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)

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

February 2022

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

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. B.V. Kiranmayee	Dr. S. Nagini
Professor	Professor & HOD
Department of CSE	Department of CSE
VNR VJIET	VNR VJIET

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

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 1: INTRODUCTION

1.1 Need for the project 3
1.2 Employing various methods 5

CHAPTER 2: LITERATURE SURVEY 7

CHAPTER 3: EXISTING SYSTEM

3.1 Detection 21
3.2 Providing assistance 22

CHAPTER 4: PROPOSED SYSTEM

4.1 Objectives 23
4.2 Methodologies 23

CHAPTER 5: FEASIBILITY STUDY 29

CHAPTER 6: SYSTEM ANALYSIS AND ARCHITECTURE

6.1 Introduction 32
6.1.1 Document Purpose 32
6.1.2 Definitions 32

6.1.3 Requirement Analysis 33
6.2 System Architecture 35
6.3 Functional Requirements 36
6.4 System Analysis 37
6.5 Non Functional Requirements 38
6.5.1 Performance Requirements 38
6.5.2 Design Constraints 38
6.5.3 Software System Attributes 39
6.6 Software Requirement Specification 39
6.7 Software Requirements 41
6.8 Hardware Requirements 42

CHAPTER 7: SOFTWARE DESIGN

7.1 UML Diagram 43
7.1.1 Definition 44
7.1.2 UML is a Language 44
7.1.3 UML Specifying 44
7.1.4 UML Visualisation 45
7.1.5 UML Constructing 45
7.1.6 Building blocks of UML 45
7.1.6.1 Things 45
7.1.6.2 Relationships 45
7.1.6.3 UML Diagrams 45
7.1.6.3.1 Structural Diagrams 46
7.1.6.3.2 Behavioural Diagrams 46
7.2 UML Diagrams 46
7.2.1 Use Case Diagram 46
7.2.2 Sequence Diagram 50
7.2.3 Class Diagram 54

CHAPTER 8: IMPLEMENTATION

8.1 Data Preprocessing 58

8.2 Classification of the collected data 70

8.2.1 Random Forest Classifier 70

8.2.2 3D CNN Classifier 72

8.2.3 NLP 79

8.3 Usable features based on the severity of the demented patient 83

8.3.1 Alarms and Reminders 83

8.3.2 Puzzles with personalised photographs 84

8.3.3 To-Do Lists 85

8.4 Detailed analysis of the data 86

CHAPTER 9: TESTING

9.1 Introduction to Testing 87

9.2 Testing Objectives 87

9.3 Types of Testing

88

9.3.1 Manual Testing 88

9.3.2 Automated Testing 89

9.4 Software Testing Methods 89

9.4.1 Black Box Testing 89

9.4.2 Grey Box Testing 90

9.4.3 White Box Testing 91

9.5 Testing Levels 92

9.5.1 Non-functional Testing 92

9.5.1.1 Performance Testing 92

9.5.1.2 Stress Testing 93

9.5.1.3 Security Testing 93

9.5.1.4 Portability Testing 93

9.5.1.5 Usability Testing 93
9.5.2 Functional Testing 93
9.5.2.1 Integration Testing 94
9.5.2.2 Regression Testing 94
9.5.2.3 Unit Testing 94
9.5.2.4 Alpha Testing 94
9.5.2.5 Beta Testing 95
9.6 Test Cases 95

CHAPTER 10: OUTPUT RESULTS AND SCREENSHOTS 96

CHAPTER 11: CONCLUSION AND FUTURE SCOPE

11.1 Conclusion 100
11.2 Future Scope 100

BIBLIOGRAPHY 101

CHAPTER 1

INTRODUCTION

Dementia is a condition in which a person's cognitive functioning, such as thinking, remembering, and reasoning, has deteriorated to the point where it interferes with their day-to-day activities. Some dementia patients are unable to manage their emotions, and their personalities may shift.

Dementia's early symptoms start with memory related issues such as short-term memory shifts and memory loss. One should also be aware that "*not every person suffering from memory loss is meant to have dementia, memory loss is just an early sign*".

During the initial days of Dementia, one's functioning is affected mildly which might cause a little less havoc to the daily living and fewer potential risks, but the later days might just turn adverse. The severity causes them to completely depend on others to cope with the everyday essential activities.

Dementia is a syndrome frequently seen in the aged citizens i.e. people ageing 85 and above. Though we see many senior citizens with this syndrome, it is not just related to the normal part of ageing itself. Only 1/3rd of people belonging to the age group of 85+ suffer with any form of dementia. People living beyond the age of 90s without showing any signs of dementia are the proofs that show that dementia is not just a consequence of biological ageing but a deterioration in the cognitive functionalities.

Out of the many different forms of dementia, Alzheimer's disease is the most occurring type accounting for 60-80 percent of cases. Alzheimer's disease is a common cause of progressive dementia in aged people. Not only Alzheimer's but there are many other causes which may cause dementia. Some symptoms of dementia are reversible.

With 10 million new cases every year, there are 55 million people living with dementia worldwide and it has a high death rate (seventh leading in the world) and many other casualties of disability which makes them dependent on the other people - dependency among older people.

A patient diagnosed with the syndrome may find it tough to remember recent incidents, although they can recall past happenings with ease. They tend to forget the names of everyday objects or their

acquaintances, lose focus while speaking and feel disoriented even when they are in a known environment.

Most people in old age suffer from memory deterioration. This may seem quite common but it can be an early sign of dementia or related medical conditions. Memory loss can also be caused as a side effect of other medications.

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, almost ten million new dementia cases are arising annually. Approximately, 50 million people are suffering from dementia at present.

Presently, dementia does not have any cure or treatment. Although many treatments are being explored in different clinical trial stages, only some disease- modifying therapies and anti-dementia medicines available currently have limited effectiveness and are originally designated for Alzheimer's disease.

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.

People affected with dementia have limited time in their hands before they lose entire control over their body to the syndrome. Thus, it is crucial that they get to make decisions and opt what they prefer in terms of support, financial and legal issues. This can be achieved if the syndrome is identified in its budding stages so that their close ones can have a clear understanding of what they are dealing with and get proper advice regarding the same.

When the syndrome is identified in patients at an early stage and proper support and required resources are provided to them, it allows them to handle their situation, plan their time ahead and thus, live prepared for the things to come. This early diagnosis can also help in clarifying that the symptoms caused due to dementia, that affect the mannerism, mind power, and communication among other problems, are not caused by other possibly treatable conditions with similar symptoms.

By diagnosing dementia in its early stages, friends and family of the affected person can understand the shifts occurring in their life. It can also assist the person with dementia by making resources, support systems, and associated details accessible to them. This will help them decide whether to proceed with drug-excluding or drug-including treatments that are at hand.

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

The symptoms and effects of Alzheimer's disease can be quite similar to those of mild cognitive impairment. In order to explicitly distinguish them, a novel method involving 3D deep convolutional neural networks has been proposed in this study. The purpose behind this is that patients affected with AD need to get diagnosed early in order to facilitate proper care. Additionally, structural MRIs are used to identify cognitively normal individuals.

A reference model was built based on the statistical data of the patients in order to differentiate between them and perform classification using several factors. These factors can include results of Mini Mental State Exam, educational level, approximate volume of total intracranial structure, economic status in the society, Atlas Scaling Factor, Clinical DementiaRating, and Normalise Whole Brain Volume.

In comparison to the model using classification techniques, the deep-learning model proposed in this paper is more precise and quicker in function. This model has many applications - some of which include forecasting progression, re-validating the misinterpreted information and also diagnosing if the patient has dementia or not based on the MRI scans and OASIS datasets. The present model depends on a variety of regions related to AD which is proved by the exploration of the features

associated. Recognising imaging biomarkers that sense dementia and using them to detect the disease in its early stages accurately can be learnt by deep neural networks. These discoveries suggest that the networks can automatically learn to do so.

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] came forth with a method that uses machine learning to detect dementia from Neuropsychiatric Symptoms and Neuroimaging Data which makes use of neuroimaging, Baseline neuropsychiatric, and clinical data from 102 individuals with Normal Cognition and 239 with Mild Cognitive Impairment were taken from the ADNI database. Using a published algorithm, items from the neuropsychiatric inventory questionnaire were transformed into MBI domains. Diagnosis at the modern follow-up was employed as the resulting variable. A ground truth classification was made based on the diagnosis at the latest follow-up.

Problem Statement: To determine if baseline mild cognitive impairment (MCI) status and brain morphology features are anticipated for follow-up prognosis, median 40 months later, in victims with normal cognition (NC) or MCI.

Methodology: Based on the Alzheimer's Disease Neuroimaging Initiative database, baseline neuroimaging, neuropsychiatric, and clinical data were extracted from 102 individuals with NC and 239 with MCI. Ground truth classification was based on diagnosis at the latest follow-up. For predicting follow-up diagnosis, a logistic tree classifier was trained with a feature selection approach based on information gain.

Limitations: ADNI database limitations and study criteria/analysis plan limitations are inherent to this study. As long follow-up time is beneficial for seeing changes in NC at baseline, results might be staggered by not having a maximum cut-off. 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: In addition to brain morphology measures for predicting diagnosis change, baseline NPS categorised for MBI domain and duration have prognostic utility. Alzheimer's disease can be predicted using the MBI total score and impulse controls and regulations.

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] utilises informant-based questionnaires, developed a machine learning-based

method to help the preliminary diagnosis of normal, mild cognitive impairment (MCI), and very mild dementia (VMD). The model took up 5,272 individuals who filled out a 37-item questionnaire to select the features which could be more important than the rest. To develop the diagnostic models, six classification algorithms were combined with the top features.

Results: The most effective among the three feature selection methods is Information Gain. Using the diagnostic model proposed in this paper, clinicians can diagnose dementia at an early stage.

Problem Statement: In the early stages of dementia, reliable diagnosis still remains a challenging issue. Using an information based questionnaire, the model aimed to develop a new method based on machine learning to help the diagnosis of mild cognitive impairment, normal cognition, very mild dementia, and dementia.

Methodology: The model engaged 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 for developing the diagnostic models.

Conclusion: This paper proposes a diagnostic model that helps in identifying early stages of dementia. This acts as a powerful tool for clinicians.

Future Scope: Further Analysis to be done on sampling techniques and classification algorithms

Shanta Rangaswamy. et al. [3] has introduced the use of algorithms involving machine learning concepts to get higher accuracy in order to detect Parkinson's disease. The data set is from UCI, an online machine learning repository. Accuracy, recall, and confusion matrices are calculated using support vector machines, K-nearest neighbours, and linear discriminant analysis algorithms. This implementation yielded 100% accuracy for SVM and KNN algorithms, and 80% accuracy for LDA algorithms.

Problem Statement: To examine the process of diagnosis with greater precision using several illness parameters. The report details the implementation of algorithms for detecting Parkinson's illness that outperform existing models.

Methodology: To use machine learning algorithms such as K-Nearest Neighbours (KNN), Linear Discriminant Analysis (LDA), and Support Vector Machines (SVM).

Limitations: The solution is tested on a dataset with just 196 items to ensure that it has 100% accuracy with KNN and SVM, which may decline with a larger 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 demonstrated one such synergy of fMRI with deep learning by

employing a basic yet accurate strategy for feature extraction and categorization of Alzheimer's disease (AD) utilising resting state fMRI data and a modified 3D convolutional neural network (CNN). The CNN model utilised is designed to work with minimum data preprocessing so that spatial and temporal information on the functional MRI can be preserved.

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: Using a modified 3D convolutional neural network (CNN) features are extracted from resting state fMRI data for feature extraction and categorization of Alzheimer's illness (AD). The CNN is fabricated in a way that it can use fMRI data with minimal preparation while still keeping both spatial and temporal data.

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 early and later phases of MCI and AD can be detected and distinguished by the CNN, suggesting that it could be useful in both early detection and better diagnosis of Alzheimer's disease. Clinical applications may be possible resulting from this.

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] proposed a strategy for automatically classifying AD and VD in patients with mixed clinical assessments that integrates local DTI measurements and GT measures from rs-fMRI data with machine learning. Indeed, multimodal MRI characteristics might be employed to distinguish AD from VD patients with good accuracy and balanced sensitivity and specificity. ANFIS appeared to outperform others in classification performance types within the pool of ML algorithms available to the user (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 constitute a pool of certified ways for analysing data in order to uncover previously unknown patterns. Other research has found that combining machine learning and quantitative MRI is an effective method for not just detecting dementia but also predicting disease progression. A recent study found that using ML in conjunction with volumetric data derived from structural MRI can help differentiate between Alzheimer'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: Because of the promising results, we are confident that ML combined with high-resolution MRI will be an effective tool for physicians to use in their clinical work, assisting them in improving diagnostic and prognostic accuracy, as well as therapy and patient management.

Future Scope: To boost the algorithm's accuracy even more, future research should combine the pattern of features revealed here with clinical and neuropsychological factors and metrics from biological samples.

Sulantha M. et al. [6] suggested a machine learning-based probabilistic technique based on regional information from a single amyloid positron emission tomography scan to assess the progression to dementia within 24 months. Importantly, during a brief observation period, the proposed strategy was developed to overcome the inherent unfavourable imbalance proportions between stable and progressive mild cognitive impairment individuals. The new algorithm outperformed existing algorithms utilising the same biomarker measurements and earlier studies using several biomarker modalities, with an accuracy of 84 percent and an under-receiver operating characteristic curve of 0.91.

Problem Statement: The main pathologic hallmark of Alzheimer's disease is amyloid protein, and indicators of neuronal degeneration are the only ones that are thought to provide accurate forecasts of clinical progression in 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 tomography scan of a single amyloid positron emission like SVM , L1 logistics, L2 logistics Random forest classifier, Random Under Sampling Random Forest (RUSRF).

Limitations: Any Alzheimer's disease study that uses in vivo data suffers from the diagnosis' inherent ambiguity. This is due to the fact that only autopsy data can confirm the diagnosis of Alzheimer's disease. As a result, any clinical diagnosis of Alzheimer's disease will always be "probable Alzheimer's disease." Around 10% of these "probable Alzheimer's disease" diagnoses are likely to be mislabelled as "Lewy Body disease" or "Frontotemporal dementia."

Conclusion: Based on A PET measurement, a novel algorithm to predict the progression of Alzheimer's disease dementia in MCI persons. For an independent test set, the predictor described here had a high rate of accuracy, with an AUC of 0.906. The unique approach corrects for the intrinsic proportional imbalance between stable and pMCI in a population of MCI patients, making it ideal for use in a clinical setting as an early diagnosis device.

Future Scope: Can be used as a tool to calculate the progression or regression of the disease.

Nirjjon. et al. [7] have created the mobi-cog application, which runs on a mobile device such as a tablet or smartphone and provides an automated and real-time dementia screening service. The MOBI-COG App is a total automation of the Mini-Cog test, a popular 3-minute dementia screening test used by primary caregivers for fast dementia screening in the elderly. The patient is asked to memorise a set of three words and complete a free-hand clock drawing task. All of these steps of the test are automated by the MOBI COG App.

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: The MOBI-COG App is a mobile application that was designed, implemented, and evaluated in this research. An automated micro cog test is used in this application. This app cannot be used in place of consulting with 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 produced a list of iTunes apps for use by persons with early-stage or moderate dementia. The procedure for selecting these apps and assessing the most important characteristics of the most popular apps was detailed. The findings will contribute to the growing body of knowledge about the innovative use of smart technology in healthcare settings.

Welsh et al. [9] created Ticket to Talk, a mobile application that encourages young people to acquire media relevant to people with dementia to use in conversations with people with dementia. Ticket to Talk has been evaluated through trials with two families and two groups of elderly individuals. We discuss the challenges of employing technology like this as a conversational tool, the importance of digital media in facilitating intergenerational relationships, and the potential to positively change people with dementia's social agency.

VocabChecker, a smartphone-based dementia screening tool created by **Shibata et al. [10]**, analyses language ability from a voice narrative via automatic speech recognition (ASR). The model focuses on four language abilities that are linked to dementia: the number of tokens, the types of tokens, the type to token ratio, and the prospective vocabulary size. It was also noted that the usage of VocabChecker allowed dementia sufferers to be recognised from older persons.

For persons with advanced dementia, **Atee et al. [11]** developed a unique technique and system of pain assessment employing a variety of technologies, including automated face recognition and analysis (AFRA), smart computing, affective computing, and cloud computing (Internet of Things).
Problem Statement: To explain a unique approach and system for assessing pain in adults with severe dementia that combines technologies such as automated facial recognition and analysis (AFRA), smart computing, affective computing, and cloud computing (Internet of Things).
Methodology: The system was conceptualised using cognition and affective computing. To address the difficult problem of recognising pain in nonverbal dementia patients, a computerised clinical system was devised. A smart device-enabled app (App) is linked to a web admin page in the system. To generate a pain intensity score, the app "PainChekTM" uses AFRA (Automated Facial Recognition and Analysis) to identify facial action units indicative of pain presence and user-fed clinical information.
Limitations: The system's ability to collect a vast volume of pain data over time aids in the detection of temporal patterns.
Conclusion: PainCheck is an evidence-based, full-scale pain management solution. Because it may be utilised by clinicians and professions in regular clinical practice, this unique technique has the potential to change pain assessment in persons who are unable to verbalise.
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.

In order to improve user experience in home dementia care **Megges et al. [12]** stressed the necessity of integrating people with dementia and their primary carers in the research and development phase of locating systems.
Problem Statement: The user experience using the location information system of people with dementia and their primary caregivers is not well established.
Methodology: Eighteen days using the prototype tracking system for four weeks. The main result is to evaluate the ease of use and the features and features of the product. The parameters of the secondary outcome were caregiver burden, perceived self-efficacy, frequency of use, and willingness to purchase a prototype. Score changes between the baseline (T1) and the end of the test period (T2)

were compared using independent and dependent sample correlations and descriptive statistics.

Limitations: It has been defined that methodological obstacles together with relying on proxy evaluations of consumer revel in by using having caregivers or experts answer in place of PwD assist contribute to the observed underutilization of finding structures. moreover, the importance of ok expertise on the use of locating structures become emphasised. Others examine the usage of GPS technology targeted on research questions concerning mobility and cognitive impairment.

Conclusion: The main conclusion is that it is relevant to perform randomised controlled trials with larger and more representative samples in a real environment. This recommendation was also made by other authors. Appropriate results measures proposed in previous studies were "search time" and "days to long-term hospitalisation", and caregiver well-being and quality of life.

Future Scope: Technical limitations often hinder the investigation of the usefulness of assistive technology, so it is advisable to focus on usability until these types of limitations are overcome. In addition, future studies involving a variety of stakeholders will help gain more insight into how to properly address PwD and caregiver needs and preferences in order to improve the user experience of location systems. There is a possibility. This type of research covers many disciplines such as economics, design, gerontology, neurology, and psychiatry, so the focus should be on interdisciplinary research.

Zorluoglu et al. [13] have developed a mobile app to improve the pain assessment and management of this vulnerable person. They conducted usability testing of a newly developed iPhone pain assessment application for potential users. In this case, it is used as a tool for EMS clinical practice to improve pain assessment in the elderly with cognitive impairment.

Sindi. et al. [14] created the CAIDE Risk Score (mobile application) App, which includes information on age, educational level, hypertension, hypercholesterolemia, obesity, and physical inactivity, and is based on the CAIDE Dementia Risk Score. The CAIDE Risk Score App is the first to use a widely accepted evidence-based approach to predict dementia risk. Users can utilise the app to actively reduce their modifiable risk factors and delay cognitive impairment and dementia.

Problem Statement: The CAIDE (Cardiovascular Risk Variables, Ageing, and Incidence of Dementia) Dementia Risk Score is a validated method that uses midlife vascular risk factors to predict late-life dementia risk (20 years later). The goal was to make this prediction tool as accessible as possible.

Methodology: The CAIDE Risk Score (mobile application) App was created using data from the CAIDE Dementia Risk Score, which includes age, years of education, hypercholesterolemia, hypertension, physical inactivity and obesity.

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 predict dementia risk using a widely used proof-based approach. Users may utilise the app to actively reduce their modifiable risk factors and delay cognitive decline and dementia.

Future Scope: The App's use will inspire users to actively reduce their risk factors, which may help to delay the onset of dementia and other chronic illnesses.

Thorpe. et al. [15] created a prototype that brings together a smartphone, wristwatch, and a variety of apps to provide six support functionalities. This has been endorsed by five people with disabilities (PwD) and their carers. Field testing in a real-world setting follows supervised usability testing. Video recordings, logbooks, system usability scale surveys, application use logs, interaction logs, and interviews are used to compile data based on the unified theory of technology acceptance and use. The data is examined in order to assess usability, utility, and user approval.

Problem Statement: Smart mobile and wearable technology's availability and popularity may potentially aid user acceptance – a major obstacle for assistive technology (AT) for people with disabilities that necessitates the use of user-centred design (UCD) methodologies.

Methodology: End-user interaction and the usage of basic prototypes are encouraged in research on co-design methodologies for users with cognitive impairment. A UCD strategy that incorporates users and includes evaluation of usability and adequacy is documented in several projects creating AT for dementia. Rather than using off-the-shelf solutions, these examples use UCD methodologies to design specialised AT equipment.

Limitations: In terms of data collecting and analysis, the user testing methodologies were time-consuming, restricting the number of participants. Because PwD differs significantly in terms of symptoms, lifestyle, background, and technological literacy, among other factors that may influence acceptance, it is impossible to draw broad generalisations about user adoption of the ubiquitous AT solution. Instead, our findings give an initial impression, indicating some promise for user adoption and highlighting major challenges that need to be investigated further.

Conclusion: This project involves the creation and testing of a widespread AT solution using off-the-shelf smart technologies. These devices have an advantage over specialised AT in terms of user uptake, which necessitates a review of their usability, adequacy, and acceptability.

Future Scope: Finally, because adoption in clinical practice is a substantial obstacle for wearable technology in healthcare, it is highlighted as an essential subject for future study. To speed and assist adoption, healthcare practitioners must devise strategies for introducing ubiquitous AT to people with disabilities and incorporating it into their daily activities.

The feasibility, safety, and value of tablet devices as innovative nonpharmacologic techniques in managing elderly mental inpatients with agitation and dementia are investigated by **Vahia et al. [16].** 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.

The creation and usability assessment of EnCare diagnostics (ECD) and the brain fit plan (BFP) in healthy older persons, cognitively impaired, and physically challenged individuals was the focus of **Boyd et al. [17].** 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. When provided shorter versions to fit their diagnosis, ECD is extremely acceptable in both healthy older persons and those with early stage dementia, according to this usability research.

Dethlefs et al. [18] developed a spoken natural language interface that allows dementia patients to communicate with cognitive stimulation software in the same manner they would with a real 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] perform analysis of ADNI data to build three categories of classification models (normal cognitive impairment (CN), mild cognitive impairment (MCI), and dementia (DEM)) based on matching of three deep learning models. We evaluated the effectiveness of the event. The ReliefF algorithm proved to be superior to information capture techniques and was used to select the most useful features.

Problem Statement: To perform an analysis of ADNI data and create a classification model that distinguishes between the categories of cognitive normal (CN), mild cognitive impairment (MCI), and dementia (DEM), based on three deep adjustments, is the main purpose behind this paper. It's about determining the effectiveness of building. Learning model: Two multi-layer perceptrons (MLP1 and MLP2).

Methodology: UK routine primary care patient records are used to create risk scores for estimating the risk of developing dementia using traditional statistical techniques and state-of-the-art machine learning algorithms. Positron emission tomography (PET) scans and local analysis of protein amyloid β were used in a random forest classifier to identify patients with stable MCI and pMCI with age. A recent EMIF-AD study introduced a machine learning technique based on Random Forest, Extreme

Gradient Boosting (XGBoost), and deep learning.

Limitations: Deep learning requires much more advanced data preprocessing and feature engineering, and much more computational time to train a single model compared to statistical implementations.

Conclusion: All models can recognize patterns that distinguish between the three classes DEM (dementia), MCI (mild cognitive impairment), and CN (normal cognition), and each of these machine learning approaches corresponds to AD VD and MCI. It suggests that you can accurately predict.

Future Scope: The ConvBLSTM model was slightly less accurate, but it was considered for comparison with the MLP model and future extensions of this work using time-related information.

M. Tanveer et al. [20] have researched from 2005 to 2019, we surveyed 165 articles and created reviews using feature extraction and machine learning techniques to gain a deeper understanding of our research on Alzheimer's disease. They provided a detailed overview of the three approaches to Alzheimer's disease, along with support vector machines (SVMs), artificial neural networks (ANNs), and deep learning (DLs).

Problem Statement: Early detection of Alzheimer's disease requires efficient automation techniques. For the classification of Alzheimer's disease, researchers have proposed several cutting-edge approaches. However, research on Alzheimer's disease needs to be better understood in order to develop more efficient learning methods. After reviewing many articles on machine learning techniques, I gave a detailed overview. This experience helped them gain more knowledge about Alzheimer's disease.

Methodology: Machine learning techniques are being explored in three main categories: Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Deep Learning (DLs), and Ensembles. A detailed overview of these three approaches to Alzheimer's disease and future directions.

Conclusion: This task analyzes papers that use three major machine learning techniques: SVM, ANN, and DL, to diagnose Alzheimer's disease. It also covers research on other learning techniques such as transfers, ensembles, and multi-kernel learning.

G. Battineni et al. [21] studied the use of support vector machines (SVMs) in predicting dementia and verified their performance through statistical analysis. They achieved optimised results with efficient performance values.

Problem Statement: Predict dementia using a support vector machine (SVM) and verify its performance through statistical analysis.

Methodologies: Support vector machine (SVM) in predicting dementia and validated its work efficiency with the help of 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, scored, categorised, and summarised the results of deep learning tasks for Alzheimer's disease by algorithms and neuroimaging types. The combination of multimodal neuroimaging and liquid biomarkers is said to provide the best classification performance. Several issues, such as transparency and reproducibility, need to be addressed in order to use deep learning to make diagnostic classifications and prognostic predictions and to be ready for clinical application in the real world.

Problem Statement: The usage of deep learning for early detection and automatic classification of Alzheimer's disease (AD) has lately gained considerable attention owing to the rapid advances in neuroimaging technology that have produced large-scale multimodal neuroimaging information. The project undertook an orderly review of publications using neuroimaging data and deep learning approaches to diagnostically classify AD.

Methodologies: Convolutional neural networks (CNNs), recurrent neural networks (RNNs) and other such deep learning techniques that use neuroimaging data without feature selection preprocessing achieve up to 96.0% accuracy in classification of AD.

Limitations: Gradient calculation methods and efficient error functions were explored in these original papers, inspired by the illustrated limitations of single-layer perceptrons. These are only capable of learning linearly separable patterns.

Conclusion: Also, owing to the complexity of deep learning algorithms with multiple hidden layers, it is complex to discover how these selected features lead to judgements. There is also uncertainty about how certain features or their subclasses lead to their relative importance.

Future Scope: Ultimately, future research needs to recreate the most important insights from deep learning about completely independent datasets. The system lags behind the penetration of neuroimaging data and the associated deep learning technology. However, it is widely used in genetics and related fields.

Vijay S. et al. [23] worked to extend previous studies by making model estimates based on a very large integrated dataset of medical and electronic health record (EHR) data. They hypothesised that

using larger, more complex data, label learning, and advanced machine learning estimation methods would significantly improve the dementia prediction algorithm. They adopted a label learning methodology created to address the problems identified in previous studies.

Problem Statement: The purpose of the study was to build a machine learning model for predicting mild cognitive impairment, Alzheimer's disease, and associated dementia from structured data using managed and electronic health data sources.

Methodologies: The patient cohort (n = 121,907) and controls (n = 5,307,045) were created for modelling using data within 2 years of the patient's diagnosis date. An additional cohort 3-8 years away from the index data will be used for forecasting. The training cohort matched age, gender, index year, and workload and was equipped with a gradient boosting machine, lightGBM.

Conclusion: The quality of the 2-year incident model on the holdout test set was 47% sensitivity and 87% under-curve area. In the 3-year model, the learned labels reached 24% (71%), but by the 8th year they had dropped to 15% (72%).

Future Scope: Further progress requires better methods, more patient and family data, and better clinical documentation to determine which patients were misidentified in the source data.

H. Niu et al. [24] have estimated the prevalence and occurrence of Alzheimer's disease (AD) in Europe. Novelty is the strength of this study, as no meta-analysis of the subject has been performed before. They have recently observed a slight increase in AD prevalence and a decrease in incidence in Europe. The authors assume that similar trends studies on Alzheimer's disease are needed in other parts of the world.

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

Methodologies: Literature searches were performed on Medline, Scopus, and CINAHL Complete using the keywords "Alzheimer's", "Alzheimer's disease", and "AD" in combination with "prevalence," "incidence," and "epidemiology." rice field. A Bayesian random effects model with a 95% confidence interval was used. I2 statistics were applied to assess non-uniformity.

Limitations: One of the main limitations of this study was that they couldn't rate more articles because they couldn't access more databases.

Conclusion: A better understanding of the effects of this disease in Europe can be enabled by the results of our meta-analysis.

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] Using machine learning techniques, proposed a two-layer model motivated by the

methods used by the Dementia Support Centre for early detection of dementia. Several supervised learning algorithms were used and the performance of each algorithm was compared to each other. This proposed model aims to reduce the time and financial burden and simplify the process of diagnosing dementia. This reduction helps to diagnose early-stage dementia quickly, cheaply, and reliably, improving current clinical practice.

Molinuevo J. L. et al. [26] have been working on the ALFA project. Its goals include pointing out early pathophysiological events and developing preventive programs for Alzheimer's disease. The ALFA project constitutes a valuable infrastructure that uses a variety of research and trials to prevent AD. The long-term ALFA + cohort helps to elucidate the natural history of the disease, model preclinical stages and design successful studies.

Problem Statement: The preclinical stage of Alzheimer's disease (AD) is ideal for developing early pathophysiological event detection and prevention programs. These are common goals of ALFA projects such as ALFA registries, parent cohorts, and ALFA + nest cohort studies.

Methodologies: Baseline visits to the ALFA parent cohort included a complete cognitive assessment, lifestyle questionnaire, DNA extraction, and MRI. Nested ALFA + studies add wet and imaging biomarkers for deeper phenotypes.

Conclusion: A total of 2743 participants aged 45-74 have been enrolled in the ALFA Parent Cohort. This cohort, consisting primarily of a cohort of cognitively normal offspring of AD patients, indicates that they are rich in genetic risk factors for AD.

Future Scope: Throughout the ALFA project, neuroimaging and biological markers present in the preclinical AD stage that precede or signal the presence of Aβ deposits in the brain are detected. In addition, the biomarkers used and validated in these studies provide endpoints for designing large-group, randomised controlled trials that will lead to the development of pharmacological and non-pharmaceutical interventions in individuals at risk for AD.

Koo B. et al. [27] have published a review of the literature on how mobile platforms, especially smartphones and tablets, are used in cognitive assessment of the elderly, and related benefits and opportunities. They summarised the findings, categorised them by level of innovation, identified research gaps, and made suggestions for future research.

Problem Statement: This article reviews the literature on how mobile platforms (smartphones and tablets) are used for cognitive assessment of the elderly, and the benefits and opportunities associated with using mobile platforms for cognitive assessment.

Methodologies: The process began with a systematic search of published literature. Next, we analysed selected articles on the key elements of mobile cognitive assessment and categorised them

according to the degree of innovation.

Conclusion: Twenty-nine articles met our selection criteria and were divided into three groups: (A) A mobile version of an existing article or a computer-based neuropsychological test. (B) A new recognition test specifically designed for mobile platforms. (C) Use new types of data for cognitive assessment. This scoping review confirms the considerable potential of mobile ratings.

Future Scope: Future work will need to explore new sensors and wearable technologies that improve reliability and validity, including more diverse participant samples, and incorporate the perspectives of the individual, caregiver, and clinician being evaluated.

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 carefully use factors like 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 genetic tests, blood tests, brain scans, cognitive and neurological tests and psychiatric evaluation, 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 played a pivotal 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 recognised self-awareness, willingness to purchase the prototype, usage frequency, 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 ageing, cardiovascular risk factors 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 candidates.

Community-based trials, developing and evaluation of the operability 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 designs
- Performing analysis over the data comprising of 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.

ML model is built to detect dementia 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
- Atlas Scaling Factor
- Clinical Dementia Rating
- Socio Economic status
- Estimated Total Intracranial Volume
- Mini Mental State Examination report
- Whole Brain Volume

The Data Classification is an ensemble classification of Random Forest Classification over dataset with features such as age, MMSE score, years of education, normalise whole brain volume, socio economic status, clinical dementia rating, atlas scaling factor estimated total intracranial volume, 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. Ensemble learning combines several classifiers to find solutions to complex problems. To create an uncorrelated forest of trees, it uses bagging and features randomness when building each tree to make better predictions by committee than by individuals.

Random forest models perform better than individual models because they consist of a large number of relatively uncorrelated models (trees) operating together. The random forest is a classification algorithm consisting of many decision trees.

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 predictive accuracy.

For both regression tasks and classifications, random forest algorithms can be used. Through cross-validation, it provides a higher level of accuracy. 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.

To minimise discrepancies in data distributions and biases, robust diagnostics should adapt to various datasets, such as MRI scans collected by multiple patient groups. The goal of deep learning is to reduce the need for domain experts in designing and extracting discriminative features.

Non-invasive neuroimaging tools such as structural MRI (sMRI), functional MRI (fMRI), and positron emission tomography (PET) are being investigated for the development of such a system. Based on the extracted features, a classifier is trained to distinguish between different groups of subjects, such as AD, mild cognitive impairment (MCI), and normal controls (NC) groups.

The sMRI has been recognized as a promising indicator of AD progression. Compared to the known diagnostic systems, the proposed system uses a deep 3D convolutional neural network (3D-CNN) pre-trained by a 3D convolutional autoencoder (3D-CAE) to learn generic discriminative AD features. A dataset collection in different domains can easily be adapted to capture characteristic AD biomarkers. The discriminative loss function is imposed on peripheral layers of 3D-CNN to increase their specificity.

Deep 3D CNN can detect and extract characteristic AD biomarkers in one domain (source) and perform task-specific classification in another domain (target) using generic and transferable features. In the proposed network, the lower layers combine a generic feature-extracting stacked 3D-CAE pre-trained in the source domain with the upper fully connected layers fine-tuned in the target domain.

NLP

Dataset: Dementia Bank

Natural language processing (NLP) is a module of artificial intelligence (AI). It can be explained as the capacity of a set of lines of code that helps the machine in understanding languages spoken by humans, called the natural language. NLP is a branch that emphasises on how the field of data science is interlinked with the human language. It is spanning across a number of industries lately.

A patient's language variation can be used as a proxy for measuring how their cognitive functions have been affected (e.g., impaired reasoning and difficulty finding words). The machine learning models can exploit these language patterns and thus help in diagnosing the patient.

Code for LSTM (Long Short Term Memory) is done by comparing it with Particle Swarm Optimisation Methodology. The Long Short Term Memory is a sequential network based on an advanced recurrent neural network technique that holds data for a significant period of time. A recurrent neural network (RNN) is mainly utilised for its continual memory. LSTM is competent enough to resolve the issue of vanishing gradient faced by these RNNs.

In CNN, we implement a convolutional layer with an embedding, followed by a max pooling layer for each of the sentences. In order to obtain features of convolution, diversified window-size filters are used on each window of words. Subsequently, to create probabilities for two classes, the outcome is sent to a softmax layer.

We make use of an LSTM-RNN model since long-range sequential correlations can not be captured by CNNs. This model has an embedding layer prior to the LSTM layer. The ultimate state contains data pertaining to the whole sentence that is passed on to a fully-connected layer followed by a softmax layer, after procuring the probabilities of the output.

In the wake of noticing that both CNN and LSTM-RNN models produce outcomes corresponding to the previous best performing approach and moreover, considering that either of them have their own complementary potentials, the idea of experimenting with a hybrid architecture was implemented. This approach involves layering an LSTM layer on top of CNN. The top layer is similar to the vanilla LSTM-RNN after the embedding layer, whereas the bottom one is identical to the vanilla CNN before the max-pooling layer. Thus, this model can be referred to as the CNN-LSTM model.

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 techniques based on machine learning to predict the output. User specific result is predicted depending on the input given by the user that 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 Mini-Mental State Examination, Years of Education, Estimated Total IntraCranial Volume, Clinical Dementia Rating, Normalise Whole Brain Volume, Atlas Scaling Factor and Socio-Economic Skills.

The algorithms responsible for classifying the patient's data based on the dataset provided form the basis of this dementia detection system. 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 instrumentative in boosting the performance of the model.

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 a 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. 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 8 GB 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. 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.

6.1.2Definitions

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: This term refers to the stage between expected cognitive decline of typical ageing and the more severe plunge of dementia.

Alzheimer's Disease: This is an accelerating disorder that results in the brain getting smaller in size and its cells ceasing to exist.

Vascular Dementia: This type of dementia involves downswing in reasoning skills. It affects various regions of the brain depriving them of functioning properly. Vascular Dementia is caused by the conditions that block or reduce blood flow to those regions.

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 model based on machine learning, under the category of supervised learning which is employed for both classification and regression. 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: SVM stands for Support Vector Machine, which 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 customers.

It is also known 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 contextual model that talks about the composition, functioning and views of the system. The description of the architecture is formal and the representation of the system is arranged in such a way that it recognises thinking about the compositions and ways of the system. This architecture of the model 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 depending on the data provided by the user as input load.
- 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.

This model is also referred to as a linear sequential life cycle model. In this prototype, 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. A broad idea for the performance is doing the whole requirement analysis simultaneously. 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. This solution will be displayed and saved in the notebook right below the code cell that initiated it. These are the ultimate plots that are stored.

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 Diagram

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 and diagrams of complex systems is the best way to learn about it. These figures have a greater influence on our notion. Looking around, you can see that charts are not a new conception and are widely used in different ways in several 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 provides insights into the system from the user's perspective by modelling the user's system capabilities with respect to what the user expects from the system.

Structural Model View

This layout represents the details and functionality of the device. 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. UML Behavioural diagrams represent the elements of a time-dependent system and convey the dynamic concepts of the system and how they influence each other. Interaction diagrams, activity diagrams, state diagrams, and use case diagrams are different types of behaviour 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 popular visual modelling language used to specify, visualise, build, and document software system artefacts.

7.1.2 UML is a language

It provides glossaries and rules for communication and works on conceptual and physical expressions. So this language is a modelling language.

7.1.3 UML Specifying

Specify the means to create an accurate, clear, complete model. Specifically, UML is involved

in specifying all the key analytical, design, and implementation decisions that need to be made in the development and representation of software-intensive systems.

7.1.4 UML Visualisation

UML includes both graphic and textual representations. This makes it easier to visualise and better understand the system.

7.1.5 UML Constructing

UML models can interface directly with a variety of programming languages, are expressive enough, and are unambiguous enough to run the model directly.

7.1.6 Building blocks of UML

UML vocabulary contains 3 types of building blocks

- Things
- Relationships
- Diagrams

7.1.6.1 Things

Things are the abstraction of data, which is the main citizen of the model. There are four types: structural, behavioural, grouped, and annotated.

7.1.6.2 Relationships

Relationships connect things together. UML relationships are dependencies, associations, generalisations, and specialisations.

7.1.6.3 UML Diagrams

A diagram is a graphic representation of a set of elements, most often represented as a connected graph of vertices (things) and arcs (relationships). There are two types of figures:

1. Structural Diagrams

2. Behavioural Diagrams

7.1.6.3.1 Structural Diagrams

The four UML structural diagrams are used to visualise, specify, build, and document the static aspects of the system. You can use any of the following figures to view the static parts of your system. The structure diagram consists of a class diagram, an object diagram, a component diagram, and a layout diagram.

7.1.6.3.2 Behavioural Diagrams

UML's five behavioural diagrams are used to visualise, specify, build, and document the dynamic aspects of the system. UML operating diagrams are roughly organised according to the most important methods that can model the dynamics of the system. Operation diagrams consist of use case diagrams, sequence diagrams, collaboration diagrams, state diagrams, and activity diagrams.

7.2 UML DIAGRAMS

7.2.1 Use Case Diagram

The driving force behind use case diagrams is to capture the dynamic aspects 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 a result, features and actors are identified when analysing the system for recording use cases.

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 features described in an organised way. Actors are the second one related to use cases. Actors

are objects that interact with the system.

Actors are external applications, internal applications, or human users. These factors need to be considered when creating use case diagrams.

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

Fig 7.1 Figure representing the Use Case Diagram of the system

Actors

- User
- Admin
- Doctor

Use Cases

Use cases are instructions on how website users can complete tasks. Describes how the system

behaves when responding to queries from the user's perspective. Every use case is represented by a chain of simple steps that start with the person's aim and cease when that aim is performed.

Graphical Representation

An oval shape can be used to represent use-cases. 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 assist you apprehend device requirements and talk with pinnacle users and area specialists. check the device. The quality manner to discover a use case is to study the actors and define what the device 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 whilst an employment case starts off evolved 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 join actors to apply cases. keep in mind that the reason for the use case is to offer some gain to the actors inside the system. Consequently, it's critical to document those interactions between actors and use cases. traces 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 kind of interplay diagram for explaining how hard and fast objects interact with each other (and where in 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:

Simply as you can use UML series diagrams to discover the logic of use instances, you could also use UML series diagrams to discover the common sense of any function, process, or complicated system. Collection diagrams are an extremely good manner to symbolise provider common sense whilst you think about a carrier as a high-level technique used by distinctive customers.

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 country, a lead, and a persona. The structure and path of objects which are, for all intents and functions, indistinguishable are depicted in the fundamental class. Within the diagram, each object represents a selected instance of magnificence. Order case is an item that’s not named

Message:

A message is an exchange of information between two items that trigger an event. Information is sent from the control convergence source point to the control convergence destination point by means of a message.

Link:

An existing association between two objects containing a class. This means an association between the opposite classes. If the object is linked to itself, use the image 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 collection begins from the consumer, wherein the user provides the Dementia scientific records. The Dementia API collects this information and sends this information to the 3D CNN version. The Dementia dataset is pre-processed through disposing of noise and splitting the dataset into trying out and training sets. Modelling and trying out of ML classifiers is achieved and the Dementia effects are anticipated. consequences concerning Dementia are exhibited to a person thru Dementia API. sequence starts with the user shifting through the Dementia API, 3D CNN version, Dementia dataset and returns to the user.

7.2.3 Class Diagram

The class diagram depicts a static view of a utility. It represents the varieties of objects dwelling within the device and the relationships between them. a category includes its gadgets, and also it can inherit from other instructions. A class diagram is used to visualise, describe, document various extraordinary components of the device, and also assemble executable software program code.It suggests the attributes, instructions, features, and relationships to offer a top level view of the software gadget. It constitutes class names, attributes, and functions in a separate compartment that helps in software program improvement. on the grounds that it's a collection of classes, interfaces, associations, collaborations, and constraints, it's far termed as a structural diagram.

Class diagrams are often used when modelling object-oriented systems. This is because the class diagram is the only UML diagram that 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 visual class representations consists of associations, collaborations, class diagrams, interfaces, and constraints. Also called a structural diagram.

The goal of the elegance diagram is for representing a static-view of your software. class diagrams are regularly used at some point of construction as those are the best diagrams that may be immediately mapped to an item oriented language. UML diagrams, which includes sequence diagrams and interest diagrams, can handiest constitute the collection drift of an utility. but, the class diagram is a little one-of-a-kind. This is the maximum extensively used UML diagram within the programming network.

The cause of the magnificence diagram is to represent a static view of your application. class diagrams are frequently used during production because they are the most effective diagrams that can be mapped without delay to an object-oriented language. UML diagrams, along with interest diagrams and collection diagrams, can best constitute the series waft of an application. However, the magnificence diagram is a little one of a kind. This is the most extensively used UML diagram in the

programming community.

The magnificence diagram’s motive is briefed as underneath:

- layout and evaluation of an application's static view.
- Describe the system obligations.
- the foundation for deployment and aspect diagrams.
- each forward and opposite engineering are used.

Class diagrams take many residences to reflect whilst drawing, however the diagram is viewed inside the pinnacle level in this case. The class diagram is a graphical illustration of a static view of a device which represents diverse components of software. The whole system is represented with the aid of the gathering of the class diagrams.

- The subsequent factors should be taken into consideration whilst developing the magnificence diagram:
- The name of a category diagram that is significant to explain a system issue.
- The element & its relationship might also get diagnosed ahead of time.
- each class duty: methods and attributes might be clearly defined.
- The traits number for every class is probably special, as pointless houses may complicate the diagram.
- At the same time as describing the unique component of the diagram, use notes. Developer/coder needs to understand the drawing on the cease.
- In the end, in advance the diagram is to be drawn on simple paper before developing a very last version and to make certain proper accuracy is maintained, it must be transformed as many times as it could be performed.

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.

Following are the questions whose answers determine the difference between partial and complete 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

The activities involved in processing crude information and preparing it to be fit for a model running on machine learning techniques form data preprocessing. This step is the first and foremost stage in the process of creation of a model.

Unprocessed data usually consists of inconsistent information, missing values, data in an impractical structure, and noises. Due to these irregularities, this data can not be used for feeding the models directly. Preprocessing of the data is essential for cleaning and making it usable enough. This enhances the efficiency and performance of the model.

Missing data can be dealt with in two different ways. We can either eliminate a specific row or estimate the mean.

Eliminating a particular row is usually performed when we are dealing with void values. Here, the selective row or column which contains null values is deleted. However, this process is not that effective as there might be data loss due to the removal of data. This might not result in accurate

results.

As an alternative approach, the mean of a particular column or row that consists of one or more missing values can be computed. The missing values can be replaced with the obtained mean. This approach is the most useful for numeric features such as year of birth, income, age, etc.

```
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

```
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, clinical dementia rating, years of education, socio economic status, normalise whole brain volume, atlas scaling factor, estimated total intracranial volume, 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 is a collection of a variety of discrete decision trees. These trees work as an ensemble or an entity. Every single tree in the random forest produces a class prediction and the one with the highest number of votes becomes the prediction of the entire model. Ensemble learning is used in this process, which is the approach that involves combining various classifiers to solve complex problems. Techniques like feature randomness and bagging are used while building a tree in order to build an unrelated forest whose combined prediction is more precise than that of any single tree.

The conglomerate of unrelated trees working as a group will surpass each of the isolated models. This is the reason behind the successful working of the random forest model.

In other words, the random forest model manages over-fitting problems and boosts the correctness of predictions using averages. It applies decision trees on several different parts of the dataset.

This classifier manages the void and skipped values in order to enhance the performance of large amounts of information. Therefore, high accuracy is achieved with the help of cross-validation. The random forest models can be applied on tasks involving both regression and classification.

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 into training for the model while the remaining 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, the first and the last layers act as the input and output layers respectively. Whereas, the four neural layers in the middle are treated as the hidden layers. 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.

Some sought-after non-invasive neuroimaging tools, such as positron emission tomography (PET), structural MRI (sMRI), and functional MRI (fMRI), have been explored for creating such a model. The former draws out features from the images at hand, and a classifier is trained to differentiate between various clusters of subjects, e.g., AD, mild cognitive impairment (MCI), and normal control (NC) groups.

The sMRI has been recognized as a favourable indicator of advancement of AD. Compared to the known diagnostic systems, the proposed system applies a deep 3D convolutional neural network (3D-CNN) pre-skilled by 3D Convolutional Autoencoder (3D-CAE) to learn generic discriminative AD features in the underlying layers. This captures characteristic AD biomarkers and can be easily refashioned into various domain-specific datasets. In higher layers of 3D-CNN, the specificity of features is enhanced by applying deep supervision, which basically means that the discriminative loss function is enforced on upper layers.

The presented deep 3D-CNN for learning generic and transferable features across various domains is able to identify and draw out the characteristic AD biomarkers in one (source) domain and perform task-specific classification in another (target) domain. The proposed network integrates a generic feature-extracting stacked 3D-CAE, pre-trained in the source domain, as deeper layers with the higher task-specific fully connected layers, which are fine-tuned in the target domain.

The autoencoder learns and automatically draws out discriminative features of AD capturing anatomical alterations caused due to it. It performs this activity in order to overcome the limitations of feature extraction in the modern techniques. The 3D-CAE convolutional filters are trained prior to being adapted to another domain dataset, e.g., to the ADNI after pre-training on the CADDementia. Then the complete neural network is built by stacking the pre-trained 3D-CAE encoding layers followed by the fully connected layers, which are fine-tuned to improve the task-specific classification performance by employing deep supervision. These models are called Deeply Supervised Adaptable 3D-CNN.

```
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')
```

```
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.

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 a module of artificial intelligence (AI). It can be explained as the capacity of a set of lines of code that helps the machine in understanding languages spoken by humans, called the natural language. NLP is a branch that emphasises on how the field of data science is interlinked with the human language. It is spanning across a number of industries lately.

A patient's language variation can be used as a proxy for measuring how their cognitive functions have been affected (e.g., impaired reasoning and difficulty finding words). The machine learning models can exploit these language patterns and thus help in diagnosing the patient.

Code for LSTM (Long Short Term Memory) is done by comparing it with Particle Swarm Optimisation Methodology. The Long Short Term Memory is a sequential network based on an advanced recurrent neural network technique that holds data for a significant period of time. A recurrent neural network (RNN) is mainly utilised for its continual memory. LSTM is competent enough to resolve the issue of vanishing gradient faced by these RNNs.

In CNN, we implement a convolutional layer with an embedding, followed by a max pooling layer for each of the sentences. In order to obtain features of convolution, diversified window-size filters are used on each window of words. Subsequently, to create probabilities for two classes, the outcome is sent to a softmax layer.

We make use of an LSTM-RNN model since long-range sequential correlations can not be captured by CNNs. This model has an embedding layer prior to the LSTM layer. The ultimate state contains data pertaining to the whole sentence that is passed on to a fully-connected layer followed by a softmax layer, after procuring the probabilities of the output.

In the wake of noticing that both CNN and LSTM-RNN models produce outcomes corresponding to the previous best performing approach and moreover, considering that either of them have their own complementary potentials, the idea of experimenting with a hybrid architecture was implemented. This approach involves layering an LSTM layer on top of CNN. The top layer is similar to the vanilla LSTM-RNN after the embedding layer, whereas the bottom one is identical to the vanilla CNN before the max-pooling layer. Thus, this model can be referred to as the CNN-LSTM model.

```
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 related to reducing the threat of developing the syndrome. The feeling of well-being, communication improvement and better interaction can be achieved through mentally challenging activities, such as doing a jigsaw puzzle, that stimulates thinking and memory. 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 estimating a system or part thereof in order to determine whether it meets the stated requirements or not. This work results in realities, expectations, and differences between its outcomes. In simple terms, testing is the use of a system to identify any gaps, errors, or needs that are not in conflict with actual desire or requirements. Software testing is an important aspect of software quality assurance and represents the final review of specifications, formats, and coding. It represents an interesting software confusion.

Testing is a process of finding errors. It plays a very important role in ensuring quality and reliability of the software. Test results are used later during the adjustment process. Typically, the test phase includes advanced system testing using various test data. Test data modification plays an important role in system testing. After processing the test data, the study program was tested using that test data. During the system testing, errors were detected and corrected using test steps and corrections are also recognized for future use. Thus, a series of tests are performed on the proposed system before the system is ready to be implemented. Therefore, the purpose of the test is to show that the system is working by showing that it has no errors.

The main purpose of the testing phase is to identify potential errors in the system. Therefore, testing can provide software that meets expectations, prevents unintended consequences, and improves long-term maintenance of your app. The appropriate method will be used depending on the purpose of the test and the software requirements. Whenever possible, the test can be run automatically.

9.2 Testing Objectives

The main goal of testing is to find various bugs in a systematic way with minimal effort and time. Formally, testing is the process of running a program to find bugs. A successful test is a test that finds errors that have not yet been found. A good test case is one that is likely to find a bug if it exists. The software more or less confirms 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 situational or requirements-based testing, it can be categorised as follows:

- Unit Test
- Integration Test
- Regression Test
- Alpha Test
- Beta Test

9.3.1 Manual Testing

Manual testing is a way to test software that runs test cases manually without using it in an automated process. These test cases are run by the tester from the end user's point of view. The test case covers all edge cases to ensure proper functioning 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 are:

• 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 methodologies are defined as the test methods and types used to ensure that the software under test meets customer expectations. Testing methods include active and inactive testing to verify the functionality of the software. Examples of test methods, unit tests, integration tests, system tests, performance tests, etc. Each test method has its own guidelines, strategies, and results.

9.4.1 Black Box Testing

Black-box testing is a software testing method that tests the functional characteristics of a software product without knowledge of the internal code structure, internal paths, and processes. Black-box testing focuses primarily on input and output flows. Also known as a behavioural test.

Fig 9.1 Diagrammatic representation of Black Box Testing

Black Box Test Type:

Functional Testing: This test involves the process of testing software-related functional requirements.

Non-functional testing: This test includes testing for non-functional requirements such as performance and scalability.

Regression Testing: Regression test includes a code fix test to see if the code fix is affecting the normal implementation.

Black-box testing focuses on verifying software performance without examining the internal structure or code. The main source of information for black-box testing is the specifications of customer requirements. A method in which a tester selects a function, assigns an input value, tests its performance, and confirms that the function returns the expected result. If the function returns the correct result, it passes the test. Otherwise, it will fail. The evaluation team reports the results to the development team and evaluates the next action. After all, the work is being evaluated, and if a serious problem occurs, it will be returned to the development team for repair.

9.4.2 Grey Box Testing

Grey box testing is a combination of software testing that combines a black-box testing strategy with a white-box testing method. In a black-box testing procedure, the tester is unaware of the internal structure of the object under test. The white-box test provides the tester with the internal code structure in advance. The internal structure is partially known to testers from grey box experiments. This includes access to internal data structures and algorithms for developing test cases. The name is Grey Box Test because the software system looks like an invisible grey box. This usually focuses on contextual errors related to web systems.

Fig 9.2 Representation of Grey Box Testing

Grey Box Test Tip:

Matrix Testing: Matrix testing strategies assess business and technical risks identified by software application developers. The engineer considers all variables in the system. Each of these variables carries its own technical and business risks and can be used at different stages of the life cycle.

Pattern test: Analyse past errors and run the test. View the code to determine the cause of the error. The analysis templates provided cover the cause of the error. This helps with design constraints as it is busy catching some errors before production.

Orthogonal array test: This is a way to check the black box. In orthogonal array tests, there is no tolerance and combination of test data. Orthogonal sequence tests are selected when more advanced installations are required, and when the test phase covers very few test cases and the test data is large. This is very useful when testing complex applications.

Retest: Every time you change the software, retest the software to make sure that the changes or new features do not affect your existing system performance. Backtesting is also performed to ensure that the bug 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 structure, code format, and software features work the same as black-box testing. Also known as structural test, clear box test, or glass box test.

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

Testing non-functional attributes is a non-functional test. In this test, the program has non-functional specifications such as user interface, stability, and performance.

9.5.1.1 Performance Testing

Performance testing is a software evaluation process used to evaluate speed, response time, stability, reliability, measurement, and software system performance under certain features. The main purpose of performance evaluation is to identify and remove obstacles to the performance of your application. This is a subset of operational engineering and is also known as "complete testing". The performance test focuses on testing software systems.

Speed Determines if the app responds quickly.
Scalability Specifies the maximum load that an application can support.
Stability determines if an application is stable under different loads.
Many users visited the site and tried to check the recovery speed.

9.5.1.2 Stress Testing

Pressure testing is the process of determining the performance of a computer, network, system, or device to maintain a specified level of performance under adverse conditions. Procedures may include laboratory capacity tests such as: Measurement of error rate or system failure rate.

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

This is the technique used to identify usability flaws. This is achieved by getting a small group of target end users to use the software, and these tests focus on application usability and application flexibility. This application was originally provided to a group of three for usability testing. They were asked to make sure the application was easy to use and clear.

9.5.2 Functional Testing

These tests are used to validate the software system against specifications. This test tests each feature of the software and validates the output against functional requirements. It mainly includes black box testing and checks if an error is displayed according to the error condition.

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 tests are usually done to test the accuracy of isolated code. White-box testing is used as part of unit testing. This is often used by developers themselves to see if there is a problem with their code. The main purpose of these tests is to analyse and fix these errors. Unit tests for dementia applications are written in Python using various methods such as mocking and asserting.

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

model accuracy
train
test
accuracy
1.0
0.9
0.8
0.7
0.6
0.5
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 examinations had to be analysed and further extensively examined by image processing. Further use of mini-mental state assessment data can be used to accurately classify dementia. In addition, the available data can be used to treat patients more effectively and focus on underprivileged areas. In addition, identifying and understanding the patterns between the data can prevent the inheritance of the syndrome to some 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.

Plag_report_doc.pdf

ORIGINALITY REPORT

18%	13%	10%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ncbi.nlm.nih.gov Internet Source	2%
2	deepai.org Internet Source	1%
3	link.springer.com Internet Source	1%
4	Herlind Megges, Silka Dawn Freiesleben, Natalie Jankowski, Brigitte Haas, Oliver Peters. "Technology for home dementia care: A prototype locating system put to the test", Alzheimer's & Dementia: Translational Research & Clinical Interventions, 2017 Publication	<1%
5	www.javatpoint.com Internet Source	<1%
6	Gloria Castellazzi, Maria Giovanna Cuzzoni, Matteo Cotta Ramusino, Daniele Martinelli et al. "A Machine Learning Approach for the Differential Diagnosis of Alzheimer and	<1%

Vascular Dementia Fed by MRI Selected Features", Frontiers in Neuroinformatics, 2020
Publication

7	aquila.usm.edu Internet Source	<1 %
8	www.science.gov Internet Source	<1 %
9	www.spiedigitallibrary.org Internet Source	<1 %
10	José Luis Molinuevo, Nina Gramunt, Juan Domingo Gispert, Karine Fauria et al. "The ALFA project: A research platform to identify early pathophysiological features of Alzheimer's disease", Alzheimer's & Dementia: Translational Research & Clinical Interventions, 2016 Publication	<1 %
11	www.bioscience.org Internet Source	<1 %
12	Submitted to Sreenidhi International School Student Paper	<1 %
13	www.researchgate.net Internet Source	<1 %
14	www.hindawi.com Internet Source	<1 %

coursetaught.com

15	Internet Source	<1 %
16	Sulantha Mathotaarachchi, Tharick A. Pascoal, Monica Shin, Andrea L. Benedet et al. "Identifying incipient dementia individuals using machine learning and amyloid imaging", Neurobiology of Aging, 2017 Publication	<1 %
17	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), 2017 Publication	<1 %
18	huggingface.co Internet Source	<1 %
19	Daniel Welsh, Kellie Morrissey, Sarah Foley, Roisin McNaney, Christos Salis, John McCarthy, John Vines. "Ticket to Talk", Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18, 2018 Publication	<1 %
20	ukdiss.com Internet Source	<1 %
21	Submitted to Glasgow Kelvin College Student Paper	<1 %

22	V K G Kalaiselvi, D Pushgara Rani, Dharini V K, Anupama Jeyashree S, Parvathi Priya Nandana K M. "Ghajini - An application For Dementia Patients", 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), 2022 Publication	<1 %
23	Submitted to Fareham College Student Paper	<1 %
24	cyberleninka.org Internet Source	<1 %
25	diseasesdic.com Internet Source	<1 %
26	Submitted to Curtin University of Technology Student Paper	<1 %
27	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1 %
28	www.guru99.com Internet Source	<1 %
29	www.scie.org.uk Internet Source	<1 %
30	Submitted to University of Sheffield Student Paper	<1 %
31	doi.org Internet Source	

SHOW AND TELL
PANDA PUNNAIAH SQUARE