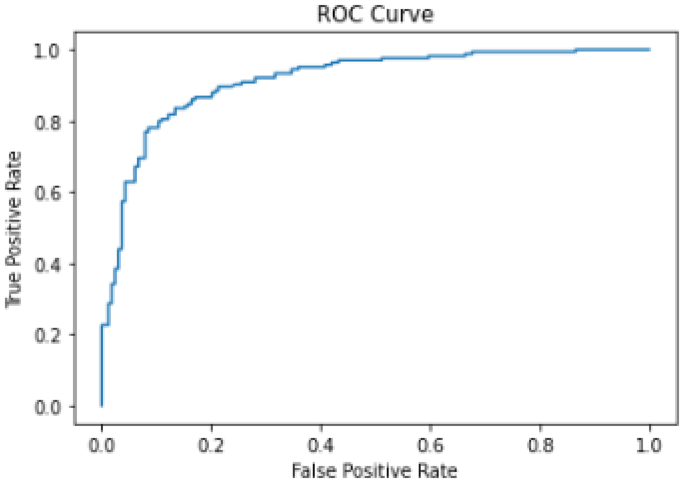
Model	Details	Accuracy
2D-CNN	Non-Tagged Utterances	82.8
LSTM	Non-Tagged Utterances	83.7

Fig 8.30 Table denoting the models and their respective accuracies

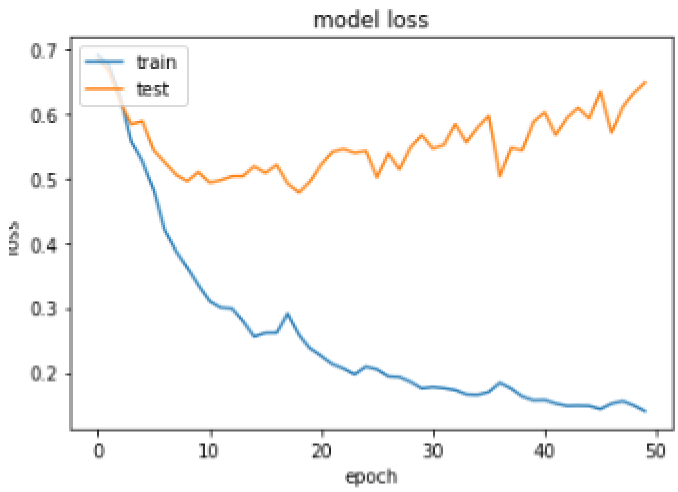
```
#evaluating model on test data

eval_list = model.evaluate(Xwords_test,y_seq_test)
print('Test Loss: ',eval_list[0])
print('Test Accuracy: ',eval_list[1])
```

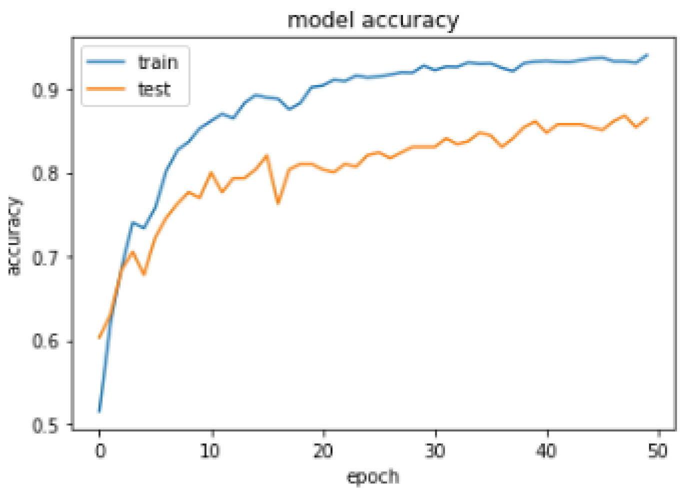
```
328/328 [==============================] - 1s 4ms/step
Test Loss:  0.5132803451724168
Test Accuracy:  0.8414633870124817
```

```
y_pred_probas = model.predict(Xwords_test)
y_pred_probas = [i[0] for i in y_pred_probas]
fpr, tpr, thresholds = roc_curve(y_seq_test, y_pred_probas)
#roc_auc = roc_auc_score(y_test, scores)
plt.plot(fpr, tpr)

plt.title("ROC Curve")

plt.xlabel("False Positive Rate")

plt.ylabel("True Positive Rate")

plt.show()
```




```
[[137  27]
 [ 25 139]]
              precision    recall  f1-score   support

           0       0.85      0.84      0.84       164
           1       0.84      0.85      0.84       164

    accuracy                           0.84       328
   macro avg       0.84      0.84      0.84       328
weighted avg       0.84      0.84      0.84       328
```





8.3 Usable features based on the severity of the demented patient

Some of the usable features that can be quite helpful for a demented patient or in the analysis of a potential patient are - Alarms and Reminders, Puzzles with personalised photographs, To-Do Lists, and Cognitive Questions and Games.

8.3.1 Alarms and Reminders

Many people with dementia need a bit of assistance with daily activities like reminding a task, taking medication, or doing an activity. To overcome this alarms and reminders will be very helpful for the demented people. A Python application can be used which sends mails and triggers alarms based on the reminders that are set by the patient or for the patient.

8.3.2 Puzzles with personalised photographs

Puzzles for dementia patients have been widely helpful. This is because brain-training exercises have been associated with reducing the risk of developing dementia. Mentally challenging activities such as doing a jigsaw puzzle stimulates thinking and memory, which increases feelings of well-being and helps improve communication and interaction. They can involve activities like jigsaw puzzles. The patient or the caretaker can upload their personalised photos that are divided into grid cells and jumbled. These cells must be rearranged by the patient to obtain the original image. In this process, the number of moves, the time taken to solve the puzzle and the difficulty level are taken into consideration while further evaluation with respect to the syndrome.

Fig 8.31 Figure representing Jigsaw puzzle yet to be solved

In addition to these, the grid sizes can be altered and the patient can make use of numbers that are allotted to each grid cell so that it aids them in solving the puzzle easily.

Fig 8.32 Figure representing cells with numbers

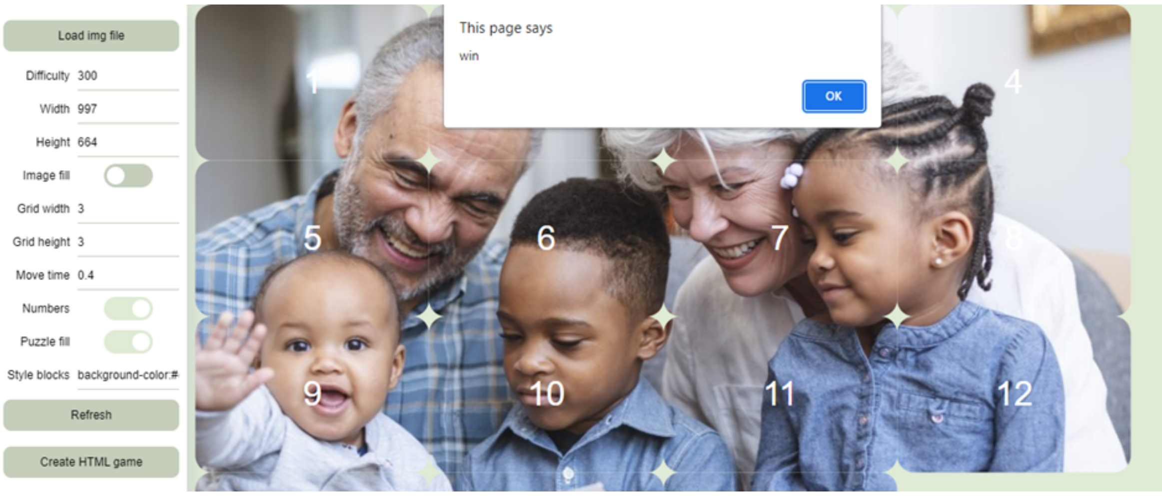


Fig 8.33 Figure representing jigsaw puzzle solved

8.3.3 To-Do Lists

People with dementia tend to forget their daily schedules, let alone things to be accomplished. If this problem should be solved, they must be constantly reminded of the list of things to be done. This objective is achieved by a website called Todoist. Todoist helps in making notes and personalised to-do lists so that patients do not miss a thing. Additionally, their memory power can also improve by regularly practising this kind of activity.

To-do lists can include API calls over - Access Tokens, Resources, Project, Items, Tasks, Orders, Labels, Notes, and Reminders. These sections form the conglomerate services provided by it.

8.4 Detailed analysis of the data

The data obtained from the myriad features will be sent to Microsoft Excel and analysis is performed using PowerQuery functions. PowerQuery is a business intelligence tool that imports data from multiple sources and performs steps like cleaning, transforming and reshaping data based on one's needs. It enables you to set up a query and reuse it by performing a simple refresh. It is also known as a data preparation and data transformation engine. PowerQuery has a graphical interface for performing the above mentioned task. Steps like extract, transform and load (ETL) processing of data.

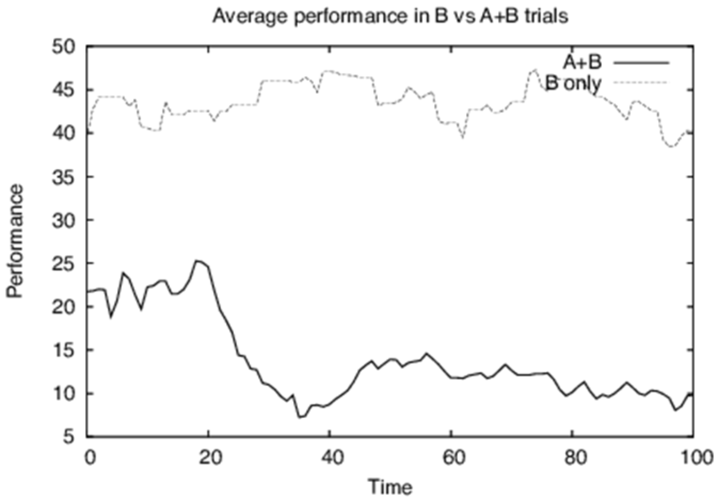


Fig 8.34 Sample graphical representation of the variation in Performance with Time

CHAPTER 9

TESTING

9.1 Introduction to Testing

Testing is the process of evaluating a system or its component(s) with the intent to find out whether it satisfies the specified requirements or not. This activity results in the actual, expected, and the difference between their results. In simple words, testing is executing a system to identify any gaps, errors, or missing requirements in contrast to the actual desire or requirements. Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design, and coding. It represents an interesting anomaly for the software. Testing is the process of detecting errors. It performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later during maintenance also. Generally, the testing phase involves the testing of the developed system using various test data. Preparation of the test data plays a vital role in system testing. After preparing the test data the system under study was tested using those test data. While testing the system, errors were found and corrected by using the testing steps and corrections are also noted for future use. Thus, a series of testing is performed for the proposed system before the system is ready for implementation. Thus, the aim of testing is to demonstrate that a program works by showing that it has no errors. The fundamental purpose of the testing phase is to detect the errors that may be present in the program. Thus, testing allows developers to deliver software that meets expectations, prevents unexpected results, and improves the long-term maintenance of the application. Depending upon the purpose of testing and the software requirements, the appropriate methodologies are applied. Wherever possible, testing can also be automated.

9.2 Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, altogether, Testing is a process of executing a program with the intent of finding an error. A successful test is one that uncovers a yet undiscovered error. A good test case is one that has a high probability of finding an error if it exists. The software more or less confirms the quality and reliability standards.

9.3 Types of Testing

Testing is categorised into many types based on the requirement that is to be tested. If any modular functionality needs to be tested then we are required to opt for unit testing. If the requirement is to test the integration of the components, then an integration test would be an option.

Major Classification of testing can be done based on the way they are being done:

- Manual Testing
- Automated Testing

Based on the situation or requirement-based testing it can be classified as:

- Unit testing
- Integrating Testing
- Regression Testing
- Alpha Testing
- Beta Testing

9.3.1 Manual Testing

Manual Testing is a software testing methodology in which test cases are executed manually without any usage from the automated process. These test cases are executed by the testers according to the end-user perspective. The test cases cover all the edge cases to ensure the proper functionality of the product.

Manual Testing is considered the important and compulsory phase for any newly developed software. This phase requires more effort and time to complete but produces bug-free products to a greater extent. Types of Manual Testing:

• White Box Testing
• Black Box Testing
• Grey Box Testing

Manual testing was performed to check if the Dementia application was working as expected or not.

9.3.2 Automated testing

Automated Testing is a process that checks if the software is functioning as per the requirement before it is released into the production environment. This process of software testing uses predefined or pre-coded scripts that are executed in a sequence by the testing tools. This flow helps in making the testing automated. This is more widely implemented in a modification stage product. This testing sets the goal of reducing the testing efforts, and delivering the products at a faster rate and more affordably.

Types of Automation Testing:
- Unit testing
- Integration Testing
- Performance Testing
- Acceptance Testing

9.4 Software Testing Methods

Software testing methodology is defined as the testing methods and types used to ensure that the software under test meets the expectations of the client. Test methods include active and inactive tests to verify the functions of the software. Examples of test methods, unit testing, integration testing, system testing, performance testing, etc. Each test technique has its own policy, strategy and outcome.

9.4.1 Black Box Testing

Black Box testing is a software testing methodology in which the functional properties of the software product are tested without having any knowledge of the internal code structure, internal paths, and flows. Black box testing mainly focuses on the input flow and output flow. It is also called behavioural testing.

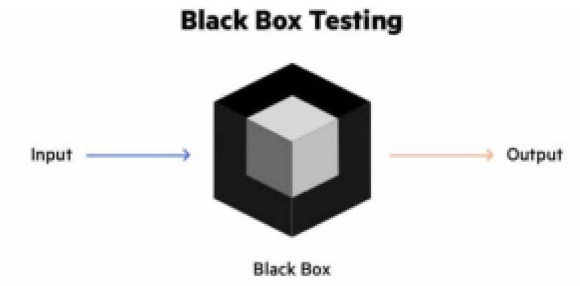


Fig 9.1 Diagrammatic representation of Black Box Testing

Types of Black Box Testing:

Functional Testing: This test includes the testing process of the functional requirements that are involved in the software.

Non- Functional Testing: This test includes the testing of the non-functional requirements like performance, and scalability.

Regression Testing: Regression Testing includes the testing after code fixes to ensure if the code fixes affected the regular implementation

The black box test focuses on checking the performance of software without checking the internal structure or code. The main source of black-box testing is how the customer requirements are specified. A method in which the inspector selects a function, assigns an input value, tests its performance, and checks whether the function produces the expected result. If the function returns the correct result, it is passed to the test, otherwise, it fails. The evaluation team reports the results to the development team and evaluates the next action. After all, work has been evaluated, if there are serious problems, we will return them to the development team for repair.

9.4.2 Grey Box Testing

Grey-box testing is a combination of software testing that combines black-box testing strategies and white-box testing methods. In black-box testing methods, the tester does not know the internal structure of the object being tested. In white-box testing the internal code structure is provided beforehand to the tester . The internal structure is partially known to the testers from grey box experiments. This includes access to internal data structures and algorithms for developing test cases. The software system looks like an invisible grey box, hence the name grey box test. It focuses on context errors that are usually related to web systems.

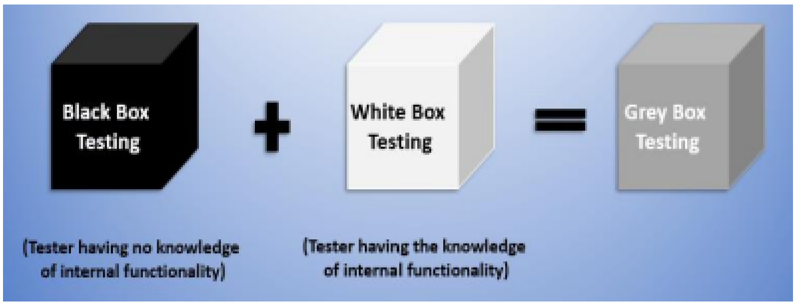


Fig 9.2 Representation of Grey Box Testing

Grey Box Test Tip:

Matrix Test: The matrix test strategy assesses the business and technical risks identified by software application developers. Engineers account for all variables in the system. Each of these variables has unique technical and business risks associated with them and can be used at different phases throughout its lifecycle.

Pattern Test: Performs a test by analysing past errors. View the code to determine the cause of the failure. The analysis included template covers the reasons for the error. This aids the design conditions as they are busy detecting some failures before production.

Orthogonal Array Test: It is a way to check the black box. In an orthogonal array test, the test data has no allowances and combinations. Orthogonal array testing is chosen when higher installation is required and if there are very few test cases to cover in testing phase and the test data is large. This is very useful when testing complex applications.

Retest: After every change to the software, the software is retested to ensure that changes or new features do not affect existing system performance. Backtesting is also performed to ensure that the defect does not affect the performance of other software.

9.4.3 White Box Testing

White box testing methods that analyse internal structure data structures used, internal organisation, code format, and software functionality can work just like black-box testing. Also known as structural testing, clear box testing, or glass box testing.

White Box Test Procedure: Input Data: Requirements, Performance Specifications, Design Documentation, Source Code.

Analysis: Perform risk based analysis to guide the entire process.

Proper test plan: Design test cases to cover all code. Perform cleaning until error-free software is available. Results are also welcome.

Results: Preparation of a final report for the entire testing process.

Test Strategy:

Enter Statements: The goal of this process is to reduce the entire statement at least once. So, it checks each line of code. On the block diagram, an entire node must be moved at least once. All lines of code are included to help you identify the error code.

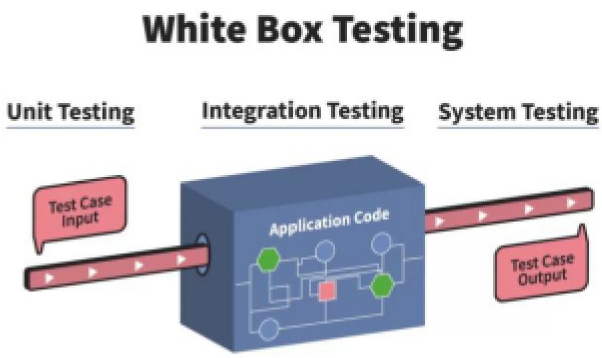


Fig 9.3 Test Strategy of White Box Testing

9.5 Testing Levels

9.5.1 Non-functional Testing

The Testing of non-functional attributes is the non-functional testing. In this testing, the program has non-functional specifications like user interface, stability, performance, etc.

9.5.1.1 Performance Testing

Performance Testing is a software evaluation process used to assess speed, response time, stability, reliability, measurement, and software system performance under a specific function. The main purpose of performance appraisal is to identify and eliminate barriers to application performance. It is the subset of operational engineering and is also known as the “Complete Test”.

The focus of performance testing is to test the software system.

Speed - Determines whether an app responds quickly

Scalability - Determines the maximum load an application can carry.

Stability - determines whether an application is stable under various loads

Many users have tried to access the site and check what the recovery speed was.

9.5.1.2 Stress Testing

Pressure testing is the process of determining the power of a computer, network, system, or device to maintain a certain level of performance under adverse conditions. Procedures may include laboratory volume testing, such as measuring error rates or system failure rates.

9.5.1.3 Security Testing

Security testing is a form of software testing that detects threats from software programs and prevents suspicious attacks by attackers. The purpose of security checks is to identify potential risks and vulnerabilities to software systems that could lead to loss of information, and revenue, whether by employees or outside the organisation.

9.5.1.4 Portability Testing

The process of determining whether software can be easily moved from one environment to another. Some of its properties are coexistence and adaptability. These tests help identify defects that were not discovered during integration or unit testing. Dementia applications have been tested in various environments such as LINUX, MAC, and WINDOWS to ensure that they work as expected.

9.5.1.5 Usability Testing

It is a technique that is used to identify usability defects. This is done by making a small set of target end-users use the software and this testing focuses on the ease of use of the application and the flexibility of the application. The application was initially given to a set of 3 people to perform the usability testing. They were asked to check if the application was easy to use and clear.

9.5.2 Functional Testing

This testing is used to validate the software system against the specifications. In this testing, each function of the software is tested, and the output is verified against the functional requirements. It mainly contains black box testing and checks if the errors are displayed according to the error conditions.

To perform the functional testing firstly we need to understand the functional requirements of the software and then compute the results using input or test data then after executing the test cases we compare the actual and the computed results. While performing the functional testing it is ensured that the Dementia application is behaving as it is expected. It is displaying the predictions and analysis accurately on the UI.

9.5.2.1 Integration Testing

In this test, all software modules are integrated and tested as a single team. Its main purpose is to identify errors in the interaction between these modules when they are combined. There are several modules in an application. The Dementia application consists of a UI module and a machine learning module. Initially, the UI module was tested separately and the ML model was tested separately in Unit testing. In the integration testing, both these modules were combined and the application as a whole was tested.

9.5.2.2 Regression Testing

Reversal test is the selection of all or part of the test cases performed, as well as re-use to ensure that existing functions are operating normally. These tests are performed to ensure that changes to the new code will not have a second effect on existing operations. Make sure the old code continues to work after making recent changes to the code. Several changes were done during the process of the development of the Dementia application. So, those changes must be tested.

9.5.2.3 Unit Testing

Unit testing is generally done in order to test accuracy of the isolated code. The white box testing is used as a part of unit testing. It’s often used to check if there are any issues in the code by the developer himself and the main aim of this testing is to analyse and fix those defects. Unit tests for Dementia applications were written in python by using various ways like mocking, assertions, etc.

9.5.2.4 Alpha Testing

It is used to detect if there are any bugs before releasing the software to the public. It is used to discover the bugs which were not discovered during previous tests. It is called alpha testing because it is done when software is developed. The development team has ensured that the Dementia detection Application is tested well after its completion. It is ensured that there are no bugs present before rendering it to the public users.

9.5.2.5 Beta Testing

Beta testing tests user acceptance when a product team distributes a virtual product in a set of targeted users to test how well it performs in the real world. A dementia disorder application was distributed among others to test it according to their will and report in case of any bugs.

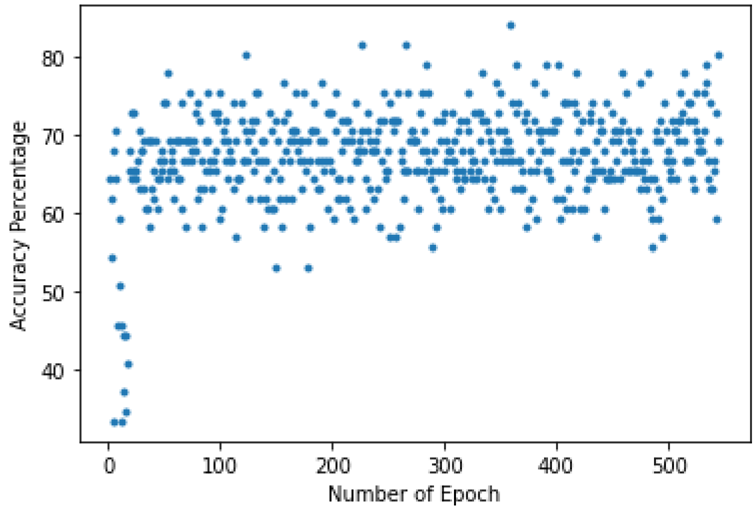
9.6 Test Cases

CASE #No	TESTCASE DESCRIPTION	RESULT
01	If all the input data fields are filled and proper data is given	PASS
02	If no input data is given and the fields are left empty	FAIL
03	If partial data fields are filled.	FAIL
04	When user tries to give string input data in A1-A10 score fields	INVALID
05	When the numeric fields are any random number instead of 0/1	INVALID
06	When string input is given under “Ethnicity”, “Country” and “Relation”	PASS

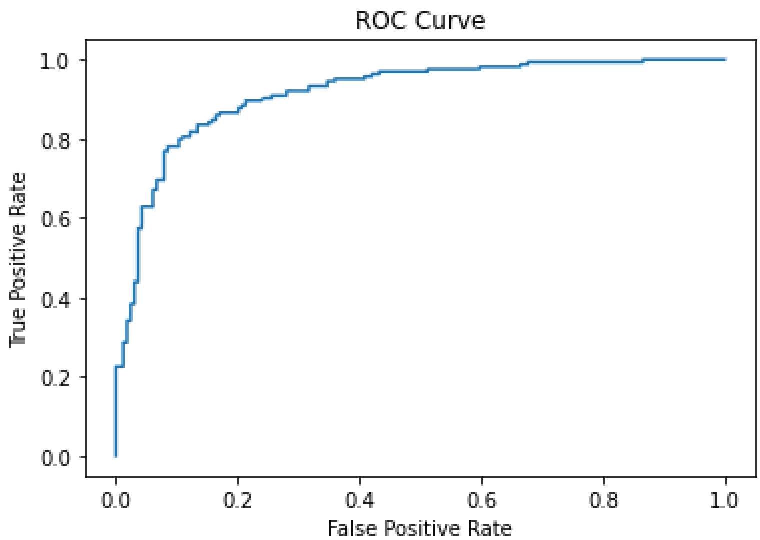
Fig 9.4 Table representing different results based on various test cases

CHAPTER 10

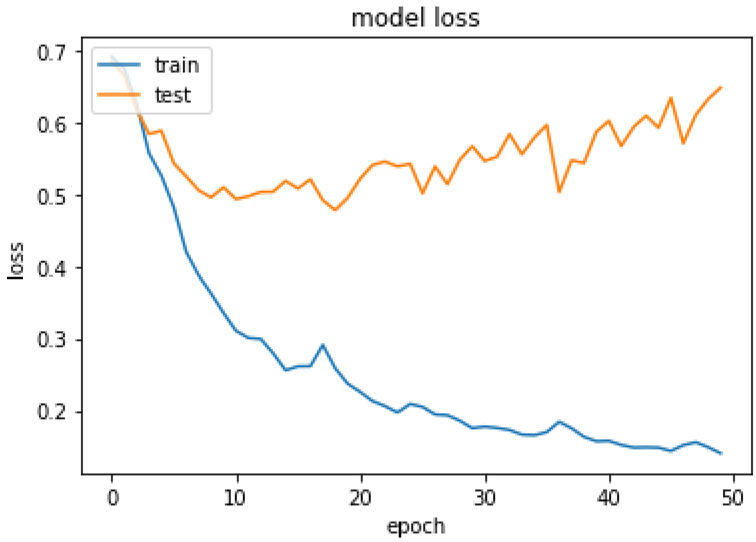
OUTPUT RESULTS AND SCREENSHOTS

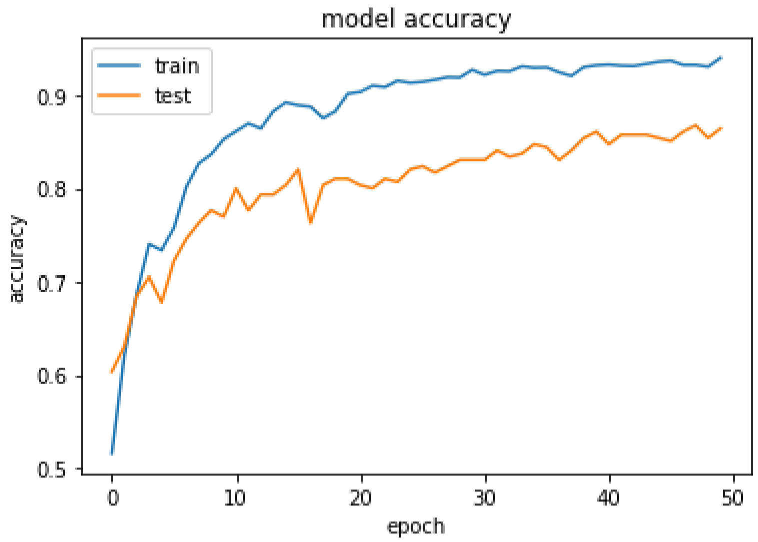


3D CNN

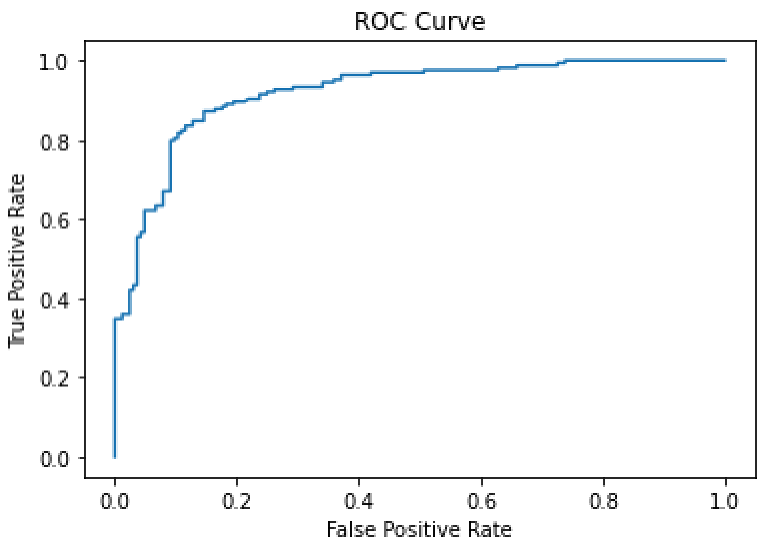


LSTM

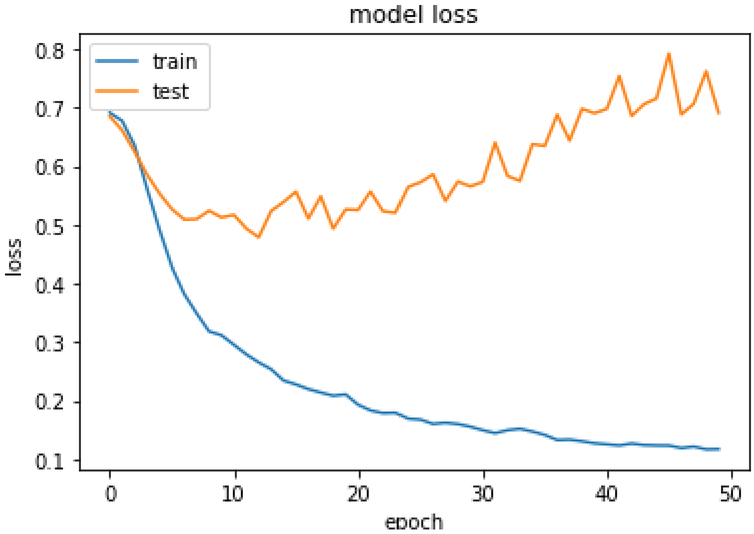


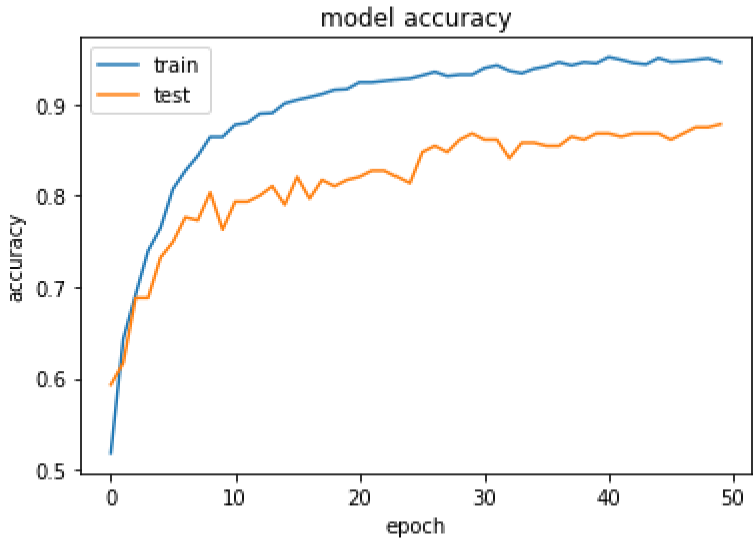


MODEL LOSS AND ACCURACY FOR LSTM

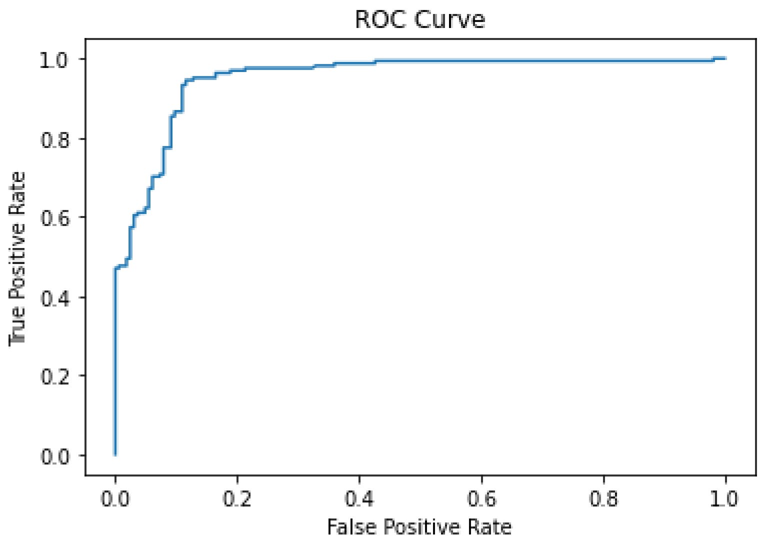


BIDIRECTIONAL LSTM

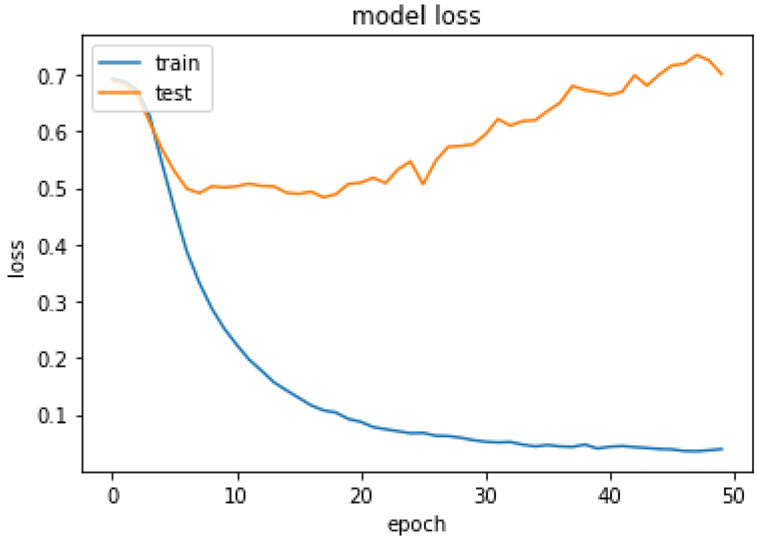


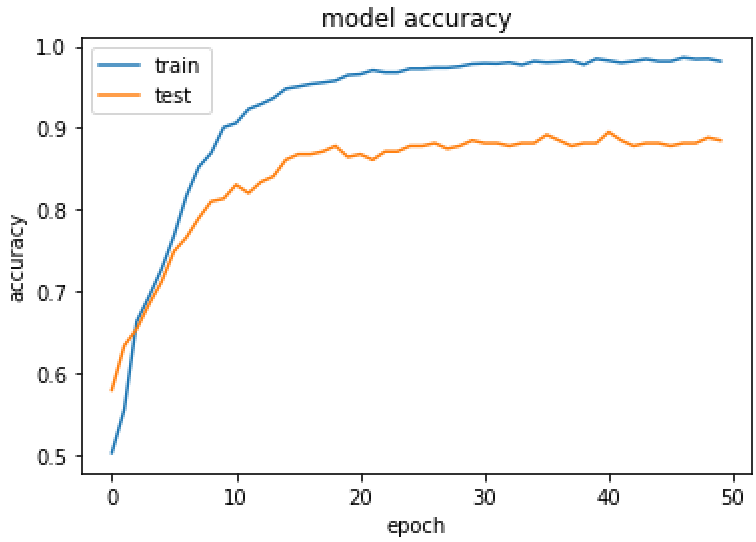


MODEL LOSS AND ACCURACY FOR BIDIRECTIONAL LSTM



CNN LSTM


model loss
train
test
0.7
0.6
0.5
0.4
0.3
0.2
0.1
loss
0
10
20
30
40
50
epoch


model accuracy
train
test
1.0
0.9
0.8
0.7
0.6
0.5
accuracy
0
10
20
30
40
50
epoch

CHAPTER 11
CONCLUSION AND FUTURE SCOPE

11.1 Conclusion

Dementia cannot be cured as a syndrome that affects memory, but its symptoms can be mitigated. In order to act upon this syndrome, dementia needs to be identified effectively, this is done at 84.82% accuracy using an Extra Tree Classifier specified in the proposed system. This was the accuracy obtained after facing many challenges like complex neural network image analysis and unavailability of data. Multiple classification methods helped us obtain the best accuracy, result and gave a clear idea on which classification method is beneficial with such inadequate data and model.

MRI images obtained from medical tests needed to be analysed and further explored excessively using image processing. The mini-mental state examination data can be further used to accurately classify dementia. Furthermore, the available and acquired data can be used to treat patients in a more effective way focusing on deprived areas. In addition to this, patterns between the data can be identified and understood in order to prevent the hereditary passage of the syndrome, to a certain extent.

11.2 Future Scope

In the final module of implementation of the project, i.e., Data Analysis, a greater number of applications can be included. Moreover, processed workflow and data can help in producing more sophisticated reports with the help of relevant doctors. Presently, data in each record may not necessarily be obtained from a single person. The future scope of this project can include being able to capture the whole data pertaining to an individual patient in order to provide better prediction.

BIBLIOGRAPHY

References

[1] Nils D. Forkert, Sascha Gill, Sophie Hu, Zahinoor Ismail, Frank P. MacMaster, Pauline Mouches, Deepthi Rajashekar, Eric E. Smith ,"Using Machine Learning to Predict Dementia from Neuropsychiatric Symptom and Neuroimaging Data" Alzheimer's Disease Neuroimaging vol. 75, no. 1, pp. 277-288, 2020.

[2] Fubao Zhu, Xiaonan Li, Haipeng Tang, Zhuo He, Chaoyang Zhang, Guang-Uei Hung, Pai-Yi Chiu, Weihua Zhou, "Machine Learning for the Preliminary Diagnosis of Dementia", Scientific Programming, vol. 2020, Article ID 5629090, 10 pages, 2020.

[3] Mathkunti, N.M., Rangaswamy, S. "Machine Learning Techniques to Identify Dementia". SN COMPUT. SCI. 1, 118 (2020).

[4] Harshit Parmar, Brian Nutter, Rodney Long, Sameer Antani, Sunanda Mitra ,"Spatiotemporal feature extraction and classification of Alzheimer's disease using deep learning 3D-CNN for fMRI data",J. of Medical Imaging, 7(5), 056001 (2020).

[5] Gloria Castellazzi, Maria Giovanna Cuzzoni, Matteo Cotta Ramusino,Daniele Martinelli, Federica Denaro, Antonio Ricciardi,Paolo Vitali,Nicoletta Anzalone,Sara Bernini,Fulvia Palesi,Elena Sinforiani, Alfredo Costa,Giuseppe Micieli,Egidio D'Angelo,Giovanni Magenes,Claudia A. M. Gandini Wheeler Kingshott, "A Machine Learning Approach for the Differential Diagnosis of Alzheimer and Vascular Dementia Fed by MRI Selected Features" ,Front. Neuroinform., 11 June 2020.

[6] Mathotaarachchi S, Pascoal TA, Shin M, Benedet AL, Kang MS, Beaudry T, Fonov VS, Gauthier S, RosaNeto P, Alzheimer's Disease Neuroimaging Initiative "Identifying incipient dementia individuals using machine learning and amyloid imaging" Neurobiol Aging. 2017 Nov;59:80-90. Epub 2017 Jul 11.

[7] S. Nirjon, I. A. Emi, M. A. S. Mondol, A. Salekin, and J. A. Stankovic, "MOBI-COG: a mobile application for instant screening of dementia using the mini-cog test," in Proceedings of the Wireless Health 2014 on National Institutes of Health, pp. 1–7, October 2014.

[8] A. P.-H. Kong, "Conducting cognitive exercises for early dementia with the use of apps on iPads," Communication Disorders Quarterly, vol. 36, no. 2, pp. 102–106, 2014.

[9] D. Welsh, K. Morrissey, S. Foley et al., "Ticket to talk: Supporting conversation between young people and people with dementia through digital media," in Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI '18, p. 375, April 2018.

[10] D. Shibata, M. Miyabe, S. Wakamiya, A. Kinoshita, K. Ito, and E. Aramaki, "VocabChecker: measuring language abilities for detecting early stage dementia," in Proceedings of the 23rd International

Conference on Intelligent User Interfaces Companion, p. 24, March 2018.

[11] M. Atee, K. Hoti, and J. D. Hughes, “A technical note on the PainChek™ system: a web portal and mobile medical device for assessing pain in people with dementia,” Frontiers in Ageing Neuroscience, vol. 10, article 117, 2018.

[12] H. Megges, S. D. Freiesleben, N. Jankowski, B. Haas, and O. Peters, “Technology for home 5 dementia care: A prototype locating system put to the test,” Alzheimer's and Dementia: Translational Research and Clinical Interventions, vol. 3, no. 3, pp. 332–338, 2017.

[13] G. Zorluoglu, M. E. Kamasak, L. Tavacioglu, and P. O. Ozanar, “A mobile application for cognitive screening of dementia,” Computer Methods and Programs in Biomedicine, vol. 118, no. 2, pp. 252–262, 2015.

[14] S. Sindi, E. Calov, J. Fokkens et al., “The CAIDE Dementia Risk Score App: the development of an evidence-based mobile application to predict the risk of dementia,” Alzheimer's and Dementia: Diagnosis, Assessment and Disease Monitoring, vol. 1, no. 3, pp. 328–333, 2015.

[15] J. R. Thorpe, K. V. H. Rønn-Andersen, P. Bień, A. G. Özkil, B. H. Forchhammer, and A. M. Maier, “Pervasive assistive technology for people with dementia: a UCD case,” Healthcare Technology Letters, vol. 3, no. 4, pp. 297–302, 2016.

[16] I. V. Vahia, R. Kamat, C. Vang et al., “Use of tablet devices in the management of agitation among inpatients with dementia: an open-label study,” The American Journal of Geriatric Psychiatry, 2016.

[17] A. Boyd, J. Synnott, C. Nugent, D. Elliott, and J. Kelly, “Community-based trials of mobile solutions for the detection and management of cognitive decline,” Healthcare Technology Letters, vol. 4, no. 3, pp. 93–96, 2017.

[18] N. Dethlefs, M. Milders, H. Cuayáhuitl, T. Al-Salkini, and L. Douglas, “A natural language-based presentation of cognitive stimulation to people with dementia in assistive technology: a pilot study,” Informatics for Health and Social Care, vol. 42, no. 4, pp. 349–360, 2017.

[19] Stamate D. et al. (2020) “Applying Deep Learning to Predicting Dementia and Mild Cognitive Impairment”. In: Maglogiannis I., Iliadis L., Pimenidis E. (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 584. Springer, Cham.

[20] M. Tanveer, B. Richhariya, R. U. Khan, A. H. Rashid, P. Khanna, M. Prasad, C. T. Lin, “Machine Learning Techniques for the Diagnosis of Alzheimer’s Disease: A Review”, ACM Transactions on

Multimedia Computing, Communications, and Applications, Volume 16, Issue 1s, Article No.: 30, pp 1–35.

[21] G. Battineni, N. Chintalapudi, F. Amenta, “Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM)”, Informatics in Medicine Unlocked, Volume 16, 2019, 100200.

[22] Jo T, Nho K and Saykin AJ (2019) “Deep Learning in Alzheimer's Disease: Diagnostic Classification and Prognostic Prediction Using Neuroimaging Data”. Front. Ageing Neurosci. 11:220. doi: 10.3389/fnagi.2019.00220.

[23] Vijay S. Nori, Christopher A. Hane, William H. Crown, Rhoda Au, William J. Burke, Darshak M. Sanghavi, Paul Bleicher, “Machine learning models to predict onset of dementia: A label learning approach”, Alzheimer's & Dementia: Translational Research & Clinical Interventions, Volume 5, 2019, Pages 918-925.

[24] H. Niu, I. Álvarez-Álvarez, F. Guillén-Grima, I. Aguinaga-Ontoso, “Prevalence and incidence of Alzheimer's disease in Europe: A meta-analysis”, Neurología (English Edition), Volume 32, Issue 8, October 2017, Pages 523-532.

[25] So, A.; Hooshyar, D.; Park, K.W.; Lim, H.S. “Early Diagnosis of Dementia from Clinical Data by Machine Learning Techniques”. Appl. Sci. 2017, 7, 651.

[26] Molinuevo JL, Gramunt N, Gispert JD, Fauria K, Esteller M, Minguillon C, Sánchez-Benavides G, Huesa G, Morán S, Dal-Ré R, Camí J. “The ALFA project: A research platform to identify early pathophysiological features of Alzheimer's disease” Alzheimers Dement (NY). 2016 Mar 3;2(2):82-92. doi: 10.1016/j.trci.2016.02.003.

[27] Bon Mi Koo, PhD, Lisa M Vizer, PhD, “Mobile Technology for Cognitive Assessment of Older Adults: A Scoping Review”, Innovation in Ageing, Volume 3, Issue 1, January 2019.