

A Framework for AI-Driven Neurorehabilitation Training The profiling challenge

MASTER'S DEGREE PROJECT

Pedro Alexandre Gomes Rodrigues

MASTER IN INFORMATICS ENGINEERING

A Framework for AI-Driven Neurorehabilitation Training

The profiling challenge

MASTER'S DEGREE PROJECT

Pedro Alexandre Gomes Rodrigues

MASTER IN INFORMATICS ENGINEERING

ORIENTATION

Eduardo Leopoldo Fermé

CO-ORIENTATION

Sergi Bermúdez I Badia



A Framework for AI-Driven Neurorehabilitation Training: The Profiling Challenge

Pedro Alexandre Gomes Rodrigues
2022



A Framework for AI-Driven Neurorehabilitation Training: The Profiling Challenge

Masters project presented by

Pedro Alexandre Gomes Rodrigues

Abstract

Cognitive decline is a common sign that a person is ageing. However, abnormal cases can lead to dementia, affecting daily living activities and independent functioning. It is a leading cause of disability and death. Its prevention is a global health priority.

One way to address cognitive decline is to undergo cognitive rehabilitation. Cognitive rehabilitation aims to restore or mitigate the symptoms of a cognitive disability, increasing the quality of life for the patient. However, cognitive rehabilitation is stuck to clinical environments and logistics, leading to a suboptimal set of expansive tools that is hard to accommodate every patient's needs.

The BRaNT project aims to create a tool that mitigates this problem. The NeuroAlreh@b is a rehabilitation tool developed within a framework that combines neuropsychological assessments, neurorehabilitation procedures, artificial intelligence and game design, composing a tool that is easy to set up in a clinical environment and accessible to adapt to every patient's needs.

Among all the challenges within NeuroAlreh@b, one focuses on representing a cognitive profile through the aggregation of multiple neuropsychological assessments. To test this possibility, we will need data from patients currently unavailable.

In the first part of this master's project, study the possibility of aggregating neuropsychological assessments for the case of Alzheimer's disease using the Alzheimer's Disease Neuroimaging Initiative database. This database contains a vast collection of images and neuropsychological assessments that will serve as a baseline for the NeuroAlreh@b when the time comes.

In the second part of this project, we set up a computational system to run all the artificial intelligence models and simulations required for the BRaNT project. The system allocates a database and a webserver to serve all the required pages for the project.

Resumo

O declínio cognitivo é um sinal comum de que uma pessoa está a envelhecer. No entanto, casos anormais podem levar à demência, afetando as atividades diárias e funcionamento independente. Demência é uma das principais causas de incapacidade e morte. Fazendo da sua prevenção uma prioridade para a saúde global.

Uma forma de lidar com o declínio cognitivo é submeter-se à reabilitação cognitiva. A reabilitação cognitiva visa restaurar ou mitigar os sintomas de uma deficiência cognitiva, aumentando a qualidade de vida do paciente. No entanto, a reabilitação cognitiva está presa a ambientes clínicos e logística, levando a um conjunto sub-ideal de ferramentas com custos elevados e complicadas de acomodar as necessidades de cada paciente.

O projeto BRaNT visa criar uma ferramenta que atenuie este problema. O NeuroAlreh@b é uma ferramenta de reabilitação desenvolvida num quadro que combina avaliações neuropsicológicas, reabilitação, inteligência artificial e design de jogos, compondo uma ferramenta fácil de adaptar a um ambiente clínico e acessível para se adaptar às necessidades de cada paciente.

Entre todos os desafios dentro de NeuroAlreh@b, foca-se em representar um perfil cognitivo através da agregação de múltiplas avaliações neuropsicológicas. Para testar esta possibilidade, precisaremos de dados de pacientes, que atualmente não temos.

Na primeira parte do projeto deste mestrado, vamos testar a possibilidade de agregar avaliações neuropsicológicas para o caso da doença de Alzheimer utilizando a base de dados da Iniciativa de Neuroimagem da Doença de Alzheimer. Esta base de dados contém uma vasta coleção de imagens e avaliações neuropsicológicas que servirão de base para o NeuroAlreh@b quando chegar a hora.

Na segunda parte deste projeto, vamos criar um sistema informático para executar todos os modelos e simulações de inteligência artificial necessários para o projeto BRaNT. O sistema também irá alocar uma base de dados e um webserver para servir todas as páginas necessárias para o projeto.

Acknowledgements

This project would not be possible without the help and orientation of Eduardo Fermé and Sergi Bermúdez i Badia.

A special thanks to Harry Vasanth for helping how to choose the server operating system and teaching me most of the security procedures the server required. Yuri Almeida, for helping me with mental support and incentivation to continue. Ana Lúcia Faria, for all the patience in helping to understand and select all the data fields present in the various datasets. All the BRaNT team for all the fun and challenges.

Lastly, I want to thank my family and friends for pressuring me to continue.

Contents

List of Figures	xiv
1 Introduction	1
1.1 Cognitive Decline	1
1.2 Cognitive Rehabilitation	4
1.3 This masters contribution	5
1.3.1 Dissertation plan	6
2 State of the Art	9
2.1 Machine Learning in Healthcare	9
2.1.1 Guttman Neuropersonal Trainer	11
2.1.2 Neuro-World	13
2.1.3 Brain Training System	16
2.2 Machine Learning	16
2.2.1 Machine Learning Algorithms	18
2.3 Computational Tools for Cognitive Neurorehabilitation	28
2.4 Infrastructure Implementation	30
2.4.1 Ubuntu Server	30
2.4.2 VMware Server	30
2.4.3 Hypervisors	30
2.5 Tools for Development	34
3 Development	37
3.1 BRaNT	37
3.1.1 The Framework and its Challenges	39
3.2 Contribution to the BRaNT challenges	42
3.3 Server Deployment	43
3.3.1 Hardware available for BRaNT	43
3.3.2 Troubleshooting	45
3.3.3 Webserver	46

4	Results and Discussion	51
4.1	Data Treatment	52
4.2	Profiling	53
4.3	Joining Data	55
4.4	CleanData	57
4.5	Classification	60
4.6	Dealing with missing data: Data reconstruction	61
5	Conclusion	69

List of Figures

1.1	Progression of Alzheimers Disease [8]	3
2.1	Multi-level pie chart of machine learning approaches and data types used for stroke prevention, diagnosis, treatment, and prognostic [17]	10
2.2	(a) Working memory task example. (b) Sustained attention task example [18]	12
2.3	Schematic representation on how the patient performance in NeuroWorld estimates the Mini-Mental Examination Score via a datadriven using on machine learning algorithms [19].	13
2.4	Screenshots of the six games used in the Neuro-World study to evaluate various cognitive impairment dimensions and estimate the Mini-Mental State Examination categories. In this example, the Neuro-World interface is Korean. However, each piece of information contains English translations manually added to the screenshots to help readers understand the written information provided in the games [19].	14
3.1	NeuroAlreh@b proposed framework	38
3.2	List of Graphics cards connected to the host machine	48
3.3	Remapping compatibility and adding modules required for pass-through	48
3.4	PCI devices listed to add to a virtual machine	49
4.1	Detailed information about the files used	52
4.2	ADNI MERGE heatmap for correlation all vs all above 60% and pValue below 5%	54
4.3	ADNIMERGE principal component analysis	55
4.4	Diagnosis present in the CleanData dataset	57
4.5	PCA to ten components for CleanData	58

4.6	Correlation between CleanData and ICA from CleanData	59
4.7	Baseline classification for CleanData	60
4.8	CleanData's ICA classification	61
4.9	Missing values per sub-dataset from the rebuilt dataset	62
4.10	Reconstructed dataset and the subsets with N or more missing values rebuilt	64
4.11	Performance of each algorithm on the vast number of datasets .	65
4.12	Performance per interaction on reconstructed data with independent component analysis	67

Chapter 1

Introduction

The population in the developed world has experienced a significant increase in life expectancy over the last 50 years [1]. This increase in life expectancy led to age-related diseases and conditions. As an example of such conditions, we have cognitive decline, leading to dementia and later Alzheimer's disease [2]. Another example is the increased risk of cardiovascular disorders and strokes.

Cardiovascular disorders and strokes may lead to brain injury, contributing to cognitive decline and dementia. These symptoms challenge a person's daily life and independence. For this reason, we need to find a way to decrease these difficulties.

1.1 Cognitive Decline

Along with the ageing process, our ability to retain new information decreases, our attention gets scattered, and our mental processing and performance get reduced [3, 4]. These problems are part of a natural cognitive decline process.

In a natural cognitive decline, a person can function well. Daily life activities are accomplishable without outstanding issues. Their ability recognition, intelligence or long-term memory are not affected. Occasionally names, words or even things can be misplaced or forgotten for a short period. In some cases, a person can forget more frequently, signify of a new condition named mild cognitive impairment [3, 4].

Mild cognitive impairment is a condition where an individual experiences a noticeable decline in his mental abilities compared with others of the same age. This decline is noticeable by the person experiencing them or others who interact. When comparing mild cognitive impairment with a normal cognitive

decline, it is noticeable that the person frequently forgets conversations and information such as planned events or appointments on the first condition [3, 4].

In some cases where mild cognitive impairment is caused due to the effects of treatable illness or diseases, a person may recover from this condition. However, in most cases, this condition is a stage between normal cognitive decline and early-stage dementia, which can often be due to a variety of diseases, such as Alzheimer's or Parkinson's disease [3, 4].

Dementia is a general term. It covers a wide range of specific medical conditions caused by abnormal brain changes that trigger a decline in cognitive abilities. Leading to the impairment of daily life activities and independent functioning. It is a leading cause of disability and death, and its prevention is a global health priority [5].

There are numerous causes of dementia. The most common one is Alzheimer's disease. It can also develop from other conditions, for example, Lewy body, Parkinson's or even vascular problems like a stroke, which is the second most common cause of dementia. People who suffer brain changes caused by multiple types of dementia have mixed dementia [6, 7].

Almost all forms of dementia are treatable. Medication and supportive measures help manage symptoms. However, most types remain incurable or irreversible and treatment results only in modest benefits [6, 7].

With a careful examination of the patient medical history, physical examinations, laboratory tests and the characteristics of thinking, daily functioning and behaviour, doctors can diagnose dementia with a high level of certainty. However, it may be tough to determine the exact type of dementia. Symptoms of each kind may overlap, making it impossible to specify the time of dementia. When this happens, the patient should see a specialist [6].

The most common cause of dementia is Alzheimer's disease. It accounts for 60 to 80% of the cases. Like dementia, a general term used to describe a group of symptoms that affect daily life activities and independence, Alzheimer's is a term used for memory loss and other cognitive abilities severe enough to affect daily life activities and independence. Alzheimer is a progressive disease that is not a part of normal ageing. It affects memory, thinking, learning and organising skills, and there is no stopping nor reversing. Its symptoms keep worsening over time [8, 9, 10].

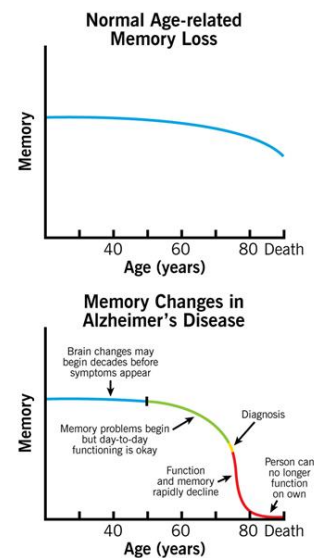


Figure 1.1: Progression of Alzheimer's Disease [8]

Alzheimer's disease most significant risk factor is age. It starts with a minor memory loss which seems to be related to the normal ageing process. However, it develops faster than it should, which leads to the diagnosis of mild cognitive impairment. As this disease advances through the brain, tasks start to feel harder to accomplish. Symptoms including disorientation, mood and behaviour changes increase, leading to confusion about events, time and place. In the last stage of Alzheimer's, the patients start to develop difficulty speaking, swallowing and walking, which leads to death [8, 9, 10]. Figure 1.1 compares an average cognitive decline with Alzheimer's disease [8].

The second most common kind of dementia is vascular dementia. Vascular dementia occurs after a stroke or other conditions that decrease blood flow to the brain. Vessels that carry nutrients and oxygen are crucial for brain functioning. If these are broken or even partially or entirely blocked, oxygen and nutrients will cease to reach certain parts of the brain leading to the death of brain cells [11, 12].

When the brain is damaged, it can not recover. However, some treatments manage the controllable factors to prevent and slow down additional damage [11, 12].

1.2 Cognitive Rehabilitation

One way to mitigate the impact of cognitive decline on a person, dementia, and other brain injury disorders is to undergo cognitive rehabilitation therapy. The term rehabilitation describes restoring the effectiveness of a patient's abilities after an illness or injury. Rehabilitation tries to reduce the impact of disabling and handicapping conditions by achieving the highest levels possible, including optimum social integration [13].

A person with cognitive deficits may experience decreased functioning in multiple domains, including attention, self-awareness, memory, reasoning and judgment. Such impairments represent significant obstacles to personal daily living activities. Neuropsychological or cognitive rehabilitation represents a multidisciplinary approach to addressing various cognitive, emotional, psychological and behavioural factors that impact their lives. Here the patient works with health professionals, families and caregivers and is systematically presented with functional therapeutic activities based on assessments and understanding of brain behaviour deficits [13, 14]. From a clinical perspective, cognitive rehabilitation connotes a methodical intervention to aid cognitive and behavioural deficits in patients. The goal is to increase patients ability to perform daily life activities [14].

Therapeutic interventions in this ambit aim to achieve functional changes through reestablishing previously learned behaviour patterns or establishing new patterns of cognitive activity or compensatory mechanisms [14].

A standard method applied in cognitive rehabilitation employs behavioural observation and ratings of human performance in the real world or physical mock-ups of functional environments. Health professionals appoint tasks and evaluate the patients' performance during such tasks. According to the patients' motor and neurocognitive impairments, the health professional responsible can assign a task within a mock-up environment or a controlled, real-world simulation. Building these environments and providing human resources to conduct such evaluations brings substantial economic costs [14]. Also, this approach is limited in systematic control of real-world stimulus challenges and its adaptability to each patients' case [14].

Problem

These tools have improved patient's livess. However, they are not flexible enough to cover the needs of all patients. Interventions require home visits

or dislocations to a clinic, which adds to the therapy's high cost and negatively affects the duration. In sum, most of the tools for cognitive rehabilitation have a high cost, with suboptimal tools lacking adaptability, intensity and duration [2].

1.3 This masters contribution

This master's project is a subproject from Belief Revision applied to Neurorehabilitation Therapy (BRaNT) financed by the FCT [15].

At the start of this master's project, BRaNT had two main problems. The first problem was that it did not have the technical infrastructure to host any database for its studies. The second problem was that it did not have enough patient data to test the possibility of creating patient profiles based on their neuropsychological assessments.

This project is composed essentially of two parts. On the first part, we assembled a server computer to run machine learning algorithms and simulations. This server will also host several databases and websites for the various studies within BRaNT. The second part of this project tests the possibility of aggregating neuropsychological assessments using data science tools and machine learning to create patients' aggregated cognitive profiles.

To create the aggregated cognitive profile, we defined a general formula for aggregating the neuropsychological assessment tools, considering weights for the relation between each tool and cognitive domains and subdomains. Then we aggregated empirically the tools based on neuropsychologists' experience and obtained the first value for the weights. Then we pre-validated the previous aggregation by using correlations obtained from patients with dementia and readjusted the weights. Finally, we defined a machine learning algorithm for future calibrations.

Six neuropsychologists with expertise in assessments had a discussion session to decide how to aggregate all the selected neuropsychological tools. This discussion concludes that since most of the tools' scores and subscores may evaluate different cognitive domains and subdomains, we have divided each tool by the number of sub-scores it involves and the number of cognitive domains and subdomains it targets. For example, the Montreal Cognitive Assessment is a cognitive screening measure that gives eight subsets of information about general cognitive functioning. Each subset corresponds to 12.5% of the total evaluation. Such a multidimensional and comprehensive tool contributes to assessing different domains and subdomains: Calculus, for example, contributes

12.5% to executive functions assessment, and it divides into two subdomains, working memory and sustained attention, each with a weight of 6.25%. For each domain to sum 100%, these values were normalised. For example, if a subdomain has only one tool entry with 50%, we have to convert it to 100%. This way, if one or more scores are missing in one or more tools, the NeuroAlreh@b system will have the ability to normalise scores according to the non-normalised values.

We have come up with two different ways to validate these empirical values. First, we checked if the weighted sum for aggregation of each tool validates the basic aggregation rules. On the second validation, we used machine learning techniques to find correlations between the different neuropsychological assessment tools and compare these correlations with our empirical ones, establishing an analogy. However, the problem here was that we did not have enough data on stroke, mild cognitive impairment and dementia patients to use a machine learning approach.

Since we are building an entirely new tool, we lack data on stroke patients to test our theory of aggregating neuropsychological assessment tools. The Alzheimer's Disease Neuroimaging Initiative database came in handy to test our hypothesis. This initiative is a longitudinal multisite observational study of elderly individuals with normal cognition, mild cognitive impairment and Alzheimer's disease. Since it includes a battery of neuropsychological assessment tools, we used their database to obtain the aggregations for Alzheimer's disease and create an example of how NeuroAlreh@b should approach the problem.

1.3.1 Dissertation plan

As mentioned, BRaNT had two main problems: 1. It did not have an infrastructure ready to deploy the services required to work correctly. And 2. It does not have enough data to test the possibility of creating aggregated cognitive profiles.

Problem 1

For the infrastructure for the BRaNT, we will assemble a computer workstation to act as a server for all the services BRaNT requires.

- First, we will start by gathering information on the best operating system

to use and testing the server stability of this operating system. Then we will deploy all the machines required for the several services.

- Second, find a way to pass through the PCI GPUs to the virtual machine that will use them for the machine learning algorithms and simulations.
- Lastly, we will ask for the help of the university's infrastructure and networks department to allocate us a public IP. This IP is to put the web server online.

Problem 2

The second problem was testing the possibility of aggregating neuropsychological assessments into a cognitive profile. Since BRaNT did not have data on stroke, we will use a database with thousands of individuals to test this possibility.

- We will start by requesting access to ADNI, the database and getting familiar with its data. Then we will prepare the data, translate it from text to number if required and get everything ready and understandable for any algorithm.
- The next step is to run correlations. If there are fields with strong correlations, the data is reduceable. The reduction may be achieved by
- The last goal is to find a way to reproduce missing data on the datasets and see how challenging it is to estimate a missing field. This reconstruction will be helpful to fill data when a patient does not have all the required assessments and may be beneficial for predicting the new results after an intervention.

Chapter 2

State of the Art

This section explains the main concepts used during the development of this project. On the theoretical part, we used mathematical algorithms from machine learning. Furthermore, we reviewed some operating systems on the practical component to find a better solution that would fit our needs.

Before diving into further details, it is essential to expressly state that this project does not focus on clinical or health neurorehabilitation interventions. Instead, the main focus is on how machine learning can contribute to this field. Explore existing tools and how to gather different neuro-assessment tools to create a cognitive profile.

Over the last decades, artificial intelligence capabilities have grown exponentially. In recent years, it has become ubiquitous. It is everywhere, from cars to smartwatches, from smart TVs to the operation room in advanced hospitals. The problem is that the more complex is the artificial intelligence system, the harder it is to explain and understand how it gets to its solutions.

According to John McCarty [16], artificial intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to biologically observable methods. There are many sub-fields within this area. One of them is machine learning.

2.1 Machine Learning in Healthcare

A literature review performed by Manisha Sirsat, Eduardo Fermé and Joana Câmara [17] gathers information about the performance and uses of machine

learning in the case of stroke. Their work reviews state of the art on machine learning techniques for brain stroke, and they classify the research studies into four categories based on their functionalities or similarity. They created a pie chart, Figure 2.1, providing the machine learning approaches and data types used for brain stroke prevention, diagnosis, treatment and prognosis.



Figure 2.1: Multi-level pie chart of machine learning approaches and data types used for stroke prevention, diagnosis, treatment, and prognostic [17]

In the literature, we were able to find three platforms for cognitive rehabilitation that use a machine learning approach to adapt and personalise training sessions: the Guttman Neuropersonal Trainer, the Neuro-World and the Brain Training System.

2.1.1 Guttman Neuropersonal Trainer

The Guttman Neuropersonal Trainer [18] is a telerehabilitation platform integrated into clinical routine rehabilitation centres. It addresses the rehabilitation of patients with cognitive impairments using advanced technologies and knowledge grounded on cognitive neuroscience, plasticity, and neuropsychology. This platform enables individual and personalised treatments, improving the traditional on-site rehabilitation process. The development of the Guttman Neuropersonal Trainer had the goal of improving the limitations of traditional face-to-face rehabilitation while managing, registering, and monitoring treatments to increase the efficiency of the process. The development of this platform supports a user-centred and model-based design methodology. The rehabilitation process defined in this platform starts by assigning a patient to a therapist responsible for the treatment. The therapist has then to perform the initial neuropsychological assessment, consisting of a set of validated tests used to evaluate cognitive functions prior to the treatment. The resulting scores calculate a cognitive profile and get stored in the system as a baseline. This profile gives the therapist relevant information to support their treatment decision.

The rehabilitation treatments consist of 3 to 5 sessions per week for a total of 60 sessions. After defining a rehabilitation session, the patient executes the assigned tasks, sending its performance back to the server. This way, the therapist can check on the patients' performance and adjusts the treatments to their evolution [18].

After completing the treatment, the therapist performs a new neuropsychological assessment, comparing it to the first one and determining the improvement of the patients' cognitive capacities [18].

The Guttman Neuropersonal Trainer consists of computerised cognitive exercises covering different cognitive functions and subfunctions. Neuropsychologists from the Guttman Institute designed every task according to cognitive paradigms to address specific cognitive subfunctions to better personalise the treatment according to the patients' specific needs [18].

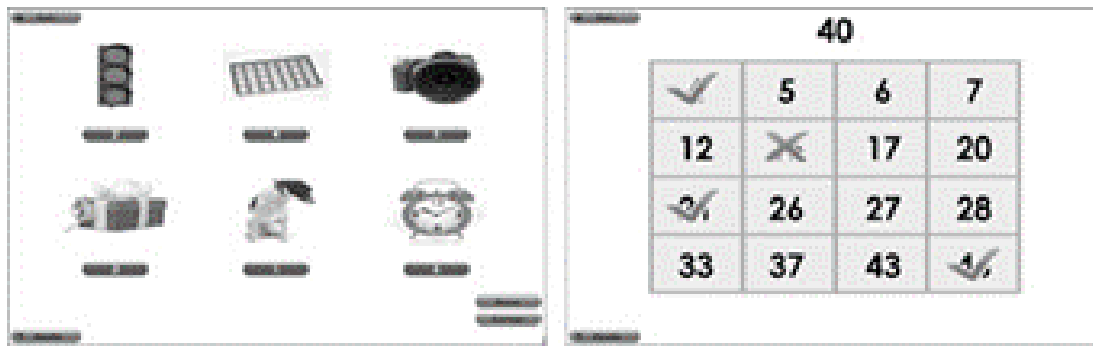


Figure 2.2: (a) Working memory task example. (b) Sustained attention task example [18]

The platform comprises 95 computerised cognitive training exercises that target several cognitive functions and sub-functions. A score relative to that task is calculated when a task is complete. This score is a value between 0 and 100 and gets stored for a later evaluation. It counts with a universal and accessible interface, which is vital in every telemedicine platform. Figure 2.2 shows design examples for two tasks designed to address a cognitive function. Figure 2.2 (a) shows a task with the primary goal of addressing working memory. The patient is presented with a sequence, and after a delay, he must press the elements in the same order. Figure 2.2 (b) contains a task focused on sustained attention. Here the patient is given a set of numbers and must find the number shown at the top of the screen [18].

2.1.2 Neuro-World

Neuro-World [19] consists of six mobile games designed to challenge visuospatial short-term memory and selective attention. This approach allows self-administer the assessment and longitudinally monitoring of one's cognitive impairment level in remote settings. To develop Neuro-World, researchers found a potential relationship between the basic cognitive processes that underlie the categories of Mini-Mental State Examination.

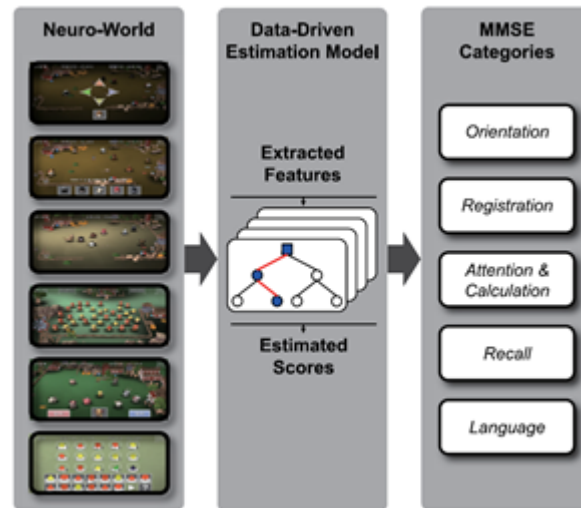


Figure 2.3: Schematic representation on how the patient performance in NeuroWorld estimates the Mini-Mental Examination Score via a datadriven using on machine learning algorithms [19].

The goal is to challenge the patients using basic cognitive processes that collectively affect the patient's responses to a set of mobile games. Then feed the results to an algorithm to identify the relationship between the patient's game performance and the Mini-Mental State Examination. Figure 2.3 represents the conceptual connections between the Neuro-World games and the Mini-Mental State Examination scores [19].



Figure 2.4: Screenshots of the six games used in the Neuro-World study to evaluate various cognitive impairment dimensions and estimate the Mini-Mental State Examination categories. In this example, the Neuro-World interface is Korean. However, each piece of information contains English translations manually added to the screenshots to help readers understand the written information provided in the games [19].

In Figure 2.4, we can observe all six games from Neuro-World [19]. Here is a small presentation for each of the games:

- **Game 1** contains a flock of animals. At some point, one animal leaves the farm, and the objective is that the patient selects the direction of where the animal is. As the patient proceeds to higher stages, the flock becomes more extensive and moves faster [19].
- **Game 2** starts with a herd of animals moving into the farm one after the other. A subset of the pack will move on, leaving the farm. The patient must remember the animals that just left and select them among the present images at the bottom of the screen. As the game evolves, animals' numbers and speeds increase [19].
- **Game 3** starts with animals coming to the farm one after the other, following a particular order. At some point, a subset of these animals leaves the farm. The patient must remember that the remaining animals have entered the farm and select them in the correct order. In higher stages, the size, the moving speed and the remaining animals increase [19].
- **Game 4** consists of animals with different visual characteristics displayed on the screen. An auditory instruction provides a graphic description of the animals that the patient must find and count. The patient must input the correct count of the animals described. In higher stages, the total of animals increases and the description becomes more complex [19].
- **Game 5** presents a new flock of animals. The patient must find an animal shown on a separate box at the bottom of the screen and select found or not found accordingly. As stages increase, the characteristics of animals increase their complexity and the number of animals present increases [19].
- **Game 6** contains a series of targets, combining multiple primitive 3D shapes and colours representing a particular pattern. The patient must understand the sequence in which the primitive shapes appear. Then, the patient has to identify the shapes that follow the presented sequence of shapes. In higher stages, the type of primitive shapes and the sequential pattern become more complicated [19].

2.1.3 Brain Training System

Maintain Your Brain [20] is a randomised controlled trial of a multi-modal digital health intervention targeting modifiable dementia risk factors to combat cognitive decline and prevent dementia. A key computerised cognitive training module is the Brain Training System within this trial.

Brain Training System contains exercises provided by a partnership with NeuroNation as 34 stand-alone activities with their internal logic and tuneable parameters. This module introduces a new approach to computerised cognitive training using logic built around the participants' initial cognitive profile, which gets updated during the evolution of their training process. Brain Training System counts with a scoring system based on each exercise performance. This scoring system allows comparing the performance from different exercises merged to the same cognitive domain level. The scoring system serves three primary purposes: Update the patients' cognitive profile during the training process, provide performance feedback to the patient and automatically identify the user engagement, compliance, or adherence issues for supervisory redress using a "red flag" system [20].

Another innovation of the Brain Training System was to entangle the user's training experience with their real-world social network. Training with Friends functionality allows the participant to interact with an online trainer using online messaging, video-conferencing and phone calls [20].

In other words, Brain Training System aims to maximize cognitive rehabilitation by implementing online supervision and a novel algorithm that automatically selects and schedules mental training exercises. The system uses the participants' cognitive profile to calculate the difficulty level of the practices and keeps updating this difficulty as the participant progresses. The Brain Training System is currently under evaluation in an online personalised intervention to reduce cognitive decline in an older age cohort [20].

2.2 Machine Learning

In this section, we explain what machine learning is and talk about how it works, its algorithms and its appliances. Note that we only used supervised learning to classify the diagnosis present on the database for the creation of cognitive profiles. Most of the information here comes from a survey by Iqbal H. Sarker [21], where he explains many machine learning algorithms and their field of use.

For more information, please read his paper.

Machine learning represents computer algorithms that, in a sense, try to emulate human intelligence [17]. The architecture of these algorithms automatically mutates or adapts through repetition to become better and better at achieving the desired task [22, 23, 21]. Machine learning builds on expertise from diverse fields. These fields are: artificial intelligence, probability and statistics, computer science, information theory, and cognitive neuropsychology. In sum, machine learning represents a group of algorithms with the distinctive ability to learn from input data, with or without a teacher [23].

We can categorise machine learning into transductive and inductive learning from a concept learning perspective. Transductive learning involves the inference from specific training cases to specific testing cases using discrete labels as in clustering or continuous labels as in manifold learning. On the other hand, inductive learning aims to predict outputs from inputs that the learner has not encountered before. From a probabilistic perspective, we have discriminant or generative models. A discriminant model measures the conditional probability of output given deterministic inputs. A generative model is fully probabilistic, whether it uses a graph modelling technique or not [23].

There are four methods for machine learning: Supervised learning, unsupervised learning, semi-supervised learning and reinforced learning . Machine learning algorithms get categorised within these methods according to the nature of the data, their learning process and the model type [23].

Supervised learning trains a model that maps an input to an output based on observations and predicts the outcome [17]. Algorithms within supervised learning use labelled examples to predict future events. This method uses a dataset to train a model to infer a function to predict the output values [22, 24, 25]. After sufficient training, it can provide targets for new inputs. The learning algorithms can also compare its output with the correct one in search of errors to modify the model accordingly, improving its accuracy [22, 24]. Algorithms within supervised learning are for classification and regression. Classification is to classify discrete target variables using predictors. Regression investigates the relationship between numerical target variables and predictors [17].

Unsupervised learning clusters the observations according to how similar they are [17]. This approach investigates how to derive a function from characterizing the dataset's structures without human intervention [22]. Unsupervised learning analyzes datasets that have not been tagged. The main

applications of this approach include generative feature extraction, relevant trend and structure detection, result grouping, and experimental reasons. Clustering, density estimation, feature learning, dimensionality reduction, association rule discovery, anomaly detection, and other processes are among the most popular tasks [24].

Semi-Supervised learning: When working with labelled and unlabelled data, semi-supervised learning is useful [22]. The main objective of this approach is to improve upon a prediction made using only the model's labelled data. The main uses of semi-supervised learning are text categorization, fraud detection, machine translation, and data labeling [24].

Reinforcement learning interacts with its environment by producing actions and discovering errors or rewards. The most relevant characteristics of reinforcement learning are the delayed reward and the trial and error search [22, 25]. This method allows systems to determine the ideal behaviour within a specific context to maximise performance. Simple reward feedback, known as a reinforcement signal, is required for the agent to learn which action is the best [22]. It is an effective technique for building artificial intelligence models that can boost automation or improve the effectiveness of complex systems like robots, autonomous driving, production, and logistics in the supply chain. [26, 21]. However, not useful for solving fundamental or straightforward problems.

2.2.1 Machine Learning Algorithms

Supervised, unsupervised, reinforced, and semi-supervised learning are the base methods for many algorithms. Here we describe some algorithms as an example for each of these methods:

Supervised Learning

As previously said, algorithms in this approach are typically for classification and regression. Classification is a predictive modeling task in which a class label is predicted for a given example. Predicting the class of a given set of data points may be done using either structured or unstructured data [24]. Here are some classification examples:

- **The Naïve Bayes:** algorithm came from the Bayes theorem, assuming independence between each pair of features [27, 25, 28]. We can use this algorithm with binary and multi-class categories in many real-world

situations, such as document or text classification, spam filtering, and others. To effectively classify the noisy instances in the data and construct a robust prediction model, the naïve Bayes classifier comes in handy. The key benefit is that it needs a small amount of training data to estimate the necessary parameters compared to more sophisticated approaches [29]. However, its performance may be affected due to its strong assumptions about feature independence. Gaussian, Multinomial, complement, Bernouli and Categorical are the common variants for naïve Bayes classifier [30].

Linear Discriminant Analysis .

- **Linear Discriminant Analysis** is a decision boundary classifier created by fitting class conditional densities to data and using Bayes' rule. It's also known as Fisher's linear discriminant generalisation since it projects a dataset into a lower-dimensional space, decreasing dimensionality and hence model complexity and computation costs. The standard linear discriminant analysis approach typically assigns each class a Gaussian density, assuming that all categories have the same covariance matrix. Linear discriminant analysis, like variance and regression analysis, aims to express one dependent variable as a linear combination of other traits or measurements [27, 30].
- **Logistic regression** is another statistical model used to solve classification issues. This algorithm typically uses a logistic function to estimate the probabilities of an event. It can overfit high-dimensional datasets and works well when the dataset can be separated linearly. To avoid overfitting, we may need regularisation techniques. A significant drawback of this algorithm is the assumption of linearity between the dependent and independent variables. Despite being used mainly for classification, this algorithm can also solve regression problems [30, 31].
- **K-Nearest Neighbors** is an "instance-based learning" or non-generalising learning, also known as the "lazy learning" algorithm. Instead of constructing a general internal model, it stores all instances corresponding to training data in n-dimensional space. Later it computes the classification from a simple majority vote of the k-nearest neighbours of each point. The accuracy typically depends on the quality of the data, and this technique is resilient to noisy training data. The biggest issue with using K-Nearest

Neighbours is choosing the optimal number of neighbours. This algorithm can solve classification and regression problems [32].

- **Support vector machine** is a common technique used for classification, regression, or other tasks. This algorithm constructs a hyperplane or set of hyperplanes in a high dimensional space. Intuitively, the hyperplane, which has the most significant distance from the nearest training data points in any class, achieves a strong separation since, in general, the greater the margin, the lower the classifier's generalisation error. It is effective in high-dimensional spaces and can behave differently based on different mathematical functions known as the kernel. Linear, polynomial, radial basis function, sigmoid and others are kernel functions used in support vector machine classifiers. This algorithm does not perform well when the dataset contains more noise, such as overlapping target classes [33].
- **Decision Tree:** Here, we have a well-known algorithm used in classification and regression tasks. The decision tree algorithm is robust for dealing with user behaviour and cybersecurity analytics domains. It splits the population into various homogeneous sets based on the most significant attributes and independent variables. A decision tree structure comprises a root node connecting to leaf nodes. The split happens from the root moving down each branch to the attribute value. The most popular criteria for splitting data in a decision tree are "gini" for the Gini impurity and "entropy" for the information gain [34, 35].
- **Random Forest:** As the decision tree, random forest is a well-known classification algorithm used in various applications. This algorithm uses "parallel ensembling", which fits several decision tree classifiers in parallel on different dataset sub-samples and uses the majority voting or averages for the outcome or final result. It thus minimises the over-fitting problem and increases prediction accuracy and control. Therefore, the random forest learning model with multiple decision trees is typically more accurate than a single decision tree-based model. It combines bootstrap aggregation and random feature selection to build a series of decision trees with a controlled variation. It is adaptable to classification and regression problems while fitting well for categorical and continuous values [36, 37, 38].
- **Adaptative Boosting:** This ensemble learning process employs an

interactive approach to improve poor classifiers by learning from their errors. Unlike random forest, which uses parallel ensembling, adaptive boosting uses sequential ensembling. It creates a robust classifier by combining many poorly performing classifiers to obtain a classifier of higher accuracy. This algorithm is called an adaptive classifier by significantly improving the model's efficiency but can trigger overfits. Despite its sensitivity to noisy data and outliers, an adaptive boost can boost the performance of decision trees based on estimators on binary classification problems [39].

- **Extreme gradient boosting:** Similar to the random forest, the ensemble learning algorithm gradient boosting creates a final model from a collection of individual models, often decision trees. This algorithm uses a gradient to minimise the loss function, similar to how neural networks use gradient descent to optimise weights. Extreme gradient boosting prevents over-fitting and enhances model generalization and performance by computing second-order gradients of the loss function to minimize loss. This algorithm is fast to interpret and can handle large-sized datasets well [24, 36].
- **Stochastic gradient descent:** This is an iterative algorithm used for optimizing objective functions with appropriate smoothness properties, where the word 'stochastic' refers to random probability. By allowing for quicker iterations in exchange for a reduced convergence rate, the objective is to lessen the computational strain, particularly in high-dimensional optimisation problems. The slope of a function used to determine how much one variable would change in response to the changes in another is known as the gradient. Gradient descent is a convex function that, mathematically speaking, produces a partial derivative of a collection of its input parameters. This algorithm was applied successfully to problems often encountered in text classification and natural language processing. The main problem with stochastic gradient descent is its sensitivity to feature scaling and needs a range of hyper-parameters, such as the regularisation parameter and iterations [24].
- **Rule-based classification:** The rule-based term usually refers to a classification scheme that uses a set of IF-THEN rules for class prediction. We can find several classification algorithms implementing this, such as Zero-R, One-R, decision trees, and many others. The most common

is the decision tree because it has several advantages, such as ease of interpretation, simplicity, speed, good accuracy, and producing precise rules for a human to understand the classification. The decision tree-based also rules significantly improve prediction models for unseen test cases [24].

On the other hand, we also have regression within the supervised learning method, which allows predicting a continuous result variable based on the value of one or more predictor variables [24]. The most significant distinction between classification and regression is that classification predicts distinct class labels, while regression facilitates the prediction of a continuous quantity. There are some overlaps between the two types of algorithms. In many sectors today, including financial forecasting or prediction, cost estimating, trend analysis, marketing, time series estimation, medication response modeling, and many others, regression models are frequently utilized [21, 30]. Now we describe some of the algorithms used for regression:

- **Simple and multiple linear regression:** This is a well-known regression technique and one of the most used machine learning modeling strategies. Here the dependent variable is continuous, the independent variables can be continuous or discrete, and the regression line is linear. This algorithm creates a relationship between the dependent variable and one or more independent variables using the best fit straight line. A variation of simple linear regression is multiple linear regression. Unlike simple linear regression, which contains only one independent variable, it enables the modeling of a response variable as a linear function using two or more predictor variables [24, 30].
- **Polynomial regression:** This is a peculiar case of multiple linear regression, which estimates the relationship as an n th degree polynomial [30, 40]. This algorithm is sensitive to outliers, so one or two outliers can affect the performance badly [30].
- **Lasso and ridge regression:** These are well known as powerful techniques typically used for building models for many features due to their capability of preventing overfitting and reducing the complexity of the model. LASSO means *least absolute shrinkage and selection operator*. This model uses the L1 regularization technique that uses shrinkage, penalising the absolute value of magnitude coefficients. As a result, LASSO appears

to render coefficients to absolute zero. Thus, LASSO regression aims to find the subset of predictors that minimizes the prediction error for a quantitative response variable. On the other hand, ridge regression uses L2 regularization, which penalises the squared magnitude of coefficients. Thus, ridge regression forces the weights to be small but never sets the coefficient value to zero and does a non-sparse solution. Overall, LASSO regression helps obtain a subset of predictors by eliminating the less important ones. Furthermore, ridge regression is helpful for a dataset where the predictors correlate with each other [30].

Unsupervised Learning

As an example of algorithms within unsupervised learning, we will focus on clustering analysis, dimensionality reduction and feature learning, and Association rule learning. Clustering analysis is a technique used for identifying and grouping related data points in large datasets without concern for a specific outcome [21]. Some clustering analysis algorithms are:

- **K-means clustering:** Here, we have a simple and robust algorithm that provides reliable results when data sets are well-separated from each other. This algorithm allocates data to a cluster so that the squared distance between the data points and the centroid is as tiny as possible. In order to make the centroids as small as feasible, the K-means method first determines the k number of centroids before assigning each data point to the closest cluster. The problem is that it can have inconsistent results since this algorithm begins with a random cluster centre selection, and extreme values can easily affect a mean making this algorithm sensitive to outliers. A variant of K-means, K-medoids, is more robust to noises and outliers [41].
- **Mean-shift clustering:** The number of clusters or restrictions on cluster shape need not be known for this nonparametric clustering technique. Mean-shift clustering aims to discover chunks of data in a smooth distribution or density of samples. This is a centroid-based algorithm. It updates the centroid candidates as the mean of the points in a given region. In the post-processing stage, these candidates are filtered, and near-duplicates get removed. Examples of application domains include cluster analysis in image processing and computer vision. Mean-shift has the disadvantage of being computationally expensive. Also, the mean-shift

algorithm does not work well in cases with high dimensions and where the number of clusters shifts abruptly [42].

- **DBSCAN:** A density-based clustering approach used in data mining and machine learning is called *density-based spatial clustering of applications with noise*. This algorithm separates the high-density clusters from the low-density clusters used in the model building. The fundamental tenet of DBSCAN is that a point belongs to a cluster if it is close to numerous other points from that cluster. In a large volume of noisy data that can contain outliers, it can detect clusters of various shapes and sizes. Unlike k-means, DBSCAN does not require a priori specification of the number of clusters in the data and can find arbitrarily shaped clusters. Although k-means is much faster than DBSCAN, DBSCAN is robust to outliers and efficient at finding high-density regions [43].
- **Gaussian mixture models clustering:** The gaussian mixture model is a distribution-based algorithm often used for data clustering. A Gaussian mixture model is a probabilistic model in which all the data points are generated by a limited number of Gaussian distributions with unidentified parameters. Expectation-maximisation, an interactive method that uses a statistical model to estimate the parameters, is handy for finding the Gaussian parameters for each cluster. The Gaussian mixture model, in contrast to k-means, considers uncertainty and returns the likelihood that a data point belongs to one of the k clusters. Gaussian mixture models clustering is more robust than k-means and works well even with non-linear data distributions [21, 30].
- **Agglomerative hierarchical clustering:** Agglomerative hierarchical clustering is the most common hierarchical clustering method. This agglomerative clustering algorithm groups objects in clusters based on their similarity. It uses a bottom-up approach, where it treats each object as a singleton cluster, then pairs clusters merging them one by one until it merges all clusters into a single large cluster containing all objects. The result is a dendrogram, a tree-based representation of the elements. Some applications of this algorithm are single linkage, complete linkage and bots. The benefit of agglomerative hierarchical clustering over k-means is that the tree-structure hierarchy produced here is more informative than the unstructured group of flat clusters given by k-means, which aids in improved decision-making in the applicable application domains[21, 30].

Unsupervised Learning also supports dimensionality reduction and feature learning which are techniques used to clean data sets before creating models. Here we will describe some algorithms for dimensionality reduction and feature learning:

- **Variance threshold:** This algorithm represents a simple approach to feature selection. The variance threshold eliminates all zero-variance characteristics with the same value in all samples while ignoring the outputs [30].
- **Pearson Correlation:** This is another method to understand a feature selection and helps find an association between features in a dataset. The results produced from this algorithm are a value between -1 and 1, meaning negative correlations and positive correlations, respectively. 0 means that the two variables do not have a linear correlation [24, 44].
- **Analysis of variance:** This statistical tool is used to verify the mean values of two or more groups that differ significantly from each other. This algorithm makes presumptions about the linear relationship between the variables, the target, and the normal distribution of the variables. F tests are used in this algorithm approach to evaluate the quality of means statistically. Then it uses these F tests to replace certain features independently of the variable goal and can be omitted [30].
- **Chi-square:** the chi-square is another statistic algorithm that estimates the difference between the effects of a series of events or variables observed and expected frequencies. This algorithm is practical for testing relationships between categorical variables [30].
- **Recursive feature elimination:** This is a brute force approach to the feature selection algorithm. It fits the model and removes the weakest features before meeting the specified number of features. Recursive feature elimination ranks its features by the coefficient of significance. Then it aims to remove dependencies and collinearity in the model. It recursively removes a small number of features per interaction to achieve this [30].
- **Principal component analysis:** One of the most popular methods in data science and machine learning is this algorithm. It uses a mathematical technique that transforms a set of correlated variables into another

set of uncorrelated variables known as principal components. Principal component analysis can extract features this way, reducing the dataset's dimensionality and building an effective model. This approach, at its core, computes the greatest eigenvalues of a covariance matrix and utilizes those values to project the data into a new subspace with dimensions equal to or less than the original subspace [45, 46].

An exciting relationship inside a dataset can be found with the aid of association rule learning, a rule-based machine learning technique. It has several application areas, including IoT services, medical diagnosis, usage behaviour analytics, cybersecurity, data mining, etc [47]. Here are some examples of association rule algorithms:

- **Apriori:** This algorithm helps generate association rules for a given dataset. This algorithm uses a “bottom-up” approach to generate the candidate itemsets. Apriori uses a property where all subsets of a frequent itemset must be frequent. Therefore, if an item set is not frequent, all its supersets are not frequent. Another approach to predictive apriori is the generation of rules. However, it receives unexpected results by combining variables' support and confidence. This algorithm is one of the most frequently used techniques in rule association mining [48].
- **ECLAT:** *Equivalence Class Clustering and bottom-up Lattice Transversal* is an algorithm that uses a depth-first search to find frequent item sets. In contrast to Apriori, this algorithm represents data in a vertical pattern. Hence, the ECLAT is more efficient and scalable for association rule learning. It is better suited for small and medium datasets, while the Apriori fits better in large datasets [49].
- **FP-Growth:** Another method for learning common association rules that are based on a frequent-pattern tree. The main difference with Apriori is that while generating rules, the Apriori algorithm generates frequent candidate itemsets. On the other hand, FP-Growth prevents candidate generation and thus produces a tree by the successful divide and conquer approach. This algorithm is challenging to use in an interactive mining environment due to its complexity. It would not fit into memory for massive data sets, making it challenging to process big data [50, 51].
- **ABC-RuleMiner:** To offer practical, intelligent services, another rule-based machine learning algorithm seeks out intriguing non-redundant rules.

By taking into account the significance or precedence of the relevant contextual characteristics, this algorithm efficiently detects association redundancy and unearths a set of non-redundant rules. ABC-RuleMiner constructs an association generation tree, a top-down approach and then extracts association rules through traversing the tree. Thus, it is more potent than traditional rule-based methods in terms of both non-redundant rule generation and intelligent decision-making, particularly in a context-aware innovative computing environment where human or user preferences are involved [52].

Reinforcement learning

An agent can learn through trial and error in an interactive environment using information from its actions and experiences using the reinforcement learning technique. In contrast to the other methods, reinforcement learning does not require sample data or instances to learn. Instead, it engages with its surroundings and learns through earning rewards. Typically, this method requires four elements: an Agent, an Environment, a Reward system, and a Policy [53].

We can split reinforcement learning into model-based and model-free techniques. Model-based reinforcement learning is the process of overriding an environment model's recommended behavior by taking actions and evaluating the outcomes. Usually, this requires an immediate reward. The critical difference between model-based and model-free is the policy network required for model-based reinforcement learning but not for model-free [53]. Following there are some examples of reinforcement learning algorithms:

- **Monte Carlo methods:** Monte Carlo techniques or experiments are a broad category of algorithms that rely on repeated random sampling to obtain numerical results. The idea is to, in theory, employ randomization to solve deterministic issues. Optimization, numerical integration, and probability distribution drawings are the three problem classes most commonly implemented with Monte Carlo techniques [26].
- **Q-learning:** This is a model-free reinforcement learning algorithm. It focuses on learning the quality of behaviours that tell an agent which action to take on which conditions. This approach can handle stochastic transitions and rewards without the need for modifications and does not require a model of the environment. Since the method determines the

greatest predicted rewards for a given behavior in a particular state, the "Q" in Q-learning often stands for quality [26].

- **Deep Q-learning:** Deep Q-learning feeds the initial state to a neural network which returns the Q-value of all possible actions as an output. Q-learning works flawlessly when there is a relatively straightforward setting to overcome. As a function approximator, deep learning is utilized when the number of states and actions is excessively complex [26].

2.3 Computational Tools for Cognitive Neurorehabilitation

There are two main categories of cognitive rehabilitation. Conventional rehabilitation, where the therapist use paper and pencil exercises to aid the patient, and computer-assisted rehabilitation. Here the patient uses a computer with a set of tasks that can emulate real-world scenarios and helps with rehabilitation. Cognitive strategies to retain or alleviate the patient's deficits in attention and concentration, visual processing, language, memory, reasoning, problem-solving, and executive functions are the basis for both of these techniques [54].

A recent proliferation of computer-assisted and other multimedia methods was born to aid neurocognitive rehabilitation, reflecting a general trend toward leveraging technology to improve the accuracy and efficiency of data capture procedures. Computerised skill training programs have emerged for various populations [14].

Computer-based cognitive rehabilitation has been an effective intervention since the early 1980s in treating the neurocognitive impairments of patients with brain injury, dementia, or schizophrenia. Computer-based cognitive rehabilitation uses a computer as an intervention tool. This computer will provide feedback on the patients' responses and reaction speed via input devices like a keyboard or a joystick. Later the results of each task are shown on the screen [14].

Virtual reality is a more recent approach to computer-assisted cognitive rehabilitation. Virtual reality consists of informatic technologies that create interactive environments involving the user while simulating the real world. It consists of specific software programs and input-output peripherals that

reproduce complex and immersive experiences. Virtual reality constitutes a new frontier for rehabilitation and presents potential advantages to the rehabilitation team. This new way of rehabilitation potentiates the quality of rehabilitation sessions, as they allow them to propose playful activities, thus increasing motivation and involvement [54].

Virtual reality instruments have many different experimental and clinical applications. They can be effective in different neurological pathologies and used at any age and for several purposes. It can improve impaired functions, stimulate and increase spare capacities, and foster a sense of well-being, improving the patient's level of participation and autonomy [54].

Virtual reality offers several features, such as goal-oriented tasks and repetition, which are essential in neurological recovery. Virtual reality tasks are described as more exciting and enjoyable by both children and adults, thus obtaining more repetitions, with positive results on therapist compliance and patient functional outcomes [54].

Computer-based tools for rehabilitation have shown great promise. They are adaptable and can facilitate rehabilitation in some conditions. With this in mind, and to reduce the costs of conventional rehabilitation, the NeuroRehabLab developed the Task Generator [2]. This tool results from a design approach with 20 rehabilitation professionals. Task Generator can generate paper-and-pencil personalised cognitive rehabilitation tasks using clinical settings parametrized through a participatory design approach. This tool is also free and worldwide accessible for health professionals, making it deployable at healthcare centres at virtually no cost [55].

Another tool is Reh@City. Reh@City is a virtual reality-based tool that improves the lack of ecological validity in paper and pencil tasks. It simulates a city with a three-dimensional environment built with streets, sidewalks, commercial buildings, parks and moving cars. Reh@City enables an integrative and personalised cognitive rehabilitation process, targeting several cognitive domains. Additionally, it enables interaction with a virtual world accessible through its interface and adapts the complexity of the scenarios to the patients' profiles [56].

2.4 Infrastructure Implementation: Operating Systems and Hypervisors

To implement the infrastructure, we required a machine with the ability to serve the project webpage and deal with many other tasks. For this reason, we studied several operating systems that could fit our needs. This section shows some of the relevant operating systems we have considered.

2.4.1 Ubuntu Server

Ubuntu Server [57] is an open-source operating system developed and maintained by Canonical. Ubuntu Server runs on most the architectures, and its main uses are the allocation of web services, databases, email, file and print servers. It is also vital for development and container deployment.

2.4.2 VMware Server

This tool server-virtualisation software is free and allows the system administrator to partition a physical server into multiple virtual machines. VMware Server [58, 59] is compatible with most operating systems.

2.4.3 Hypervisors

A hypervisor, also known as a virtual machine monitor, is software that creates and runs virtual machines. It isolates the hypervisor operating system and resources from the virtual machines and enables the creation and management of those virtual machines. This software treats the host machine resources as a pool that can quickly reallocate between existing guests or new virtual machines [60].

The hypervisor allocates and manages virtual machine resources' scheduling against the physical resources. The hardware still has to do the execution, so the CPU still executes instructions as requested by the virtual machines while the hypervisor manages the scheduling [60].

Besides having sub-systems isolated as virtual machines, a hypervisor allows the virtualization of multiple operating systems running alongside each other and sharing the same virtualized hardware resources. Without it, hardware can only run one operating system at a time [60].

We can use hypervisors with two different types. A type 1 hypervisor runs directly on the host's hardware to manage guest operating systems. It takes the place of a host operating system, and virtual machine resources are scheduled directly to the hardware by the hypervisor. This type of hypervisor is the most commonly used in an enterprise data centre or other server-based environments. On the other hand, a type 2 hypervisor runs on a conventional operating system as a software layer or application. It works by abstracting guest systems from the host operating system. It schedules all resources from its virtual machines against a host operating system, eventually providing the hardware necessary to execute all the tasks. Type 2 hypervisors are better for individual users who want to run multiple operating systems on a personal computer [60].

Here we present several hypervisors type 1, which we considered for our case:

- **VMware ESXi**

VMware [59] is a known company for providing hypervisors. ESXi is a robust type 1 hypervisor. This operating system can effectively partition hardware to consolidate applications and cut costs. It is one of the most commonly used hypervisors because of its efficient architecture, reliability, performance and support. VMware ESXi offers a free subscription with a limit on the number of logical CPUs per host and virtual machine. A paid subscription removes these limitations.

This operating system would be suitable for most of the basic operations intended. However, to pass through a graphics card or any PCI device to a virtual machine, VMware ESXi requires a paid subscription [59].

- **Citrix Hypervisor**

Citrix Hypervisor [61] enables organizations of any size or type to consolidate and transform compute resources into virtual workloads for today's data centre requirements. Meanwhile, it ensures a seamless pathway for moving workloads to the cloud. This operating system integrates existing networking storage infrastructures enabling schedule zero downtime maintenance by live migrating VMs between Citrix Hypervisor hosts. It also offers high availability, which will start the virtual machine on another server if the first fails .

The Xen Project is an open-source hypervisor used as the basis for many

different commercials and open-source applications. Xen Project is the base of Citrix Hypervisor. While using it, Citrix Hypervisor comes with extra features and supports provided by Citrix [61].

Citrix Hypervisor allows hardware-assisted virtualization using virtualization extensions from the host CPU to virtualise guests. This way, it creates hardware virtual machines which do not require any kernel support. This operating system uses Quick Emulator (QEMU) to emulate PC hardware, including BIOS, IDE disk controller, VGA graphic adaptor, USB controller, and network adapter, among many others. Using the Citrix Hypervisor tools will improve the performance of hardware-sensitive operations like access to disks or networks [61].

Like many other tools, Citrix Hypervisor allows the management of multiple servers connected to shared storage as a single entity by using resource pools. Resource pools allow moving and running virtual machines on different servers. Each pool can contain up to 64 servers running the same version of Citrix Hypervisor at the same patch level and with broadly compatible hardware. This operating system also allows hosts to share their storage as a repository. Storage repositories store virtual disk images, which contain the contents of the virtual disks that the virtual machines use. Storage repositories are flexible, with built-in support for SATA, SCSI, NVMe and SAS drives that are locally connected, and iSCSI, NFS, SAS, SMB and Fibre Channel remotely connected. This abstraction of storage repositories and virtual disk images allows fast snapshots and cloning to exposed storage that support them [61].

- **Nutanix AHV**

Nutanix AHV [62] is a type 1 hypervisor based on open-source KVM. It has a Linux based architecture, including CentOS kernel and KVM complex code to virtualise the computing and storage.

Nutanix AHV architecture has three components: KVM-kmod, Libvirt and Qemu KVM [62].

KVM-kmod or KVM Kernel Module supports virtual machines using hardware support. It consists of a loadable kernel module that provides core virtualisation infrastructure and a processor-specific module. In most cases, the provided versions are sufficient to run qemu with KVM support [62].

Libvirt is the daemon service utility to manipulate VMs. Libvirt is simply a virtualisation management collection of software that provides a convenient way to manage virtual machines and other virtualisation functionality, such as storage and network interface management. These software pieces include a long-term stable C API, a daemon, and a command-line utility. A primary goal of libvirt is to provide a single way to manage multiple different virtualisation providers or hypervisors [62].

QEMU and KVM act as hypervisors built into the Linux kernel. However, QEMU is a generic and open-source machine emulator and virtualiser and cannot simulate a complete hardware environment like a CPU. KVM helps QEMU to access hardware virtualisation features on different architectures. It also adds the acceleration feature to the QEMU process. In short, when together, QEMU acts as the hypervisor and KVM as the accelerating agent [62].

Nutanix AHV is free and has a broad community. However, there was a better solution to what we pretended [62].

- **Proxmox VE**

Proxmox VE [63] is a Debian Linux platform for running virtual machines. This open-source operating system allows Kernel-based Virtual Machines and Container-based virtualisation.

One of the main goals of the design of this operating system was to make administration as easy as possible. Proxmox VE is usable on a single node or a cluster of many nodes. Every node is easily manageable with its web-based management interface [63].

Proxmox VE [63] is a versatile platform. Its web-based interface integrates a unique multi-master design, giving a clear overview of all virtual machines, containers, and clusters. This interface allows the deployment and management of every needed virtual device from any node without requiring additional management nodes or massive databases. Proxmox VE also comes with a unique file system. Proxmox Cluster file system is a database-driven file system that enables storing thousands of virtual machines. Using Corosync, these files get replicated, in real-time, to all cluster nodes. The file system stores all data inside a persistent database on a disk. Nonetheless, a copy of the data resides in RAM providing maximum storage of 30 MB.

Proxmox VE allows role-based administration for more advanced usage where users and permissions can control access to every object [63].

In sum, the Ubuntu server fits the purpose of one service at a time. It is an operating system suitable for testing the server's stability and operating each virtual machine. The operating system we required to operate the whole server was a hypervisor. In this study, we learned about some of the most commonly used hypervisors and concluded that Proxmox best fits our needs.

2.5 Tools for Development

In this section, we briefly describe some of the tools we used for the second problem of this project.

ADNI

The first among all these tools is the database. We used the Alzheimer's Disease Neuroimaging Initiative (ADNI) [64] database. The ADNI is a longitudinal multisite observational study of elderly individuals with normal cognition, mild cognitive impairment and Alzheimer's disease. ADNI has a battery of NPAs that we used in machine learning techniques to find correlations between the different neuropsychological assessment tools and see the possibility of creating a profile resulting from gathering multiple results from multiple assessments.

Python

The second tool worth mentioning is Python [65, 66]. Nowadays, Python is one of the most used programming languages. It is the primary tool used for this project. Python is a multi-purpose, high-level programming language that makes the code easy to read and maintain. It also supports multiple programming paradigms and is compatible with most informatic systems. Python also contains many libraries with frameworks and tools. Most are open source and simplify complex software development.

MongoDB

ADNI does not have an API to access files. Instead, all data comes in the CSV format, which is acceptable in Python. However, MongoDB was used to prevent

having multiple backups of these files, ensure their accessibility on multiple machines, and ensure that files do not change during this project.

MongoDB [67, 68] is an open-source document-oriented database for high-volume data storage. Instead of using tables and rows as in relational databases, Mongo uses collections and documents. Documents consist of key-value pairs, which are the basic unit of data in MongoDB. Collections contain documents and functions equivalent to relational database tables. Each document structure can be different, with various fields. This structure aligns with how developers construct their classes and objects in their respective programming languages. Documents do not need to have a defined schema. Instead, it creates fields alongside their need. The data model available allows the representation of hierarchical relationships to store arrays and other more complex structures more effectively.

Once uploaded, all files were accessible using PyMongo. PyMongo is a recommended set of tools for working with MongoDB using Python [67].

Pandas and Numpy

Data stored in MongoDB was ready to be used. PyMongo does a great job when it comes to retrieving this data when we need to consult a row from a document or get the information from one column. However, Pandas and Numpy were the chosen tools to manage and interact with the data.

Pandas [69] is an open-source package, developed resourcing to Numpy and widely used for data science and analysis. Like Numpy, it supports multidimensional arrays and works perfectly with various modules inside the Python ecosystem. This tool simplifies the process of data cleansing, filling, and normalisation. It also allows merges and joins for different tables and is incredible for data visualisation and inspection.

Jupyter

Another helpful tool was JupyterLab [70]. JupyterLab is an interactive development environment that enables users to create and share documents that combine live code with narrative text, mathematical equations, visualisations, interactive controls and other rich outputs. JupyterLab provides a customisable web graphical interface where users can interact with a file browser, terminals, text editor and a collection of extensions.

GitHub

When developing a project, this size is always an excellent idea to keep a version control tool to help maintain the code and quickly transfer it between systems. For this project, we chose to use GitHub to keep everything together.

GitHub [71] is an open-source project based on another open-source tool named Git. Git is a version control system created by Linus Torvalds in 2005, which allows the entire codebase and its history to be available on every developer's computer for easy branching and merging. The version control system helps developers track and manage project code changes. It allows the branching of the code to work in a duplicated version, which can later merge with the main code. GitHub is a cloud-based Git repository hosting service. Essentially, making a lot easier for individuals and teams to use Git for version control and collaborations. GitHub provides a simple web interface that some even use for other projects like writing books.

Chapter 3

Development

As aforementioned, this project was composed of two main problems, the infrastructure and the profiling component. On the infrastructure component, we aimed to build a system for the BRaNT project. This system must serve a web server with a database and, at the same time, aid in developing, training and maintaining artificial intelligence models.

3.1 BRaNT

BRaNT is a project proposed to enhance Task Generator. As mentioned before, Task Generator can generate paper and pencil tasks for rehabilitation. However, its design cannot adapt its tasks nor monitor the cognitive process of the patient. BRaNT will use belief revision, machine learning, gamification, and remote monitoring capabilities to create NeuroAlreh@b . This tool will enable health professionals to provide long-term personalised cognitive rehabilitation therapy at home [2].

NeuroAlreh@b is under development within a framework which combines neuropsychological assessments and rehabilitation, artificial intelligence, and game design. This framework is composed of three main tasks, each containing their challenges. In the first task, we need to establish an optimal cognitive profile. The second aims to develop cognitive training tasks according to the patients' cognitive profile established on the first task. In the last challenge, we need to adapt the cognitive training task from session to session according to the patients' performance. 3.1 contains a schematic description of the project's framework.

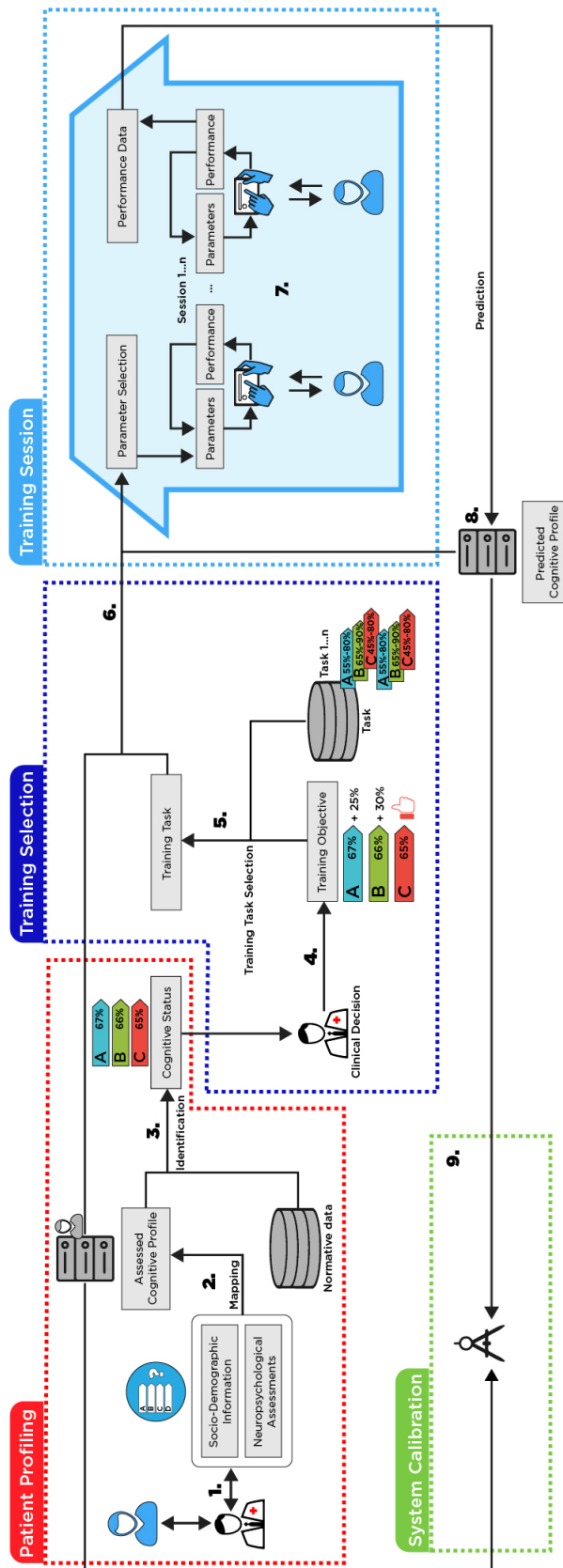


Figure 3.1: NeuroAlreh@b proposed framework

3.1.1 The Framework and its Challenges

Cognitive rehabilitation is the most effective approach to addressing cognitive deficits [72]. However, current cognitive rehabilitation tools are a) challenging to adapt to every patient since they demand the application of an extensive battery of neurocognitive assessment instruments whose results are interpreted manually and often prone to errors in the selection of cognitive training tasks, b) have a high implementation cost, since they involve several sessions performed in clinical environments by neuropsychologists and c) session to session adaptation to the patient performance is not always performed, which may limit the rehabilitation potential and motivation of the patient.

To address those main limitations, we developed a framework for personalized and adaptive delivery of CR that can be divided into four different components, as indicated in Figure 3.1. This section presents a high-level description of the various processes from each component present in the framework (numbers 1 – 9 in the Figure). We also present the challenges involved in NeuroAlreh@b development.

The Patient Profiling: This component aims to create a multi-dimensional patient profile that integrates several NPAs to determine a baseline cognitive status.

1. In this step, neuropsychologists use validated neurocognitive assessment tools to assess patients' cognitive functioning. A neurocognitive assessment tool is a standard part of integrated medical care and is necessary to implement and evaluate rehabilitation procedures. A comprehensive neurocognitive assessment has the following objectives: i) to characterize cognitive abilities, performance in activities of daily living, personality traits, emotional and behavioral functioning, in comparison to the levels of premorbid functioning of the patient; (ii) to quantify the nature and severity of cognitive and functional deficits, symptoms and signs present in the context of the structural and functional integrity of brain functioning, to differentiate normal and pathological cognitive decline (iii) to define a baseline level of performance in cognitive, functional, and emotional functioning domains, which can be examined in a longitudinal registry, through repeated evaluations, thus enabling monitoring of the clinical evolution of the patient, in terms of recovery of functions, and response to interventions (e.g., rehabilitation, psychotherapy, pharmacological therapy) or the disease's

progression; (iv) to identify personal resources and preserved functions that are equally useful for planning and implementing compensatory and preventive intervention procedures, as well as evaluating their effectiveness, with the aim of promoting the patients' well-being and quality of life. **Challenges:** To represent a profile, one must define a formal language. This language has the purpose of representing the profiles and expressing their properties for computing metrics and determining their dynamics. After setting the general structure, we need to identify which cognitive, behavioural, emotional and functional abilities we should consider in a patient profile. Another essential part is to specify which are the relevant neuropsychological assessment tools. These tools must compile a comprehensive multidimensional evaluation of each cognitive, functional and emotional domain. It is also crucial to determine which socio-demographic information to consider for profiling purposes.

2. By aggregating the different neuropsychological assessments and considering the SDI of the patient, the system creates an Assessed Cognitive Profile. **Challenges:** To the best of our knowledge, there is a gap in the literature regarding how to integrate data from multiple and heterogeneous neurocognitive assessments and consolidate them into a consistent profile.
3. The assessed cognitive profile is insufficient to determine the patient's cognitive status. It is necessary to compare it with the normative data available for each neuropsychological assessment tool. **Challenges:** The normative data for each neuropsychological assessment tool is usually given according to different clinical conditions and divided by sociodemographic groups. Like in step 2, this needs to be aggregated and compared to the consolidated profile with an objective and quantifiable distance metric.

The Training Selection: After determining the patient's cognitive status, which gives information about the preserved domains, types, and extent of the impairments, it is time to define the most appropriate cognitive training tasks for neurorehabilitation therapy.

4. Based on the patient's cognitive status, neuropsychologists determine specific training objectives for each patient, i.e., in which cognitive domains the rehabilitation training must be focused to regain or

compensate for lost cognitive abilities and functional independence.

Challenges: The ultimate goal of cognitive rehabilitation is to help patients regain independence and autonomy in their daily activities. Establishing objectives on a system that mostly accepts numerical values can be challenging when the objectives are usually set subjectively. The system should be able to perform this translation.

5. After establishing the training objectives and the set of available cognitive training tasks in the system, NeuroAlreh@b computes which tasks are most appropriate for the training. This computation is possible because each task has its profile, which details which cognitive domains are required to perform a given task and how to parametrize each task difficulty for the different cognitive domains. **Challenges:** Each task needs to include information about which cognitive domain it trains and its intensity. Additionally, it should include constraints regarding the minimal and maximal values for a particular cognitive domain of the suitable patient profile. Optimally provide the subset of tasks selected for a training objective, considering that the number of selected tasks is also limited given the time and number of sessions assigned to the cognitive training.
6. The system establishes the initial parameters for the tasks by combining the initial profile and cognitive training tasks. These initial parameters determine the task's difficulty according to the different cognitive domains. **Challenges:** Combination based on a tuple is not trivial. The tuple contains the task, the value of the cognitive domain in the profile and the associated difficulty. The system will adjust these using machine learning techniques (see System Calibration).

The Training Session: This part describes the training sessions performed by a patient at home.

7. At home, the patient executes the prescribed training sessions. Each training session consists of a set of predefined tasks. The patient executes these tasks using a tablet or a personal computer. The NeuroAlreh@b itself will calculate the patient's performance at each iteration and redefine its difficulty by changing its parameters to maintain a patient score in a task between 50% to 70% of success to avoid frustration and keep patients engaged [73]. **Challenges:** The system must establish a relation between the scores in various tasks, each task parameter and the resulting difficulty

for each cognitive domain of the patient profile to maintain the average score between 50% and 70%. This part of the system employs machine learning for calibration.

8. After a complete training session, the system aggregates the performance in all tasks and estimates if there was evolution or involution in the different cognitive domains of the patients' profile, defining a *predicted cognitive profile*. **Challenges:** Defining the predicted cognitive profile involves multiple challenges. It is necessary to define the profile dynamics when adding new information, given a profile representation. The system must perform minimal changes in the profile to accommodate the new information. This minimal change requires applying belief revision techniques adapted to the profile representation languages mentioned in step 1.

The System Calibration: This part describes the system's calibration when comparing the predicted profile with newly acquired accurate profile data.

9. At the end of each training session, the patient performs a new neurocognitive assessment and the cycle restarts. The system compares the newly assessed cognitive profile with the predicted cognitive profile. If there are differences between them, the system analyzes the possible causes of the divergence and recalibrates the system adequately. **Challenges:** The divergences can have different origins. They can be caused by a wrong prediction in step 8, by a non-accurate model of the relationship between tasks and cognitive domains in step 5, or by a suboptimal adjustment of the parameters in step 7.

3.2 Contribution to the BRaNT challenges

As mentioned, this project consists of two problems. In the first problem, we will use the available hardware to build and implement a computer to serve web pages and run artificial intelligence models and simulations. For the second problem, we will use an existing database with a battery of neurocognitive assessments to study the possibility of aggregating such assessments into a consistent cognitive profile.

BRaNT's primary focus is on brain stroke and the degenerative neuro deficits related to that condition. However, it does not have the required data on the

brain stroke to test its hypothesis. With the help of the Alzheimer's Disease Neuroimaging Initiative, we will try to develop this hypothesis for Alzheimer's. This way, we can have a proof of concept and some experience that BRaNT can take advantage of when the time comes.

3.3 Server Deployment

3.3.1 Hardware available for BRaNT

The budget for the hardware was around 4500€, which came with some short hands. An ideal solution would be a 2U server. These are easy to upgrade and have a much lower probability of failing. Most of these servers offer a next-day assistance service during the warranty period and even smaller monitoring hardware that allows changing bios definitions and installing or restoring an operating system remotely.

Within our budget, the priority was to have enough processing capabilities to run our machine learning simulations. Most of these simulations run in TensorFlow, meaning we can use GPUs instead of the CPU to run those. With this in mind, we got three graphics cards in our system.

The hardest part of building a system with three graphics cards is finding a motherboard with three PCIe x16. Most motherboards sold only offer two of these slots. In the end, we were able to get an Aorus X570 Pro, which supports the three graphics cards as intended.

To control everything, we got the second-best processor at the time. AMD had just released these, and we were able to get one.

In terms of memory, we were able to get four sticks of 16GB DDR4 at 3200MHz. These should be enough to deal with all the requests from the web servers and the simulations simultaneously.

Regarding hard drives, the system has one M.2, which will run the primary operating system and allocate images for the operating systems required. This M.2 has a 1TB capacity, which is more than enough for the work designed for it. The remaining drives will host all the virtual machines. We will assemble these four SATA SSDs with 1 TB each on a RAID 10 array.

The finished system had the following hardware:

- **CPU:** AMD Ryzen 9 3900X
- **GPU:** 3 x Nvidia GeForce RTX 2080 SUPER
- **MotherBoard:** Aorus X570 Pro
- **SSD:** 4 x SATA 1000GB + M.2 1000GB
- **RAM:** 64GB DDR4 at 3200MHz
- **PSU:** 2600W

The system came pre-assembled with Windows 10 pre-installed. However, to better test the stability, we installed Ubuntu Server and served a quick page while we chose which would be the final operating system.

Selecting an Operating System

With Ubuntu Server installed, we left it running for a while. The system looked stable, and it was time to install the final operating system, as we wanted every different system sandboxed to avoid corruption and instability between systems. For example, if the operating system running artificial intelligence becomes unstable, it should not affect the web servers and vice-versa.

The first system we thought of was VMware Server. This operating system would be able to virtualize every operating system that we wanted at a later stage. However, VMware Server is a hypervisor type 2, which means it is a single point of failure. This system would require a fully-edge host operating system to manage the resources before allowing VMware Server to create and serve its virtual machines.

The solution we were looking for was a hypervisor of type 1.

After considering all the operating systems mentioned in the state-of-the-art section, we used Proxmox VE. Like other systems based on Linux KVM, machines have good portability, backups are easy to schedule, and whenever a problem pops up, there is a vast community able to help solve it. Another significant input for this choice was that Proxmox VE has a free community edition and is one of the most used hypervisors after the paid VMware ESXi.

With the operating system selected, it was time to prepare our machine for its purpose. At the time of this project, there were two main components designated for this machine, a complete web server with all the requirements to

allocate multiple pages and the capacity to run artificial intelligence algorithms and simulations.

We started by asking the university for 10 IPs. These will allocate the server and eventual virtual machines we need to deploy in the future. With the IPs sorted and the machine already tested, we installed Proxmox and sent the machine to the server room.

3.3.2 Troubleshooting

After having the server in the server room for a while, eventually, the system went offline. We requested access to reboot the machine, and the system seemed to be back to stability. However, it was not the case. Once the system was online, we checked the logs to find what had happened. No logs reported any service misfiring or any trouble at all. The server stood running without any issues for about an hour. After some time, we tried to start up a new virtual machine. At this point, the option for cloning virtual machines was disabled. For this reason, we started a backup of the virtual machine we had.

This backup job went flawlessly until around 22% when we lost connection to the machine. At first, the thought was that the networking service could have stopped working. However, we restarted the machine the next day and began the backup anew. At the same time, in around 22% of the backup job, the server stopped responding, making it look like a hardware problem.

We requested access and later the removal of the system from the data centre to better debug and find the problem. We noticed that the backup would no longer reach 22% with free access to the system. It would fail a lot faster, around 5-6%. To better debug the issue, we went back to the initial process, installed Ubuntu, and left the system running. The system was running without any issues for three days. We suspected that the issue had to be something wrong with the version of Proxmox installed. With this in mind, we downloaded and installed a newer version.

On the first try, the loading stall. The system powered off on the second and third try of this installation. Since it was not supposed to happen, we were sure the problem had to be hardware. Here we left the system overnight running on the BIOS menu. On the following day, the machine was still running. However, its temperatures were over 80°C, which was not typical for a device that was not even running an OS. Upon a closer look, we could see that the pump from water cooling was not running, we found our problem. The solution was activating

the warranty and swapping the water cooling for air cooling. Since this server will be in a room with controlled temperatures, we do not need to worry about temperatures or noise. With this in mind, we decided to swap the water-cooling for an air-cooling system avoiding further water-cooling-related problems.

During all this debugging and time offline, we had a pilot study going on, and we needed a database and two websites. The solution was to use one of the vast AWS services. For this case, Amazon Elastic Compute Cloud allocated a database and both websites for a short period. Despite being a paid service, it enables the fast deployment of the required services, and we can proceed with the study smoothly. This service allowed us to download the database and upload it to our server.

Reinstall

It was time to put everything back and start our virtual machines with the problem fixed. We returned the computer to the university and installed the Proxmox server with the initial definitions. This time we requested nat and VPN access to the machine, where we proceeded to install everything as we did before. The machine has been running ever since without any issues.

3.3.3 Webserver

In order to serve pages and allocate services required for all the studies that require databases and all the web pages required for such studies, the services we needed were MySQL Server, Apache HTTP Server, Fail2Ban and Certbot.

MySQL is an open-source multithreaded and multi-user structured query language (SQL) database server. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a corporate network's vast amount of information. In this case, data stored in a computer database requires a management system such as MySQL Server. Databases in MySQL Server are relational and store data in separate tables rather than putting all the data in one big storeroom [74].

The following service required was Apache HTTP Server. Apache HTTP Server is an open-source server for modern operating systems, including UNIX and Windows. It is a collaborative effort to create a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server. A worldwide group of volunteers manage and maintain this project [75].

To secure the server from attacks coming from the internet, we proceeded to install Fail2Ban. Fail2Ban is a tool that updates firewall rules on the go. It rejects the IP addresses for a specified time. It comes with filters for various services, including Apache. Fail2Ban continuously scans system logs looking for failed login attempts. Out-of-the-box, Fail2Ban can monitor various network services, such as SSH, Apache, FTP, SMTP and MySQL, among others [76].

To keep all the pages allocated on this webserver certified, we used Certbot. Certbot is a free, open-source software tool for automatically issuing certificates on manually administrated websites to enable HTTPS. Once configured, Certbot will issue a certificate and renew it every 60 days [77].

As a bonus, we also installed Observium on this virtual machine. Observium is a low-maintenance auto-discovering network monitoring platform supporting various device types, platforms, and operating systems. Observium focuses on providing a beautiful and powerful yet simple and intuitive interface to a network's health and status [78].

Observium will monitor everything, including the physical system monitoring and reporting any misfire or trouble with the system, sending an email to the administrator to proceed with maintenance and solve any problem that comes to existence.

The machine was ready, systems were stable, and web pages and projects were ready to be allocated on the first virtual machine. It was time to proceed to the second machine.

Artificial Intelligence Machine

Deploying such a machine required learning and getting more familiar with Linux systems. This machine aimed to use all the graphics cards to help with all the calculations and simulations. However, before passing through all the graphics cards, we must prevent the host operating system from loading the graphics cards.

First, we must see if the host operating system detects the graphics cards. 3.2 shows that the operating system detects all graphics cards perfectly, which means we can pass these cards to the virtual machine that uses them for artificial intelligence and simulations.

```
Linux brant 5.13.19-1-pve #1 SMP PVE 5.13.19-2 (Tue, 09 Nov 2021 12:59:38 +0100) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 21 11:14:47 WET 2022 on pts/0
root@brant:~# lspci | grep VGA
03:00.0 VGA compatible controller: NVIDIA Corporation TU104 [GeForce RTX 2080 SUPER] (rev a1)
08:00.0 VGA compatible controller: NVIDIA Corporation TU104 [GeForce RTX 2080 SUPER] (rev a1)
09:00.0 VGA compatible controller: NVIDIA Corporation TU104 [GeForce RTX 2080 SUPER] (rev a1)
root@brant:~#
```

Figure 3.2: List of Graphics cards connected to the host machine

We started by enabling the IOMMU. We went to the GRUB file and added the “amd_iommu=on” statement to the host operating system bootloader kernel. We confirmed that remapping was supported and added the required modules to the modules file, as shown in 3.3.

```
root@brant:/etc# cp modules modules.backup
root@brant:/etc# vim modules
root@brant:/etc# dmesg | grep 'remapping'
[ 0.788112] AMD-Vi: Interrupt remapping enabled
root@brant:/etc# cat modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
root@brant:/etc#
```

Figure 3.3: Remapping compatibility and adding modules required for pass-through

After a reboot, the system was ready to pass the graphics cards to the virtual machine. 3.4 shows one of the three graphic cards present on the system.

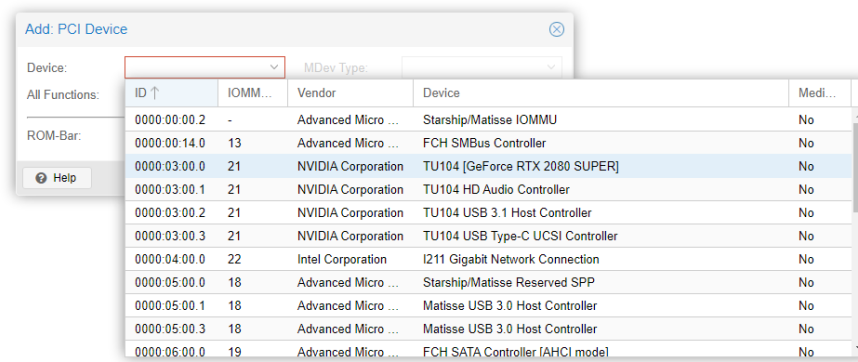


Figure 3.4: PCI devices listed to add to a virtual machine

Despite the learning curve, we were able to pass the graphics cards to the system, where we can now use them to calculate every artificial intelligent algorithm or simulation.

Chapter 4

Results and Discussion

To the best of our knowledge, there is a gap in the literature regarding integrating data from multiple neuropsychological assessments into a consistent profile. Since this is a crucial part of the BRaNT project, we decided to find a way to profile patients. However, BRaNT's primary focus is on stroke, mild cognitive impairment and dementia patients. Moreover, we do not have enough data to use the machine learning approach for profiling patients. With this in mind, this project's main focus consisted of implementing a study case for dementia and evaluating the performance of different algorithms for dimensionality reduction and data regression. The objective is to determine which algorithms can maximise information on a dataset with incomplete, variable and discrepant neuropsychological assessment data. We used many tools to develop methods to test this hypothesis. Here we will describe the most critical ones.

Here we will describe the path we took in this project. The goal here was to get the data from Alzheimer's Disease Neuroimaging Initiative, make the data useable and perform some calculations on it in order to test the possibility of creating a cognitive profile and reduce the data.

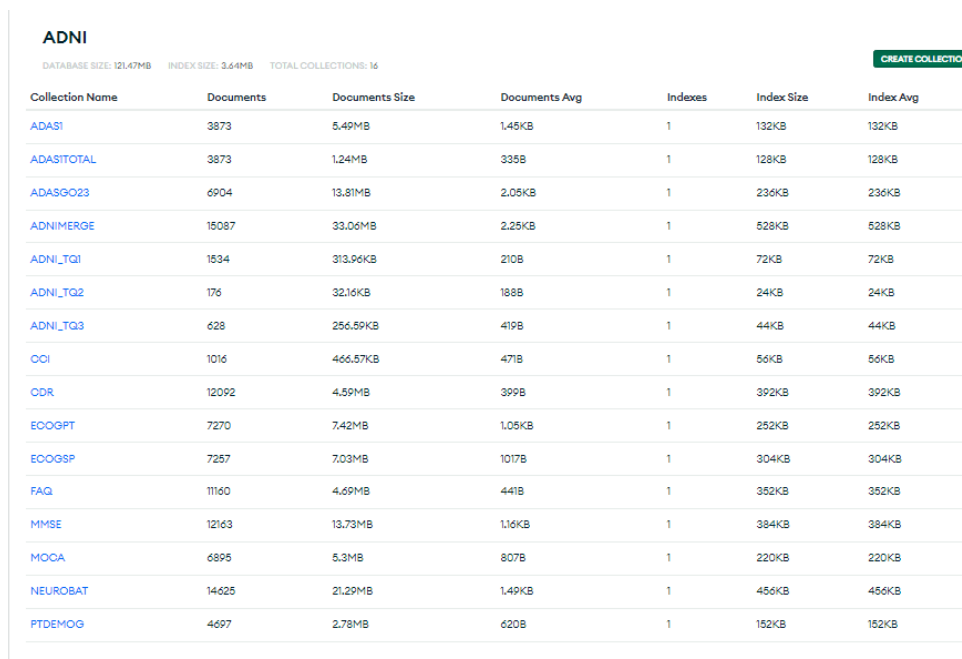
Later, we use classification algorithms to see if the possible cognitive profile is, in fact, able to represent the data. This way, we can improve the performance of algorithms used in the future within the BRaNT project.

Then we will describe how we accomplished the data recreation for the missing fields. This process is essential. It gives us an alternative for missing data, which may occur when a patient misses an evaluation or if some fields are lost during such an evaluation.

4.1 Data Treatment

Before anything, we requested access to the ADNI database. After a while, access was granted to their database, which contains Study Data, Genetic Data and Image Collections.

We focused on the study data since the BRaNT would not use images or genetic information for any training session or acceptance into the program. Here, we can find several fields, such as assessments, biospecimens, genetics, imaging, medical history, etc. For our work, we just needed the Neuropsychological Assessments. At this point, we downloaded all the required files and sent them to MongoDB.



Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
ADAS1	3873	5.49MB	1.45KB	1	132KB	132KB
ADASITOTAL	3873	1.24MB	335B	1	128KB	128KB
ADASGO23	6904	13.81MB	2.05KB	1	234KB	234KB
ADNIMERGE	15087	33.06MB	2.25KB	1	528KB	528KB
ADNI_TQ1	1534	313.96KB	210B	1	72KB	72KB
ADNI_TQ2	176	32.16KB	188B	1	24KB	24KB
ADNI_TQ3	628	256.59KB	419B	1	44KB	44KB
CCI	1016	466.57KB	471B	1	56KB	56KB
CDR	12092	4.59MB	399B	1	392KB	392KB
ECOGPT	7270	7.42MB	1.05KB	1	252KB	252KB
ECOOSP	7257	7.03MB	1017B	1	304KB	304KB
FAQ	11160	4.69MB	441B	1	352KB	352KB
MMSE	12163	13.73MB	1.16KB	1	384KB	384KB
MOCA	6895	5.3MB	807B	1	220KB	220KB
NEUROBAT	14625	21.29MB	1.49KB	1	456KB	456KB
PTDEMOG	4697	2.78MB	620B	1	152KB	152KB

Figure 4.1: Detailed information about the files used

Figure 4.1 contains information about the files we used and uploaded to MongoDB. These files were all in CSV, and we spent some time understanding and cleaning them for use.

Once everything was uploaded, we created a class to download data for use. This class has several methods, and most of them work with each other to achieve the goal of downloading what the current code requires. This class uses the MongoClient from PyMongo to connect and download everything. It allows a simple query to the database or downloads a whole table or only the labels on it. This class also allows downloading data according to a specific column or

uploading a new table as a CSV file.

We set up this data for use with the files uploaded and a way to access them. For every file, we have created a class with the relevant name. The class of each file is responsible for downloading and treating the data. Before implementing each class, we had to understand and select which would be the fields used.

We created a list for each file with the fields to download upon this selection. In the beginning, we downloaded the whole table. However, after several interactions, we implemented a way to download just the required columns, saving time and increasing performance.

Nevertheless, upon downloading the data and selecting it, we were required to convert everything to numeric. All the fields came as a string. Pandas would automatically convert numeric fields to float. However, some of these fields had a space with a number, preventing this automatic conversion. We had to filter the non-numeric part for these fields and convert the number to numeric.

Other fields like demographic data and diagnosis came in text. For example, the diagnosis field in its majority would have three diagnoses, mild cognitive impairment as MCI, normal controls as CN and Alzheimer's disease as AD. To deal with these, we created a key to represent the text as a number. Following the example, to represent 'CN', we used a 0, 'MCI' a 1 and 'AD' a 2. This representation allowed us to run our simulations without the complications of having a text somewhere in our data.

4.2 Profiling

Before anything, we had to check if there was a possibility of assembling a profile. For this, we started with the data present in ADNIMERGE. This data table contains the most crucial ADNI tables merged into one. This file contained 15087 rows with the results from the other tables present in the study.

We used ADNIMERGE to perform a correlation analysis using Pearson's Product-Moment Correlation Coefficients and applied a filter for correlation above 60% and an error probability of 5%. The results presented in Figure 4.2 show that we can identify moderate and robust correlations between various neuropsychological assessments. Note that these correlations ignored fields that missed values. For example, if field A from row 100 has a value, but field B from row 100 doesn't, the values A and B are ignored for the correlation calculation. In any case, we can conclude that simplifying various fields is possible. We can reduce the data and represent it with fewer fields.

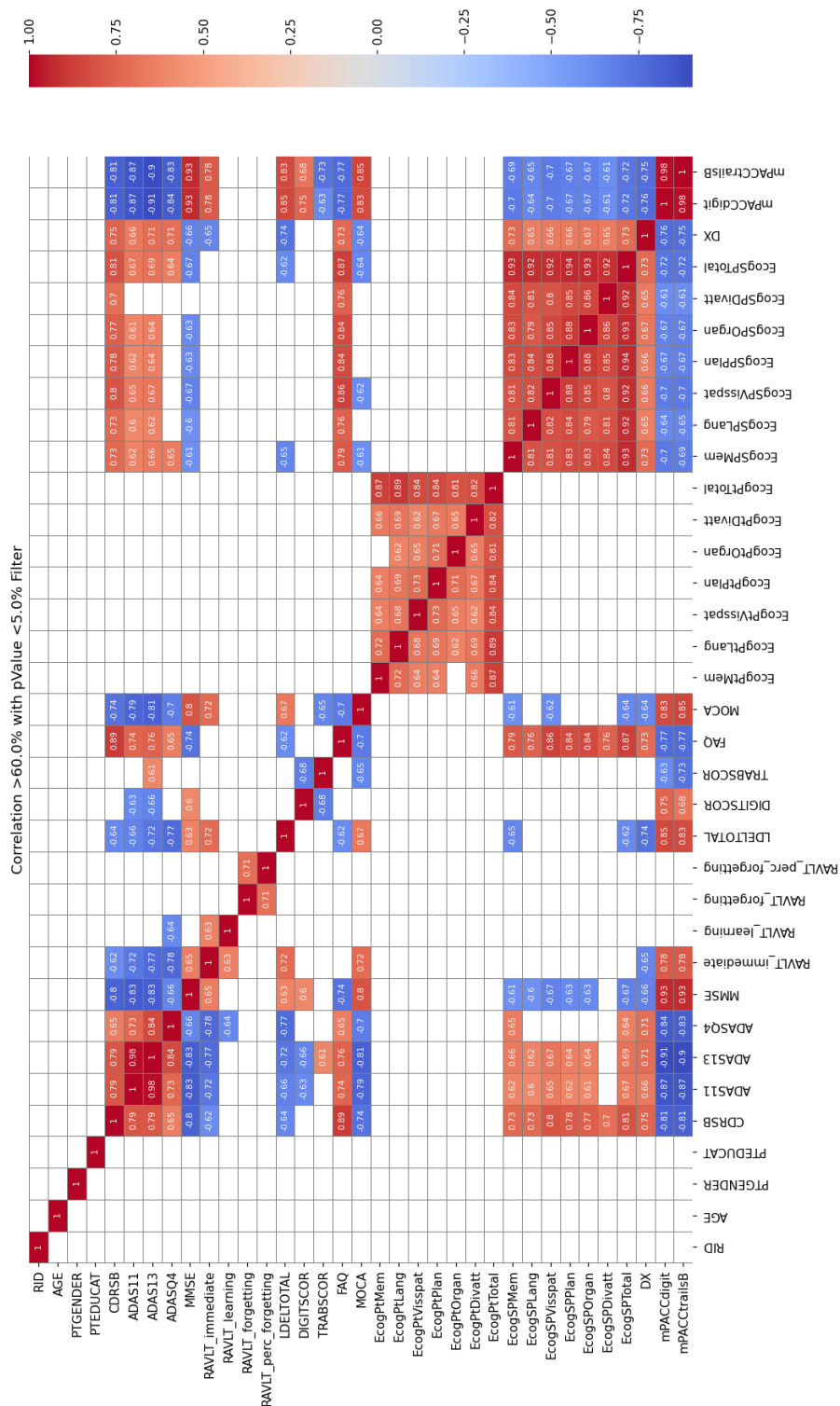


Figure 4.2: ADNI MERGE heatmap for correlation all vs all above 60% and pValue below 5%

We tried to apply a principal component analysis. To achieve this, we had to clean the dataset. The initial dataset was composed of 15087 rows over 36 columns. We removed columns that lacked too many records, and from the remaining columns, we removed the rows with at least one field missing. Our principal component analysis in Figure 4.3 had a dataset of 5393 rows over 32 columns.

The results show that it can represent this data with just two principal components. The scree plot in Figure 4.3 shows that principal component 1 is responsible for more than 80% of the data representation, and the following 2 components represent another 10%.

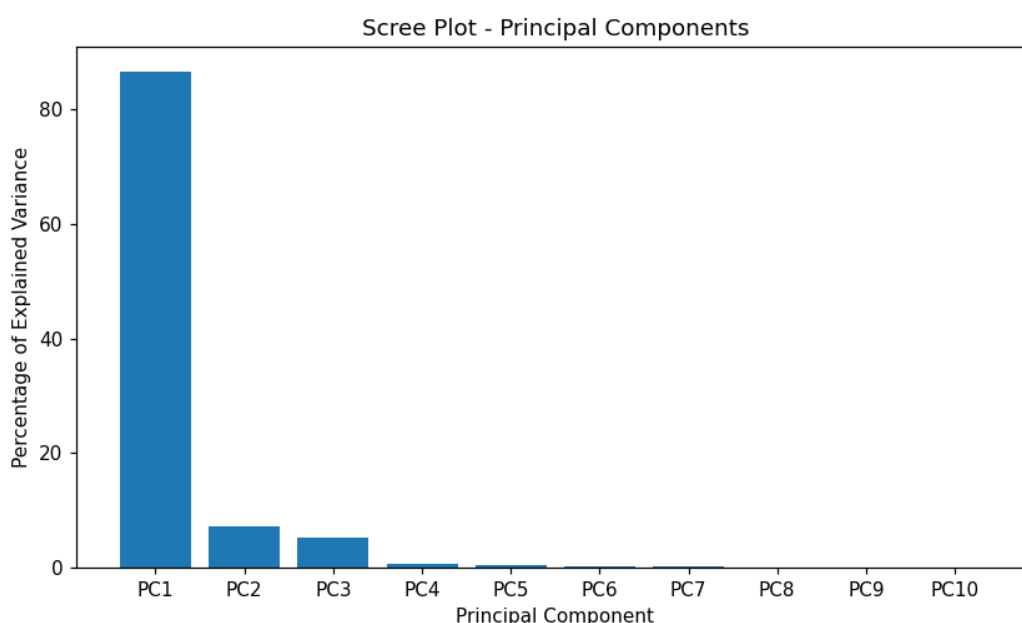


Figure 4.3: ADNIMERGE principal component analysis

4.3 Joining Data

At this point, we did not have much, but we can say that reducing data is possible, and since the results of various exams correlate with each other, we decided to go forward and create a mega dataset.

This new dataset contains all the data from our neuro assessments. The assessments used were: Mini-Mental State Examination, Montreal Cognitive Assessment, Clinical Dementia Rating, Everyday Cognition from study partner

and self-reported, Alzheimer's Disease Assessment Scale-Cognitive, Functional Assessment Questionnaire, Neuropsychological Battery and some demographic data.

The process of joining all the datasets was tedious. As mentioned before, we had to create a class for every dataset. Here we had to partially clean these datasets by removing empty columns, converting numeric strings to float and selecting the relevant fields. With all these classes ready and tested, we created a new class responsible for joining all the output from the vast amount of data sets.

The class responsible for joining data started by removing fields, such as the ID of each file, that was irrelevant. We used two keys to identify which rows correspond to make a primary key.

The data on all files contain a RID, the participant roster identification number and a visit code with the field name VISCODE. These two keys are enough to identify a person in a given intervention. Furthermore, using this identification method, we joined the various tables into a single table, a single dataset.

The resulting dataset had 6863 rows over 286 columns, including many empty fields that required further cleaning.

Like the previous one, cleaning the new dataset, we started by removing columns with most rows with one or more missing values. This filtration resulted in a dataset with 5364 rows over 194 columns. We selected only patients who attended multiple interventions from this dataset, ending with 570 patients who attended three interventions. The resulting dataset had 1710 rows over 194 columns, and we named it CleanData.

Figure 4.4 shows the number of diagnoses present in our dataset in terms of content. Of all 1710 rows in this dataset, 501 were normal controls, 963 were diagnosed with mild cognitive impairment, and 246 had Alzheimer's.

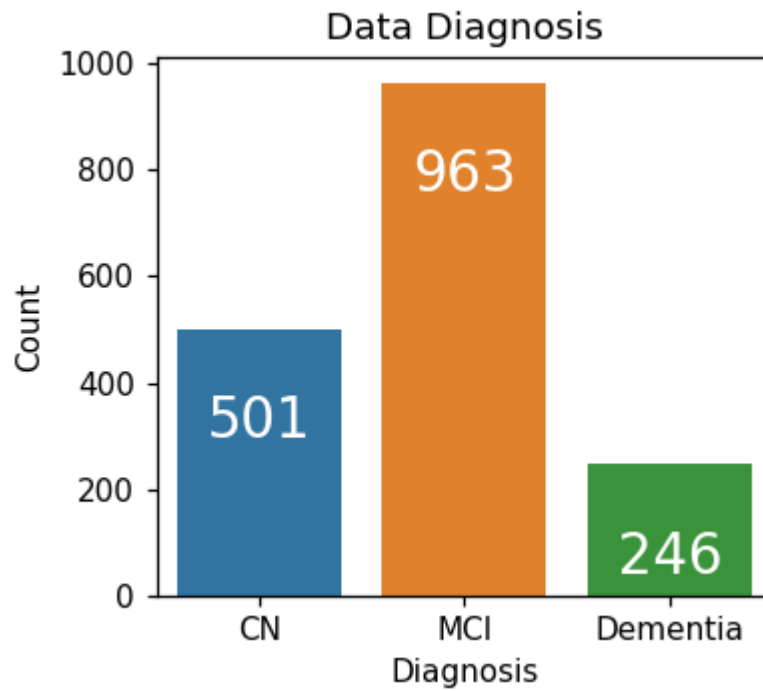


Figure 4.4: Diagnosis present in the CleanData dataset

4.4 CleanData

With the new dataset created, we started with principal component analysis. The goal was to see if the data was representable with just a few components. The results, however, showed that this technique was not good enough to reduce data anymore. When trying to reduce 194 variables to just ten components, data ceases. Figure 4.5 shows the components resulting from this procedure. Now, instead of having three components representing almost 90% of the data, we have many components below 20%. This means this method is no longer effective.

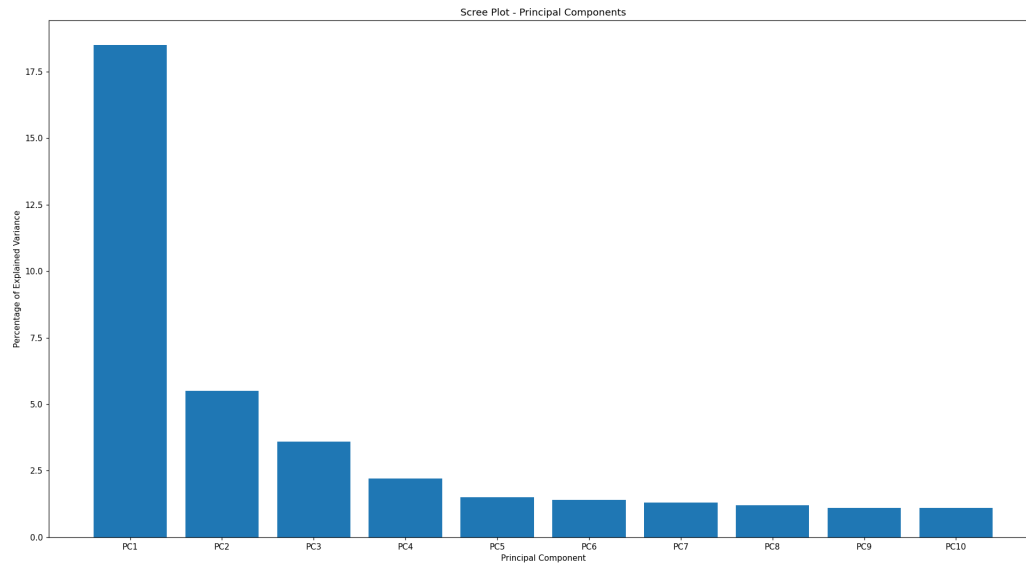


Figure 4.5: PCA to ten components for CleanData

With these results, we have decided on a completely different approach. We tried to reduce the data to 10 components using independent component analysis. We calculated the correlations between the CleanData dataset and these components to see if this procedure had any results. From this correlation shown in Figure 4.6, we conclude that we now have ten components utterly different from each other, with just a few overlaps. We must mention that we ignored all demographic data and the diagnosis while creating these components and, later, the correlation.

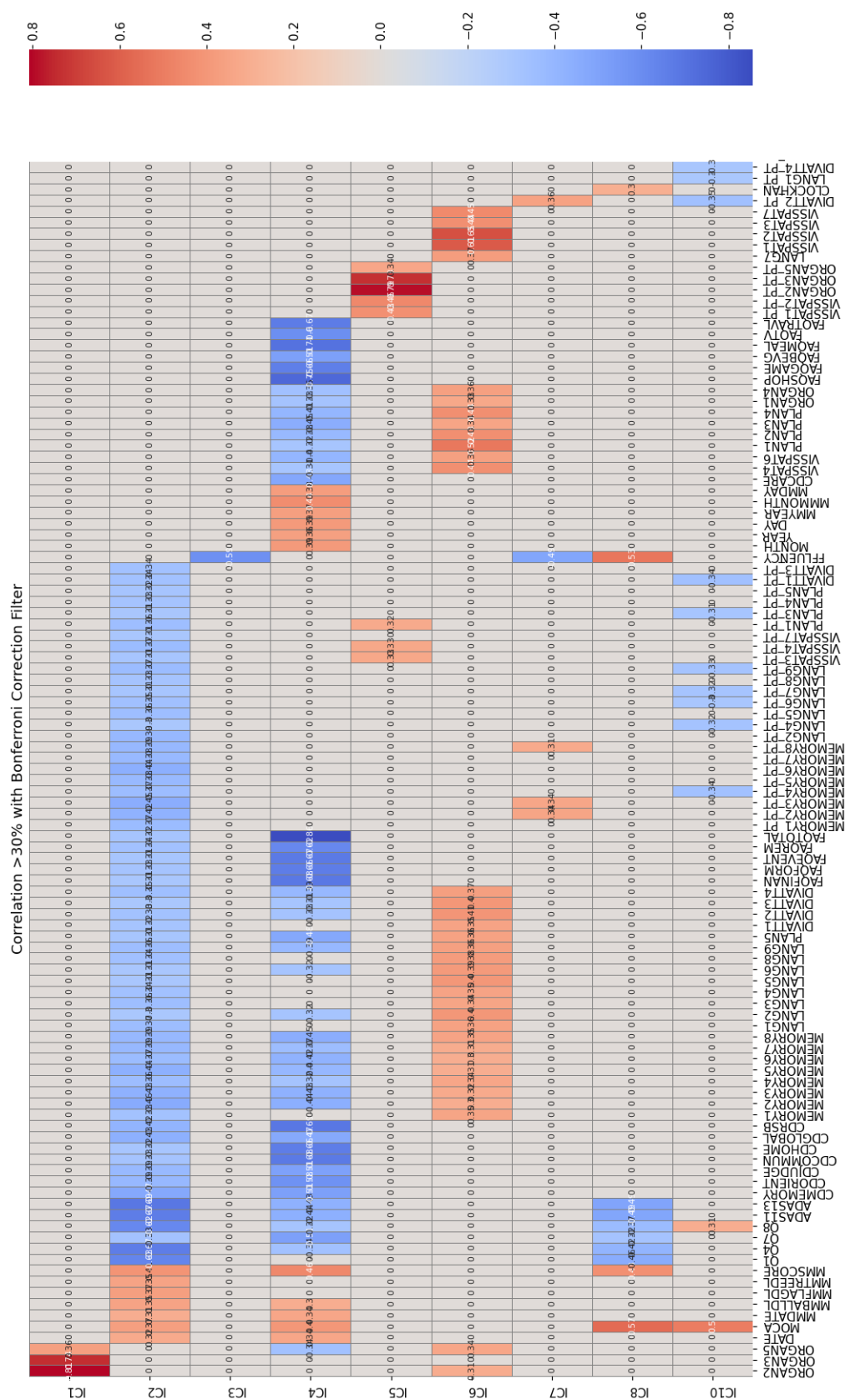


Figure 4.6: Correlation between CleanData and ICA from CleanData

4.5 Classification

At this point, we proceeded to use classification algorithms to better understand how these components can represent our dataset. The algorithms chosen for this were: Decision Tree, K-Nearest Neighbors, Linear Discriminant Analysis, Naïve Bayes, Support Vector Classification, Linear SVC, and Random Forest.

As a baseline for each algorithm, we created classification models testing the diagnosis classification for each algorithm, as shown in Figure 4.7. We used two methods a standard 70% for training with 30% for testing and cross-validation training using ten subsets. Figure 4.7 contains the results for this baseline. As expected, some algorithms have performed better than others.

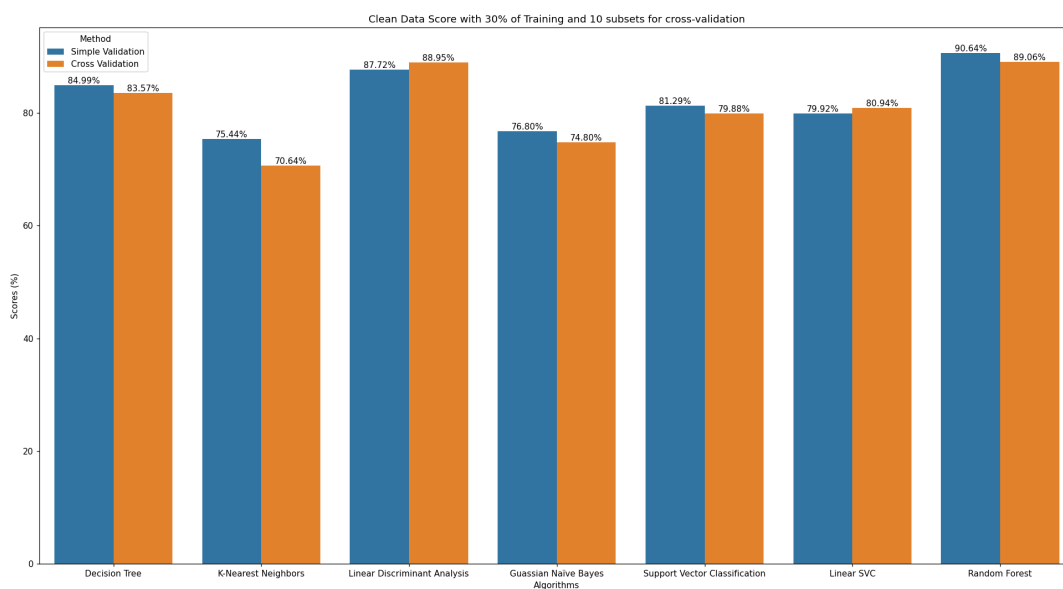


Figure 4.7: Baseline classification for CleanData

We repeated the process with the independent components. The procedure was the same, a standard set with 30% for training and the remaining 70% for testing and a cross-validation training method using ten subsets. Figure 4.8 shows the results obtained. Compared to the CleanData dataset's baseline, the results decreased by about 5 to 10%. These results looked promising since we reduced from 194 to 10 columns losing only about 10% accuracy.

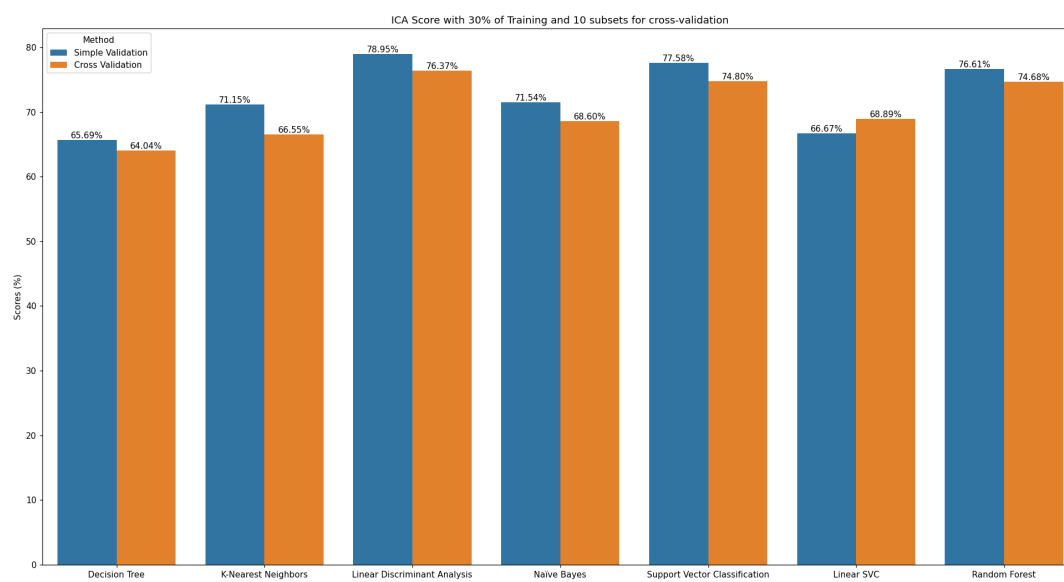


Figure 4.8: CleanData's ICA classification

According to the profiling challenge, the independent component analysis seems to have the ability to solve profile patients. If we look at the correlations, they evaluate different cognitive areas, meaning they are usable as variables for the patients during the NeuroAlreh@b training sessions. Still, we feel that further study by a neuropsychologist is required to see how solid these components are from a clinical perspective.

4.6 Dealing with missing data: Data reconstruction

As mentioned, the dataset that birthed CleanData had 6863 rows over 286 columns. Ignoring the columns not present in our CleanData, we have a dataset with 6863 rows over 194 columns. This dataset contained rows with missing values which we wanted to rebuild. To prepare the original data set, we removed the rows present in CleanData and rows without any diagnosis, leaving a dataset with 5025 rows over 194 columns.

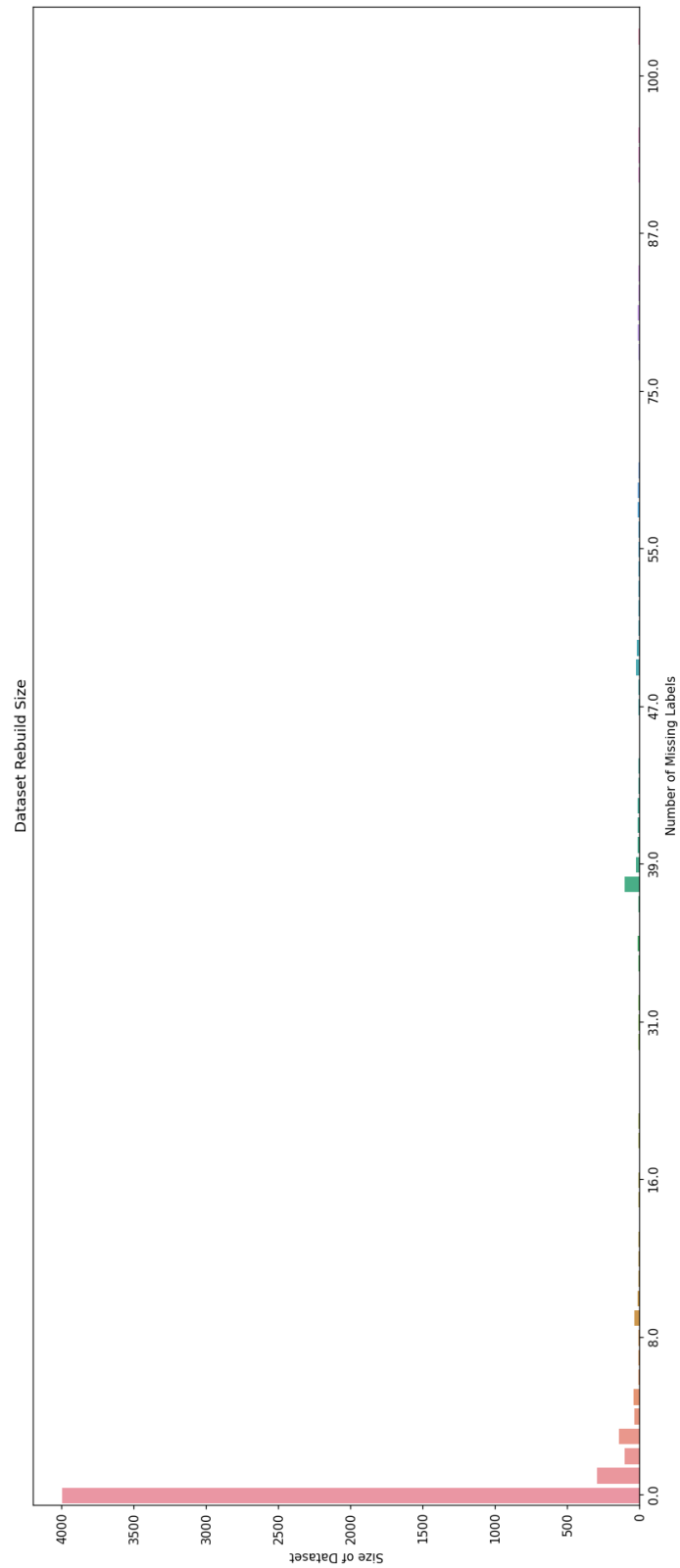


Figure 4.9: Missing values per sub-dataset from the rebuilt dataset

To rebuild the dataset, we counted the missing values. In total, there were 1028 rows with one or more values missing. Figure 4.9 shows the subsets from 0 up to 150 missing values. Each column shows the number of missing values, where the first column does not contain any missing values, and the last column has 150 missing values. Using the K-Nearest neighbours technique designed for this case, we matched the missing value for the average value from the five nearest rows. To calculate this distance, we use the Euclidean Distance.

Once rebuilt, it was time to test the classification for the rebuilt data. We created a subset for all rows missing N or more variables, as shown in Figure 4.10. The first dataset contains all the rows rebuilt in this case.

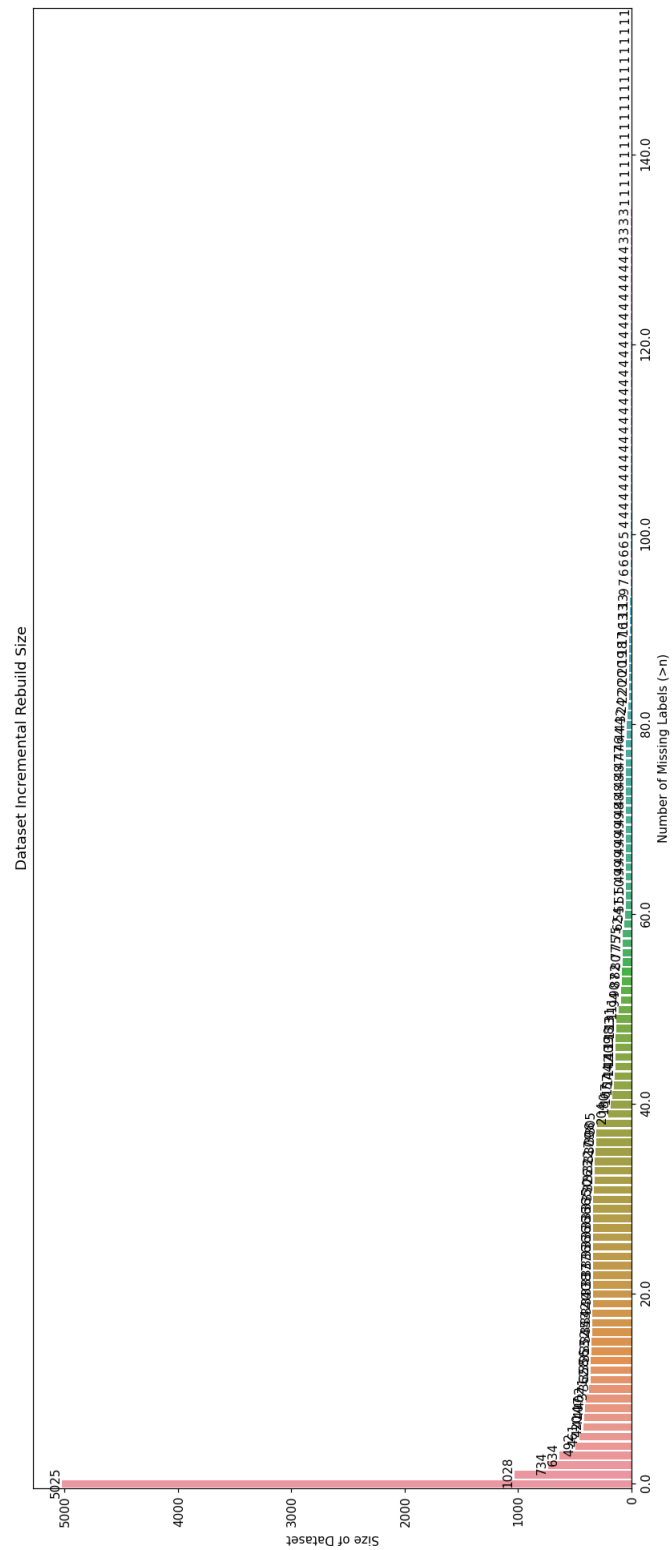


Figure 4.10: Reconstructed dataset and the subsets with N or more missing values rebuilt

We created the same classification models as we did with the first file. This time we had many datasets to represent the same algorithms. Instead of presenting one image for each dataset, we drew the graph using data points. Each column of points results from the diagnosis classification on each algorithm for a dataset. Figure 4.11 shows the resulting graph.

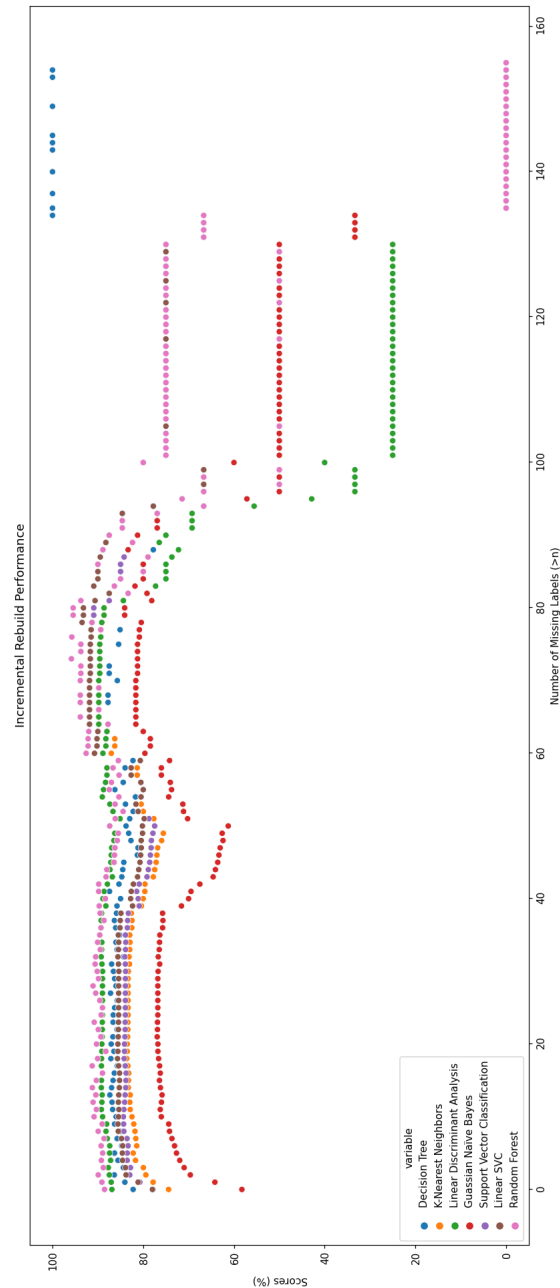


Figure 4.11: Performance of each algorithm on the vast number of datasets

Most algorithms performed well on datasets with less than 100 missing values, achieving above 75% accuracy. Above 100, the datasets are small, which seems to be a fact that is negatively impacting the results. Nonetheless, this experiment concludes that Random Forest and Linear Discriminant Analysis are the most solid algorithms. Both of these algorithms have consistently high accuracy. It also shows that since the data does not follow a gaussian pattern, Gaussian Naïve Bayes cannot perform well.

With this conclusion, we fed the reconstructed dataset to the independent component analysis model and calculated the independent components for each row. The goal was to do like on the first file and see if the accuracy would stay the same as back then.

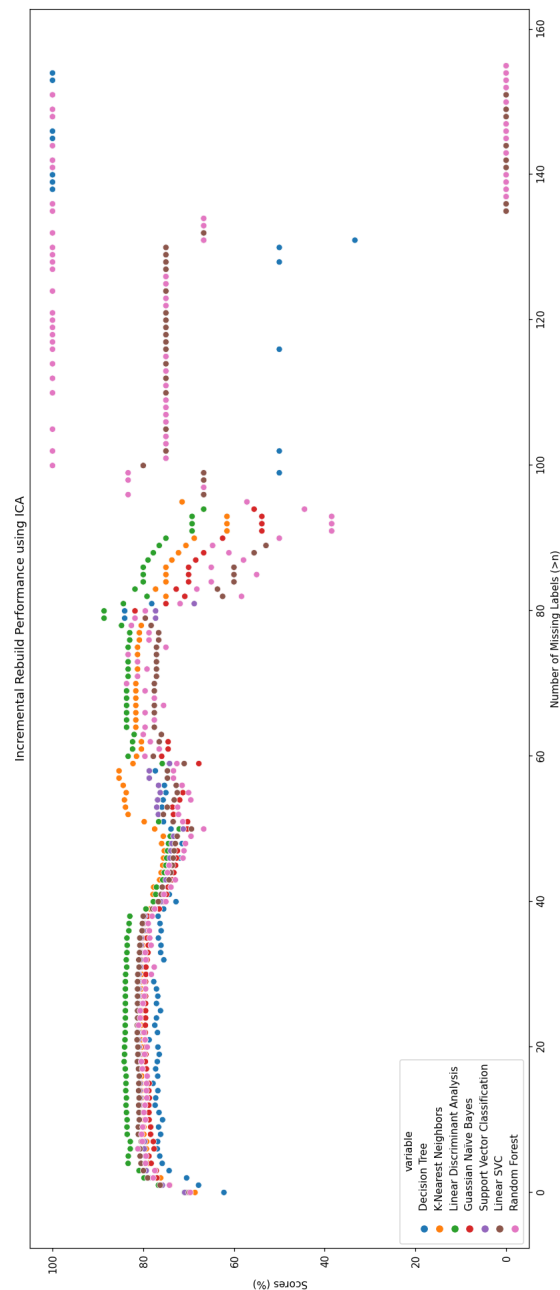


Figure 4.12: Performance per interaction on reconstructed data with independent component analysis

As with the first file, we lost about 10% overall accuracy algorithms. However, now all the algorithms' results are much closer, and although we do not think that it should consider, some algorithms could hit 100% accuracy on above 100 missing values. In this turn, Linear Discriminant Analysis scored better and is the most consistent algorithm. The results from this approach are in Figure 4.12.

Chapter 5

Conclusion

In this master's project, we started by searching previous studies that used artificial intelligence, in this case, machine learning, for rehabilitation systems.

To overcome the fact that the BRaNT project does not have data on stroke patients, we looked for other data sources. Luckily, the Alzheimer's Disease Neuroimaging Initiative had enough data on Alzheimer's Disease that we could access and use to make our study.

The provided data was in CSV format and belonged to multiple studies. After uploading it to MongoDB, we selected one of these files to start working. The file in question had the name ADNIMERGE.

The ADNIMERGE file contained 15087 rows of data from the critical tables and the patient diagnosis. After selecting which data we would use, we ran Pearson's Product-Moment Correlation Coefficient while applying a filter for correlation above 60% and an error probability of 5%. The result concluded that different studies could correlate, and we can select data from these studies to start profiling patients.

With the conclusions taken from the ADNIMERGE, we assembled a new table. This table contained all the information from all the neuropsychological assessments we had where we reran the Correlation Coefficient, obtaining a similar result. Then we tried to profile patients using Principal Component Analysis and Independent Component Analysis, concluding that the Independent Components analysis can perform better on classification algorithms than the Principal components.

Regarding the classification, we used Decision Tree, k-Nearest Neighbours, Linear Discriminant Analysis, Naïve Bayes, Support Vector Classification, Linear Support Vector Classification and Random Forest classifiers to identify the patients with Dementia. Here, we concluded that Linear Discriminant Analysis

is the most stable algorithm despite not always being the best. With an average performance above 75%.

We also proposed a way to deal with missing data. We used the k-Nearest Neighbours to reconstruct datasets with partial information. Furthermore, we tested it with the same algorithms obtaining classifications with an average result above 80% when recovering less than 54% of the fields. This time Random Forest had the best results.

These results are promising. Since there is a gap in the literature regarding consolidating multiple neuropsychological assessments, we consider this an excellent step in a new direction. We tested and concluded that it is, in fact, possible to create a cognitive profile using data from multiple assessments.

This procedure can be a solution for the lack of data or partial data, allowing the creation of cognitive profiles using the previous methodology. The BRaNT project will use these techniques with the Stroke population using their specific neuropsychological assessments for their population. The last part may be helpful for cases where the patient missed an assessment.

As for the first component, we assembled, tested, and deployed a server which is now running Proxmox Virtual Environment. Currently, it only has a virtual machine running as a web server and is ready to deploy any new virtual machine required for future projects.

As for future work, investigating the possibility of joining neuropsychological assessments must continue. Despite the black-boxing layer, it may be helpful to use artificial neural networks to replicate both the reconstruction and the creation of cognitive profiles.

Bibliography

- [1] Friedrich Breyer, Joan Costa-Font, and Stefan Felder. "Ageing, health, and health care". In: *Oxford Review of Economic Policy* 26.4 (Dec. 2010), pp. 674–690. ISSN: 0266903X. DOI: 10 . 1093 / oxrep / grq032. URL: <https://academic.oup.com/oxrep/article/26/4/674/451101>.
- [2] Yuri Almeida et al. "AI-Rehab: A framework for AI driven neuro rehabilitation training - The profiling challenge". In: *HEALTHINF 2020 - 13th International Conference on Health Informatics, Proceedings; Part of 13th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2020*. 2020, pp. 845–853. ISBN: 9789897583988. DOI: 10.5220/0009369108450853. URL: <http://neurorehabilitation.m-iti.org/TaskGenerator/>.
- [3] Alzheimer Association. *Mild Cognitive Impairment (MCI) | Symptoms & Treatments* | [alz.org](https://www.alz.org/alzheimers-dementia/what-is-dementia/related_conditions/mild-cognitive-impairment). 2021. URL: https://www.alz.org/alzheimers-dementia/what-is-dementia/related_conditions/mild-cognitive-impairment.
- [4] *Mild Cognitive Impairment: Symptoms, Causes, Treatments & Tests*. URL: <https://my.clevelandclinic.org/health/diseases/17990-mild-cognitive-impairment>.
- [5] Kathryn Richardson et al. "Anticholinergic drugs and risk of dementia: Case-control study". In: *The BMJ* 361 (Apr. 2018), k1315. ISSN: 17561833. DOI: 10 . 1136 / bmj . k1315. URL: <https://my.clevelandclinic.org/health/diseases/17990-mild-cognitive-impairment>.
- [6] *What is Dementia? Symptoms, Causes & Treatment* | [alz.org](https://www.alz.org/alzheimers-dementia/what-is-dementia). URL: <https://www.alz.org/alzheimers-dementia/what-is-dementia>.
- [7] *Dementia: Causes, Symptoms, Treatment, Diagnosis, Prevention*. URL: <https://my.clevelandclinic.org/health/diseases/9170-dementia>.

- [8] *Alzheimer's Disease: Symptoms, Causes, Treatments*. URL: <https://my.clevelandclinic.org/health/diseases/9164-alzheimers-disease>.
- [9] Alzheimer's Association. *What is Alzheimer's Disease? Symptoms & Causes* | [alz.org](https://www.alz.org). 2020. URL: <https://www.alz.org/alzheimers-dementia/what-is-alzheimers>.
- [10] National Institute on Aging. *What Is Alzheimer's Disease?* | *National Institute on Aging*. 2017. URL: <https://www.nia.nih.gov/health/what-is-alzheimers-disease>.
- [11] *Vascular Dementia: What Is It, Symptoms, Causes & Treatment*. URL: <https://my.clevelandclinic.org/health/diseases/22216-vascular-dementia>.
- [12] *Vascular Dementia | Symptoms & Treatments* | [alz.org](https://www.alz.org). URL: <https://www.alz.org/alzheimers-dementia/what-is-dementia/types-of-dementia/vascular-dementia>.
- [13] Barbara A. Wilson. "Cognitive Rehabilitation: How it is and how it might be". In: *Journal of the International Neuropsychological Society* 3.5 (1997), pp. 487–496. ISSN: 1469-7661. DOI: 10.1017/S1355617797004876. URL: <https://www.cambridge.org/core/journals/journal-of-the-international-neuropsychological-society/article/abs/cognitive-rehabilitation-how-it-is-and-how-it-might-be/F611EB5761428D58B4C1FB54CA599FF2>.
- [14] Thomas D. Parsons. "Neuropsychological Rehabilitation 3.0: State of the Science". In: *Clinical Neuropsychology and Technology* (2016), pp. 113–132. DOI: 10.1007/978-3-319-31075-6{_}7. URL: https://link.springer.com/chapter/10.1007/978-3-319-31075-6_7.
- [15] *Bolsa de Investigação para aluno licenciado no âmbito do projeto BRANT (PTDC/CCI-COM/30990/2017)*. URL: <https://www.arditi.pt/pt/concursos-arquive/bolsa-de-investigacao-para-aluno-licenciado-no-ambito-do-projeto-brant-ptdc-cci-com-30990-2017.html>.
- [16] John McCarthy. *WHAT IS ARTIFICIAL INTELLIGENCE?* Nov. 2007. URL: <https://www-formal.stanford.edu/jmc/whatisai/node1.html>.
- [17] Manisha Sanjay Sirsat, Eduardo Fermé, and Joana Câmara. *Machine Learning for Brain Stroke: A Review*. Oct. 2020. DOI: 10.1016/j.jstrokecerebrovasdis.2020.105162.

- [18] Javier Solana et al. "Improving brain injury cognitive rehabilitation by personalized telerehabilitation services: Guttman neuropersonal trainer". In: *IEEE Journal of Biomedical and Health Informatics* 19.1 (Jan. 2015), pp. 124–131. ISSN: 2168-2194. DOI: 10.1109/JBHI.2014.2354537.
- [19] Hee Tae Jung et al. "Remote Assessment of Cognitive Impairment Level Based on Serious Mobile Game Performance: An Initial Proof of Concept". In: *IEEE journal of biomedical and health informatics* 23.3 (May 2019), pp. 1269–1277. ISSN: 2168-2208. DOI: 10.1109/JBHI.2019.2893897. URL: <https://pubmed.ncbi.nlm.nih.gov/30668485/>.
- [20] Courtney Campbell Walton et al. "Design and Development of the Brain Training System for the Digital "Maintain Your Brain" Dementia Prevention Trial". In: *JMIR Aging* 2019;2(1):e13135 <https://aging.jmir.org/2019/1/e13135> 2.1 (Feb. 2019), e13135. ISSN: 2561-7605. DOI: 10.2196/13135. URL: <https://aging.jmir.org/2019/1/e13135>.
- [21] Iqbal H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions". In: *SN Computer Science* 2021 2:3 2.3 (Mar. 2021), pp. 1–21. ISSN: 2661-8907. DOI: 10.1007/s42979-021-00592-x. URL: <https://link.springer.com/article/10.1007/s42979-021-00592-x>.
- [22] expert.ai. *What is Machine Learning? A definition | Expert.ai*. 2017. URL: <https://www.expert.ai/blog/machine-learning-definition/>.
- [23] Issam El Naqa and Martin J. Murphy. "What Is Machine Learning?" In: *Machine Learning in Radiation Oncology* (2015), pp. 3–11. DOI: 10.1007/978-3-319-18305-3_1. URL: https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1.
- [24] Jiawei Han, Micheline Kamber, and Jian Pei. "Data Transformation by Normalization". In: *Data Mining: Concepts and Techniques* (2011), pp. 113–115. ISSN: 1469-994X. DOI: 10.1016/B978-0-12-381479-1.00001-0. URL: <http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf> <http://scholar.google.com/schol>.

- [25] Sunil Ray. *Commonly Used Machine Learning Algorithms* | Data Science. Sept. 2017. URL: <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>.
- [26] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. "Reinforcement Learning: A Survey". In: *Journal of Artificial Intelligence Research* 4 (May 1996), pp. 237–285. ISSN: 1076-9757. DOI: 10.1613/JAIR.301. URL: <https://www.jair.org/index.php/jair/article/view/10166>.
- [27] George H John and Pat Langley. "Estimating Continuous Distributions in Bayesian Classifiers". In: (). DOI: 10.5555/2074158.2074196. URL: <http://robotics..>
- [28] Sheshadri Iyengar Raghavan Bhagyashree et al. "Diagnosis of Dementia by Machine learning methods in Epidemiological studies: a pilot exploratory study from south India". In: *Social Psychiatry and Psychiatric Epidemiology* 53.1 (Jan. 2018), pp. 77–86. ISSN: 09337954. DOI: 10.1007/s00127-017-1410-0. URL: </pmc/articles/PMC6138240/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6138240/>.
- [29] Iqbal H. Sarker. "A machine learning based robust prediction model for real-life mobile phone data". In: *Internet of Things* 5 (Mar. 2019), pp. 180–193. ISSN: 2542-6605. DOI: 10.1016/J.IOT.2019.01.007.
- [30] Fabian Pedregosa FABIANPEDREGOSA et al. "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: <http://scikit-learn.sourceforge.net..>
- [31] S. le Cessie and J. C. van Houwelingen. "Ridge Estimators in Logistic Regression". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41.1 (Mar. 1992), pp. 191–201. ISSN: 1467-9876. DOI: 10.2307/2347628. URL: <https://onlinelibrary.wiley.com/doi/full/10.2307/2347628%20https://onlinelibrary.wiley.com/doi/abs/10.2307/2347628%20https://rss.onlinelibrary.wiley.com/doi/10.2307/2347628>.
- [32] David W Aha et al. "Instance-based learning algorithms". In: *Machine Learning* 1991 6:1 6.1 (Jan. 1991), pp. 37–66. ISSN: 1573-0565. DOI:

- 10.1007/BF00153759. URL: <https://link.springer.com/article/10.1007/BF00153759>.
- [33] S. S. Keerthi et al. "Improvements to Platt's SMO Algorithm for SVM Classifier Design". In: *Neural Computation* 13.3 (Mar. 2001), pp. 637–649. ISSN: 0899-7667. DOI: 10.1162/089976601300014493. URL: <https://direct.mit.edu/neco/article/13/3/637/6485/Improvements-to-Platt-s-SMO-Algorithm-for-SVM>.
- [34] by J Ross Quinlan, Morgan Kaufmann Publishers, and Steven L Salzberg. "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993". In: *Machine Learning* 1994 16:3 16.3 (Sept. 1994), pp. 235–240. ISSN: 1573-0565. DOI: 10.1007/BF00993309. URL: <https://link.springer.com/article/10.1007/BF00993309>.
- [35] Iqbal H. Sarker et al. "IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model". In: *Symmetry* 2020, Vol. 12, Page 754 12.5 (May 2020), p. 754. ISSN: 2073-8994. DOI: 10.3390/SYM12050754. URL: <https://www.mdpi.com/2073-8994/12/5/754/htm%20https://www.mdpi.com/2073-8994/12/5/754>.
- [36] Leo Breiman. "Random Forests". In: *Machine Learning* 2001 45:1 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: <https://link.springer.com/article/10.1023/A:1010933404324>.
- [37] Leo Breiman. "Bagging predictors". In: *Machine Learning* 1996 24:2 24.2 (1996), pp. 123–140. ISSN: 1573-0565. DOI: 10.1007/BF00058655. URL: <https://link.springer.com/article/10.1007/BF00058655>.
- [38] Yali Amit and Donald Geman. "Shape Quantization and Recognition with Randomized Trees". In: *Neural Computation* 9.7 (July 1997), pp. 1545–1588. ISSN: 0899-7667. DOI: 10.1162/NECO.1997.9.7.1545. URL: <https://direct.mit.edu/neco/article/9/7/1545/6116/Shape-Quantization-and-Recognition-with-Randomized>.
- [39] Yoav Freund and Robert E Schapire. "Experiments with a New Boosting Algorithm". In: (1996). URL: <http://www.research.att.com/>.
- [40] *Polynomial Regression | What is Polynomial Regression*. URL: <https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/>.

- [41] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In: [https://doi.org/ 5.1](https://doi.org/5.1) (Jan. 1967), pp. 281–298. URL: <https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992>.
- [42] Keinosuke Fukunaga and Larry D. Hostetler. "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition". In: *IEEE Transactions on Information Theory* 21.1 (1975), pp. 32–40. ISSN: 15579654. DOI: 10.1109/TIT.1975.1055330.
- [43] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: (1996). URL: www.aaai.org.
- [44] Iqbal H. Sarker et al. "Context pre-modeling: an empirical analysis for classification based user-centric context-aware predictive modeling". In: *Journal of Big Data* 7.1 (Dec. 2020), pp. 1–23. ISSN: 21961115. DOI: 10.1186/S40537-020-00328-3/FIGURES/9. URL: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00328-3>.
- [45] H. Hotelling. "Analysis of a complex of statistical variables into principal components". In: *Journal of Educational Psychology* 24.6 (Sept. 1933), pp. 417–441. ISSN: 00220663. DOI: 10.1037/H0071325. URL: [/record/1934-00645-001](https://record/1934-00645-001).
- [46] Karl Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: <https://doi.org/10.1080/14786440109462720> 2.11 (Nov. 2010), pp. 559–572. ISSN: 1941-5982. DOI: 10.1080/14786440109462720. URL: <https://www.tandfonline.com/doi/abs/10.1080/14786440109462720>.
- [47] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. "Mining Association Rules Between Sets of Items in Large Databases". In: *ACM SIGMOD Record* 22.2 (Jan. 1993), pp. 207–216. ISSN: 01635808. DOI: 10.1145/170036.170072. URL: https://www.researchgate.net/publication/200043124_Mining_Association_Rules_Between_Sets_of_Items_in_Large_Databases_SIGMOD_Conference.

- [48] Rakesh Agrawal and Ramakrishnan Srikant. "Fast Algorithms for Mining Association Rules". In: *Proceedings of the International Joint Conference on Very Large Data Bases, Santiago Chile*. (1994), pp. 487–499.
- [49] Mohammed J. Zaki. "Scalable algorithms for association mining". In: *IEEE Transactions on Knowledge and Data Engineering* 12.3 (2000), pp. 372–390. ISSN: 10414347. DOI: 10.1109/69.846291.
- [50] Stephanie A. Harmon et al. "Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets". In: *Nature Communications* 2020 11:1 11.1 (Aug. 2020), pp. 1–7. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17971-2. URL: <https://www.nature.com/articles/s41467-020-17971-2>.
- [51] Amitabha Das, Wee-Keong Ng, and Yew-Kwong Woon. "Rapid association rule mining". In: (2001), p. 474. DOI: 10.1145/502585.502665.
- [52] Iqbal H. Sarker and A. S.M. Kayes. "ABC-RuleMiner: User behavioral rule-based machine learning method for context-aware intelligent services". In: *Journal of Network and Computer Applications* 168 (Oct. 2020). ISSN: 10958592. DOI: 10.1016/J.JNCA.2020.102762.
- [53] Athanasios S. Polydoros and Lazaros Nalpantidis. "Survey of Model-Based Reinforcement Learning: Applications on Robotics". In: *Journal of Intelligent & Robotic Systems* 2017 86:2 86.2 (Jan. 2017), pp. 153–173. ISSN: 1573-0409. DOI: 10.1007/S10846-017-0468-Y. URL: <https://link.springer.com/article/10.1007/s10846-017-0468-y>.
- [54] Maria Grazia Maggio et al. "Virtual reality and cognitive rehabilitation in people with stroke: An overview". In: *Journal of Neuroscience Nursing* 51.2 (Apr. 2019), pp. 101–105. ISSN: 08880395. DOI: 10.1097/JNN.0000000000000423. URL: https://journals.lww.com/jnnonline/Fulltext/2019/04000/Virtual_Reality_and_Cognitive_Rehabilitation_in.9.aspx.
- [55] Ana Lúcia Faria and Sergi Bermúdez I. Badia. "Development and evaluation of a web-based cognitive task generator for personalized cognitive training: A proof of concept study with stroke patients". In: *ACM International Conference Proceeding Series*. Vol. 01-02-Octo. Association for Computing Machinery, Oct. 2015, pp. 1–4. ISBN: 9781450338981. DOI: 10.1145/2838944.2838945. URL: <http://dx.doi.org/10.1145/2838944.2838945>.

- [56] Ana Lúcia Faria et al. "Benefits of virtual reality based cognitive rehabilitation through simulated activities of daily living: a randomized controlled trial with stroke patients". In: *Journal of NeuroEngineering and Rehabilitation* 13.1 (Nov. 2016), pp. 1–12. ISSN: 17430003. DOI: 10.1186/S12984-016-0204-Z/TABLES/5. URL: <https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-016-0204-z>.
- [57] Jack Wallen. *Ubuntu Server: A cheat sheet - TechRepublic*. URL: <https://www.techrepublic.com/article/ubuntu-server-the-smart-persons-guide/>.
- [58] "What Is a Virtual Machine? How Can I Use VMware Server? Hardware Windows or Linux Operating System VMware Server Application Virtual Machine Windows Application Virtual Machine Linux Application Virtual Machine Windows Application Virtual Machine VMware s". In: (1998). URL: <http://www.vmware.com/support/pubs/>.
- [59] VMware - *Delivering a Digital Foundation For Businesses*. URL: <https://www.vmware.com/>.
- [60] *What is a hypervisor?* URL: <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>.
- [61] *Technical overview | Citrix Hypervisor 8.2*. URL: <https://docs.citrix.com/en-us/citrix-hypervisor/technical-overview.html>.
- [62] *What is Nutanix AHV ? - HyperHCl.com*. URL: <https://hyperhci.com/2020/01/26/what-is-nutanix-ahv/>.
- [63] Proxmox Server Solutions GmbH. *Proxmox VE Virtualization Management Platform*. 2020. URL: <https://www.proxmox.com/en/proxmox-ve>.
- [64] Juan Felipe Beltrá N 1ªa et al. "Inexpensive, non-invasive biomarkers predict Alzheimer transition using machine learning analysis of the Alzheimer's Disease Neuroimaging (ADNI) database". In: *PLOS ONE* 15.7 (July 2020), e0235663. ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0235663. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0235663>.
- [65] TIOBE Software BV. *TIOBE Index | TIOBE - The Software Quality Company*. 2018. URL: <https://www.tiobe.com/tiobe-index/>
<https://www.tiobe.com/tiobe-index/>
<https://www.tiobe.com/tiobe-index/>
<https://www.tiobe.com/tiobe-index/>

- [66] MindfireSolutions. "Python: 7 Important Reasons Why You Should Use Python". In: *Medium.com* (Oct. 2017), pp. 5–8. URL: <https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b>.
- [67] *PyMongo 4.0.1 Documentation*. URL: <https://pymongo.readthedocs.io/en/stable/>.
- [68] David Taylor. *What is MongoDB? Introduction, Architecture, Features & Example*. Nov. 2021. URL: <https://www.guru99.com/what-is-mongodb.html>.
- [69] ActivateState. *What Is Pandas In Python? Everything You Need To Know*. 2020. URL: <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>.
- [70] *JupyterLab is Ready for Users. We are proud to announce the beta... | by Project Jupyter | Jupyter Blog*. URL: <https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906>.
- [71] Kinsta. *What Is GitHub? A Beginner's Introduction to GitHub*. 2021. URL: <https://kinsta.com/knowledgebase/what-is-github/>.
- [72] Caroline Van Heugten, Gisela Wolters Gregório, and Derick Wade. "Evidence-based cognitive rehabilitation after acquired brain injury: a systematic review of content of treatment". In: *Neuropsychological rehabilitation* 22.5 (Oct. 2012), pp. 653–673. ISSN: 1464-0694. DOI: 10.1080/09602011.2012.680891. URL: <https://pubmed.ncbi.nlm.nih.gov/22537117/>.
- [73] David Sharek and Eric Wiebe. "Using Flow Theory to Design Video Games as Experimental Stimuli:" in: <http://dx.doi.org/10.1177/1071181311551316> (Sept. 2011), pp. 1520–1524. ISSN: 10711813. DOI: 10.1177/1071181311551316. URL: <https://journals.sagepub.com/doi/10.1177/1071181311551316>.
- [74] *MySQL :: MySQL 8.0 Reference Manual :: 1 General Information*. URL: <https://dev.mysql.com/doc/refman/8.0/en/introduction.html>.
- [75] *Welcome! - The Apache HTTP Server Project*. URL: <https://httpd.apache.org/>.
- [76] *Fail2ban*. URL: https://www.fail2ban.org/wiki/index.php/Main_Page.

- [77] *Certbot* | *Certbot*. URL: <https://certbot.eff.org/>.
- [78] *Observium*. URL: <https://www.observium.org/>.