UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

Architecture for a decision support system for the automatic definition and adjustment of physical rehabilitation routines to patients affected by neurological diseases

Sergio Martínez Cid

December, 2023

Architecture for a decision support system for the automatic definition and adjustment of physical rehabilitation routines to patients affected by neurological diseases

UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

Architecture for a decision support system for the automatic definition and adjustment of physical rehabilitation routines to patients affected by neurological diseases

Autor: Sergio Martínez Cid
Tutor académico: David Vallejo Fernández

December, 2023

Sergio Martínez Cid

Ciudad Real – España

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE	**VOCAL**	**SECRETARIO**
Fdo.:	Fdo.:	Fdo.:

Abstract

Neurological diseases have a considerable impact on society, partly due to global demographic growth and population ageing. Among these conditions, stroke stands out as one of the leading causes of disability and mortality worldwide. This disorder not only affects the cognitive abilities of those who suffer from it but also their physical capacities. In general terms, patients with neurological diseases need to undergo a physical rehabilitation process that generally extends over several months.

Given the significant impact of neurological diseases, Information Technology systems have emerged to simplify the rehabilitation process. A concrete example is remote rehabilitation, typically conducted at the patient's home as an alternative to traditional rehabilitation. This approach has various advantages, such as reducing costs associated with patient transportation, and specialised personnel. However, this approach faces challenges, such as ensuring effective remote supervision to guarantee the correct execution of exercises.

The project described in this document is part of the development of a commercial system within the company Furious Koalas, where the project author has been involved. The project has involved the design and development of an architecture to enhance the rehabilitation of patients with neurological diseases, providing guidance in their movements and offering support to therapists in monitoring patient progress.

The project has focused on enhancing therapists' ability to delegate tasks in the system, and making its use more intuitive. A key strategy to increase delegation has been the incorporation of a module for detecting compensatory movements in the system. To improve usability, the existing module of Explainable Artificial Intelligence (XAI) has been improved, which aims to provide explanations about the system's automatic decisions to increase user confidence. In addition, automatic decisions have been optimised by improving the system's aggregation processes. Finally, the architecture has been formalised to facilitate future developments. Through these improvements, the general aim is to reduce the therapist's workload and consequently increase the quality time they dedicate to their patients.

Resumen

Las enfermedades neurológicas ejercen un considerable impacto en la sociedad, debido en parte al crecimiento demográfico global y al envejecimiento de la población. Entre estas afecciones, el ictus o accidente cerebrovascular se destaca como una de las principales causas de discapacidad y mortalidad a nivel mundial. Este trastorno no solo afecta las capacidades cognitivas de quienes lo padecen, sino también sus capacidades físicas. En términos generales, los pacientes de enfermedades neurológicas necesitan llevar a cabo un proceso de rehabilitación física que comúnmente se extiende a lo largo de varios meses.

Dada la significativa repercusión de las enfermedades neurológicas, han surgido sistemas de Tecnologías de la Información que buscan simplificar el proceso de rehabilitación. Un ejemplo concreto es la rehabilitación remota, típicamente en el domicilio del paciente, como alternativa a la rehabilitación tradicional. Este enfoque presenta diversas ventajas, tales como la reducción de costes asociados al transporte de pacientes y al personal especializado. No obstante, este enfoque se enfrenta a desafíos, como la realización de una supervisión remota efectiva para garantizar la correcta ejecución de los ejercicios.

El proyecto descrito en este documento se sitúa en el marco del desarrollo de un sistema comercial dentro de la empresa Furious Koalas, donde el autor del proyecto ha tomado parte. El proyecto consiste ha consistido en el diseño y desarrollo de una arquitectura para mejorar la rehabilitación de pacientes de enfermedades neurológicas, proporcionando orientación en sus movimientos y ofreciendo soporte al terapeuta para la supervisión del progreso del paciente.

El proyecto se ha enfocado en mejorar la capacidad de los terapeutas para delegar tareas en el sistema y en hacer su uso más intuitivo. Una estrategia clave para incrementar la delegación ha sido la incorporación de un módulo de detección de movimientos compensatorios en el sistema. Para mejorar la usabilidad, se ha mejorado el módulo existente de Explicabilidad de Inteligencia Artificial (XAI), el cual tiene como propósito ofrecer explicaciones sobre las decisiones automáticas del sistema para aumentar la confianza de los usuarios. Además, se han optimizado las decisiones automáticas al mejorar los procesos de agregación del sistema. Finalmente, se ha formalizado la arquitectura para facilitar futuros desarrollos. A través de estas mejoras, y con carácter general, se pretende disminuir la carga de trabajo del terapeuta y, en consecuencia, incrementar el tiempo de calidad que dedican a sus pacientes.

Acknowledgement

Quisiera dar las gracias a aquellas personas que me han apoyado, que me han formado y me han hecho crecer como persona. En primer lugar, quiero agradecer a mi familia, por su amor sincero y por haberme guiado y acompañado hasta llegar a este punto. En particular, quiero agradecer a mis padres y a mi hermana por su paciencia y comprensión, que me ha permitido llegar tan lejos. También quiero aprovechar esta oportunidad para recordar a mi abuelo Adrián y a mi tía Pili, que nos han dejado durante este último año. Siempre os llevaré en el corazón.

También quiero agradecer a los profesores que me han enseñado en la escuela, en el instituto y en la universidad. He tenido suerte de que muchos de ellos me han transmitido no solo su conocimiento, sino que también me han transmitido su pasión y han contribuido a mi formación como persona.

Finalmente, quiero agradecer al tutor de este proyecto, David, por haber confiado otra vez en mí para este proyecto.

Sergio

A mis padres y a mi hermana

Contents

List of Figures

List of acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Augmented Reality
ASD	Adaptive Software Development
CRUD	Create, Read, Update and Delete
DBMS	Database Management System
DSS	Decision Support System
ER	Entity Relationship
FIS	Fuzzy Inference System
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
IT	Information Technology
JSON	JavaScript Object Notation
LIME	Local Interpretable Model-agnostic Explanations
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
ORM	Object Relational Mapper
OWA	Ordered Weighted Averaging
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
TOWA	T-norm OWA
UL	upper limb
WOWA	Weighted OWA
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

This chapter introduces the Master's thesis project detailed in this document. First, a review of stroke from a medical and socioeconomic perspective is presented. Although the master's thesis is applicable to most neurological diseases, emphasis has been placed on stroke due to its prevalence. Then, the background in which the project has been executed is described. Next, the project proposal is outlined. The chapter concludes by providing a list of the remaining chapters in this document.

1.1 Overview of stroke

Stroke is one of the major leading causes of death around the world. Annually, approximately 6.5 million people worldwide die due to stroke, and up to 50% of stroke survivors have major disabilities [FBN+22, Don18]. While the elderly are the major demographic affected by stroke, it is noteworthy that young individuals can also be afflicted with stroke [GS11]. Since stroke can have significant sequels, rehabilitation emerges as a crucial component in addressing stroke, along with early detection and treatment. Rehabilitation can commonly takes from 60 to 90 days [DGB+21], and in the conventional approach to rehabilitation, constant supervision by health care professionals is essential.

Stroke, also known as cerebrovascular accident, is defined as the symptoms of focal loss of cerebral function due to poor blood flow [War98]. There are two types of stroke: ischemic stroke and intracerebral hemorrhage [SFA10]. Ischaemic stroke, the more prevalent type, is caused by insufficient blood flow in the brain, generally due to blood clots, leading to the death of brain cells [KX20]. On the other hand, intracerebral haemorrhage is characterised by damage to brain tissue caused by blood loss within the brain, commonly due to hypertension [KX20]. Stroke can result in a wide range of physical and cognitive disabilities, as well as psychological disorders. One of the most common physical disabilities resulting from a stroke is paresis, which involves paralysis or weakness on one side of the body, typically affecting the limbs and the face [HvLGZ02]. It is noteworthy that the side of the body impacted by the stroke is generally dependent on the hemisphere affected. In cases where the left hemisphere is affected, the right limbs and the right side of the face experience paresis, and vice versa [AWS+99]. Beyond paresis, stroke patients may encounter challenges related to balance and coordination, as well as difficulties in performing daily activities. Cognitive disabilities following a stroke can manifest in various ways. Memory problems are common, where individuals may have difficulty remembering recent events or retaining new information. It is also worth noting that stroke is a risk factor for dementia [KLM+18]. Language and communication can also be impacted, resulting in difficulties with speaking, understanding, or finding the right words [Dun94]. Psychological disorders can also occur following a stroke. Post-stroke depression is a common mental disorder, and it can manifest as feelings of sadness, hopelessness, and loss of interest in activities that were once enjoyable [NR02].

It is also worth examining not only the effects of stroke on patients' health, but also its socioeconomic impact on the patients' environment. Direct costs are associated with the diagnosis and treatment of the disease [MVI03]. Common components of direct costs include expenses related to transportation to the hospital, hospital stays, diagnostics, and treatment. However, direct costs continue after the immediate treatment has been provided. Home modifications may be required for stroke patients to increase patient independence, and make life safer at home. In addition, some patients cannot take care of themselves after stroke due to their disabilities. It is for this reason that the cost of dedicated carers should also be accounted for.

In many instances, care for stroke patients is provided by family members rather than professional caregivers. While there may not be a direct monetary cost when family members assume the caregiving role, the continuous strain of caring for stroke patients can lead to the development of the *Caregiver Stress Syndrome*. It is a condition characterised by physical, mental, and emotional exhaustion resulting from individuals neglecting their own health while focusing on caring for someone else [LSRC14]. This syndrome is one of the most relevant examples of indirect costs associated to stroke. Indirect costs are defined as potential resource losses as a consequence of stroke, and they include the loss of productivity and informal caregiving, as mentioned earlier. Given that stroke often occurs in later stages of life, the loss of productivity due to impairment is not as significant a cost as direct costs [MVI03].

Effective stroke rehabilitation relies on several key principles. One of the major predictors of success in stroke rehabilitation is whether patients stay motivated throughout the rehabilitation process. Another important aspect of rehabilitation is early commencement. Commencing rehabilitation promptly is associated with better outcomes [LBK11]. Finally, rehabilitation programs are most effective when they incorporate task-specific training [Dob04].Functional exercises such as climbing stairs, using a spoon, or getting up from the floor with a chair (see Figure 1.1) can significantly increase patients' quality of life.

There are two main approaches for performing rehabilitation: the traditional approach, conducted in a hospital or similar facility under the supervision of therapists, and home rehabilitation, where exercises are performed at home, often with the support of Information

Figure 1.1: Functional exercise: getting up from the floor with a chair

Technology (IT) systems. The traditional approach offers the advantage of therapists overseeing patient movements to ensure correct and safe exercise execution. However, this approach comes with associated costs, including transportation to a dedicated facility and the expenses associated with maintaining such facilities. In cases where rehabilitation occurs in a hospital setting, the costs of an extended hospital stay solely for rehabilitation purposes are greater than those associated with early discharge [AMR+00]. On the other hand, home rehabilitation presents several advantages. The major advantage of home rehabilitation is the reduced costs, but it is not the only one. The convenience of home rehabilitation is another significant advantage. This approach not only allows for greater flexibility, removing the dependence on therapist availability, but also eliminates the difficulties and stress associated with transportation. Particularly in low-/middle-income countries and rural areas, where access to dedicated rehabilitation facilities may be limited [BdAS+20, RPOD95], this method proves to be especially beneficial. Furthermore, engaging in rehabilitation at home imparts a greater sense of comfort and relaxation to the patients than a clinical setting [SW00]. Finally, the performance of rehabilitation activities in the patients' homes contributes to an enhanced level of independence. In particular, home rehabilitation has been linked with a positive effect on activities of daily living including self-care, and on mobility [KLO+20].

In the socioeconomic context previously outlined, home rehabilitation systems emerge as a solution to mitigate the impact of stroke. In particularly, a home rehabilitation system that allows therapists to delegate tasks can leverage the advantages of home-based rehabilitation, such as reduced costs and increased convenience, while simultaneously addressing the potential drawbacks associated with autonomous execution of exercises.

1.2 Project background

This project has been executed in the context of a web system called *Physio Galenus* (see Figure 1.2). This commercial system, currently under development, employs immersive technologies such as Augmented Reality (AR) to guide patients in performing rehabilitation exercises at home. Additionally, it integrates gamification techniques, including scores, to motivate patients and encourage consistency in their rehabilitation routines. *Physio Galenus* is designed to be offered as a service targeted towards therapists. Its value proposition lies in providing a user-friendly and cost-effective solution for home rehabilitation that can be adapted to diverse settings. It is primarily targeted towards therapists through a SaaS model. The system's requirements are minimal, needing only a laptop, an internet connection, and a standard webcam. This stands in contrast to alternative solutions that may require more expensive equipment, such as the Microsoft Kinect™[1], and as a result increase costs and installation complexity. The use of web technologies further facilitates adoption by eliminating the need for installations. To make the system more intuitive, the system adopts a gym metaphor, organising rehabilitation exercises into routines, each with specified sets and repetitions for individual exercises. In addition, the system incorporates a Decision Support System (DSS) whose purpose is to automatically adjust the rehabilitation routines that patients are assigned in order to reduce therapists' workload and increase the quality time that they spend with patients. The system comprises three distinct applications: the client application for patients, the client application for therapists, and the server.

Physio Galenus is being developed by Furious Koalas[2], a company that works in areas such as gamification and serious games. The author of this project started working in this company as an intern, and he is involved in the development of *Physio Galenus*, with a focus on the server. The author's previous bachelor's thesis centered around creating a DSS that autonomously adjusted patients' routines according to the execution data. This system also incorporated an XAI module for providing explanations.

1.3 Project proposal

The project described in this document focuses on making the system more intuitive for therapists, and increasing the degree to which therapists can delegate to the system. Although

[1] https://learn.microsoft.com/es-es/windows/apps/design/devices/kinect-for-windows

[2] https://www.furiouskoalas.com/

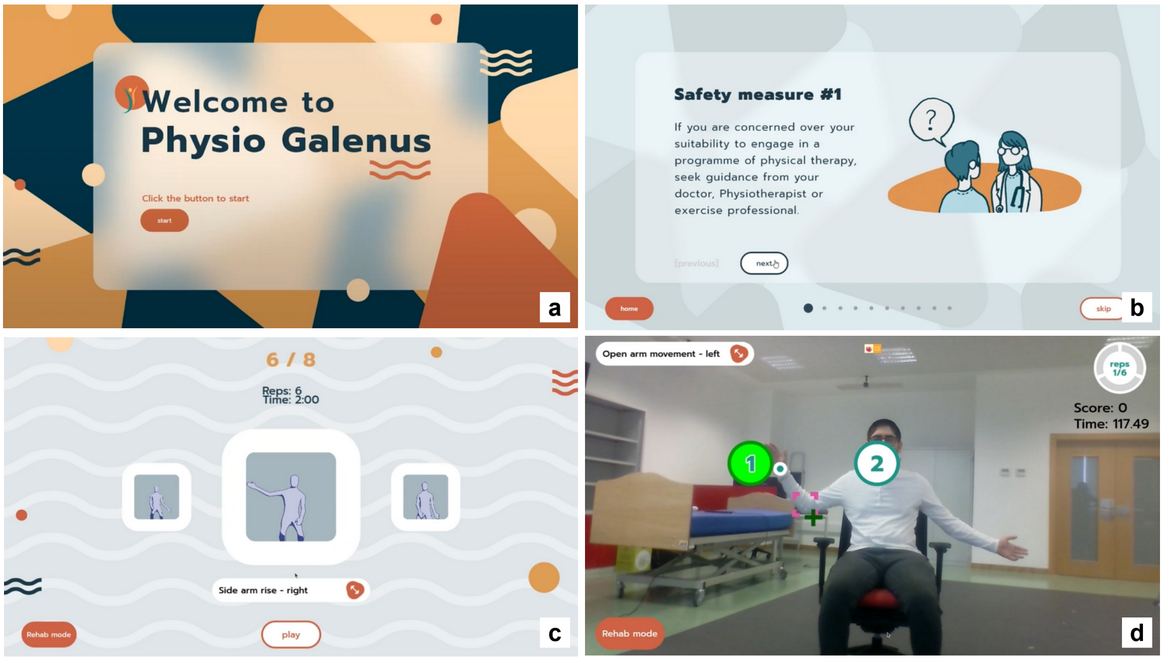


Figure 1.2: Screenshots taken from the patient's application. a) Presentation screen. b) Tutorial on usability and security. c) Rehabilitation exercise selector. d) Main screen for performing physical rehabilitation.

the bachelor's thesis' prototype provided the functionality for guiding patients' movements when performing the exercises and automatically adjusting rehabilitation routines, there was major room for improvement. In particular, its usage was not very intuitive, and the level of delegation for therapists is limited. A significant challenge in home rehabilitation arises from the lack of direct supervision by therapists while patients perform exercises. Although autonomous exercise execution brings benefits, it is essential to ensure that patients perform exercises correctly. One of the major concerns in rehabilitation are compensatory movements, which are pathological movements that reduce the effectiveness of rehabilitation. The inability to detect compensatory movements, limits therapists' ability to delegate effectively. An objective of the project is to address this limitation by developing the capability to detect compensatory strategies employed by patients, and incorporate this information in the decision logic of the DSS.

Another challenge that complicates the delegation process is the lack of transparency of automatic decisions. In the bachelor's thesis, an XAI module was integrated to provide explanations for the decisions made by the DSS. However, a flaw identified in the thesis was the excessive length of these explanations. The extended length of explanations poses a potential challenge for therapists, as spending too much time reviewing them to ensure the adequacy of the routine could negate the benefits of automatic adjustment. To address this, the current project aims to enhance the XAI module to produce more concise explanations. If explanations are more concise while conveying a similar amount of information, therapists can use the system more simply and intuitively.

An additional issue identified in the previous work was related to the aggregation strategies employed by the DSS. In particular, a step of the DSS involves aggregating an output value called *performance increment*, which determines the level of performance of patients. The arithmetic mean, which was previously used, does not consider the data distribution of the aggregated information. This becomes crucial when adjusting the routine, as a conservative approach is desired where more weight is given to lower performance scores in the aggregation process. To address this, diverse aggregation operators were explored, and the aggregation strategy of the DSS was revamped. Additionally, the insights gained from the study of aggregation operators also contributed to the new developments of the project.

Finally, the architecture of the system has been formalised, providing a precise and unambiguous description of its structure and behaviour. This formalisation makes the system easier to reason with in future work. This is especially important considering the goal of commercialising the system. Having a formalised architecture will facilitate future developments, ease the integration of new developers, and contribute to the system's overall maintainability [MNR+21].

1.4 Document structure

The rest of the document has been structured into the chapters listed below:

Chapter 2: State of the art
In this chapter, the theoretical foundations and prevailing techniques related to OWA operators, compensation movement detection, and XAI are explored.

Chapter 3: Objectives
This chapter presents the general objective, as well as the specific objectives of the project.

Chapter 4: Methodology
In this chapter, the methodology and resources used during the project are detailed.

Chapter 5: Architecture
In this chapter, the architecture of the system is described, as well as the problems that arose during the project, and the justification for the solutions that were implemented.

Chapter 6: Results
In this chapter, the final results of the project are presented.

Chapter 7: Conclusions and future work
In this chapter, the degree to which the objectives were achieved is evaluated. Additionally, possible future work and improvements are discussed.

Chapter 2

State of the art

2.1 Aggregation Operators

The aggregation of several input values into a single output value is a fundamental concept in many fields, including Mathematics, Physics, Engineering and other sciences. Specifically, aggregation operators are a major part of a wider concept called information fusion, and they are used in many tasks, such as noise reduction, accuracy improvement, information summarization and extraction, and decision-making [TN07]. Within the field of Computer Science, there have been many applications in disciplines such as Artificial Intelligence (AI) and decision-making. Some examples include fusion of data provided by sensors in robotics, multicriteria decision making, fusion of images in computer vision, and ensemble methods in data mining [TN07]. The values to be aggregated are generally real numbers, since many real-world data and measurements involve real numbers, but they can also be more complex mathematical objects, such as probability distributions [Yag17] or fuzzy sets [Zad65]. If the number of input values is fixed, then an aggregation operator can be defined as a mapping of n variables. In general, the number of input values may not always be fixed. Therefore an aggregation operator is defined as a mapping $A : \bigcup\limits_{n\in\mathbb{N}} X^n \to X$ [CMM02]. They may also be defined as a family of n-ary operators, for $n \in \mathbb{N}$. The most common aggregation operators include the arithmetic mean, the geometric mean, the maximum and the minimum [CMM02].

It is worth discussing a few properties of aggregation operators. Some of these properties can only be met by those aggregation operators that aggregate values that have an order relation, such as real numbers. Firstly, aggregation operators are bounded, meaning that the bounds of the domain and of the range are preserved. This means that $A(0, ..., 0) = 0$ and $A(1, ..., 1)$. Next, an aggregation operator is monotonic if the order of the aggregated values is preserved. Formally, $x_1 \leq y_1, ..., x_n \leq y_n \Rightarrow A(x_1, ..., x_n) \leq A(y_1, ..., y_n)$. An aggregator operator is said to be symmetric if the aggregated value is equal for all permutations of the input values. Finally, an aggregation operator can be described as idempotent. While the notion of idempotency is originally from binary operators, it can be extended to n-ary operators. An aggregation operator is idempotent if $A(x, x, ..., x) = x$ [CMM02].

One of the most commonly used aggregation operators in Computer Science, and particularly in Fuzzy Inference System (FIS), are the Ordered Weighted Averaging Aggregation Operators [Yag88]. They were introduced by Ronald R. Yager, and they are a family of mean operators that lie between the *and* operator and the *or* operator. OWA operators of order n are defined as a mapping $F : U^n \rightarrow U$, with U being the universe set of the values to be aggregated ($U = [0, 1]$ without loss of generality). F is associated with a weighting vector $W = \begin{bmatrix} W_1 & W_2 & \dots & W_n \end{bmatrix}^T$, such that $W_i \in [0, 1]$ and $\sum_i W_i = 1$ Therefore, $F(a_1, a_2, \dots, a_n) = W_1 b_1 + W_2 b_2 + \dots + W_n b_n = W^T B$, with b_i being the ith largest element in $(a_1, a_2, \dots, a_n)$. It can be shown that OWA operators are bounded, monotonic, symmetric, and idempotent [Yag88].

Weighted averages are a family of aggregation operators that is similar to OWA operators. The main difference lies in how the weights are assigned to values. In a weighted average, each weight is assigned an input value. However, in an OWA operator, weights are not assigned to input values but to ordered positions of input values. Additionally, OWA operators are a generalisation of other aggregation operators. Most notably, the arithmetic mean can be expressed as a OWA operator where every element of the weight vector is $W_i = \frac{1}{n}$. Additionally, the min operator can be expressed as an OWA operator with $W_i = n$ if $i = n$ and $W_i = 0$ otherwise. Similarly, the max operator can be expressed as an OWA operator with $W_i = n$ if $i = 1$ and $W_i = 0$ otherwise. These two extreme cases of OWA operators are the cases where OWA operators behave like a pure *or* or *and* operator, respectively. In general, it can be observed that the closer the total weight is to being in W_1, the closer the OWA operator is to being a pure *or* operator. A measure of the orness of an OWA operator can be calculated as follows:

$$orness(W) = \frac{1}{n-1} \sum_{i=1}^{n} ((n - i) * W_i)$$

According to this definition, the orness of the min operator is 1, the orness of the max operator is 0, and the orness of the arithmetic mean is $\frac{1}{2}$. In addition, the andness of an OWA operator is defined as the complement of the orness: $andness(W) = 1 - orness(W)$.

Another useful metric for OWA operators is the dispersion metric [Yag88]. This metric evaluates the extent to which the input values are used in the aggregation. For example, $W_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ and $W_2 = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$ have the same orness, but W_1 only considers the third largest value. The dispersion of a OWA operator is calculated as follows:

$$dispersion(W) = -\sum_i W_i \ln(W_i)$$

The dispersion is minimum if there is a i such that $W_i = 1$, and its value is 0. It is maximum for the OWA operator equivalent to the arithmetic mean, and its value is $\ln(n)$.

As seen previously, OWA operators are a generalisation of other aggregation methods. However, there are other aggregation operators that generalise OWA operators. Namely, OWA operators are a particular case of Choquet integrals [MS93] Choquet integrals belong to a family of aggregation operators called fuzzy integrals or nondadditive integrals. Fuzzy integrals are based on the concept of capacity or fuzzy measure [Cho54]. Fuzzy measures are similar to classical measures, with the only difference being the additive property. In classical measures, the additive property states that the measure of the union of disjoint sets is equal to the sum of their individual measures. For example, the length of two segments is equal to the sum of the length of the segments. In the case of fuzzy measures, this property does not necessarily apply. Instead, the additive property is replaced by the monotonicity property, which is a weaker constraint. Let $N = U^n$, a fuzzy measure on N is defined as a mapping $\mu : 2^N \to [0, 1]$, satisfying the following two properties:

- $\mu(\emptyset) = 0, \mu(N) = 1$ (The normalisation property)
- $A \subseteq B \to \mu(A) \leq \mu(B)$ (The monotonicity property)

Let μ be a fuzzy measure on N and $f : N \to [0, 1]$, the Choquet integral is defined as follows:

$$\mathcal{C}_\mu(f) := \sum_{i=1}^{n} (f_{(i)} - f_{(i-1)})\mu(A_i)$$

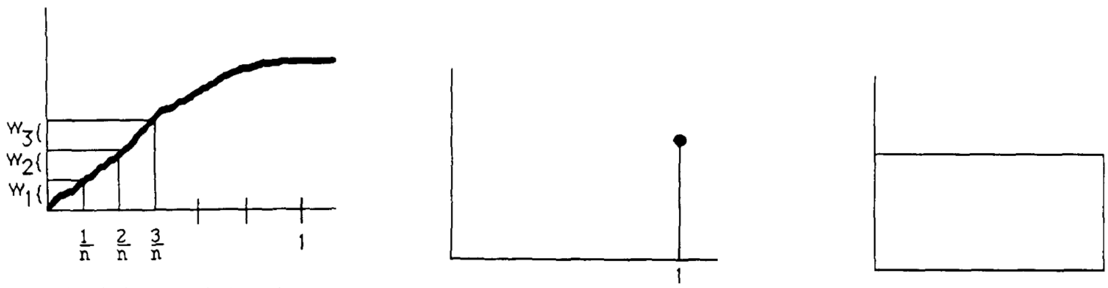
With $f(i) = f_i$, $f_{(i)}$ being the ith smallest value of f, $A_i := \{(i), \ldots, (n)\}$, and $f_0 := 0$. The Sugeno integral can be defined similarly:

$$\mathcal{S}_\mu(f) := \bigvee_{i=1}^{n} (f_{(i)} \wedge \mu(A_i))$$

Both Choquet [Cho54] and Sugeno [Sug74] integrals have been widely used in Multi-Criteria Decision Aid under uncertainty. Choquet integral operates on real numbers, generally in the range $[0, 1]$, and it has been proven to be a general version of many other aggregation operators, including the OWA operator or the weighted average [Mar00]. When dealing with ordinal variables, the Sugeno integral is used rather than the Choquet integral. This is because arithmetic operations such as addition cannot be used on ordinal variables. Since the Sugeno integral only depends on the min and max operators, it can aggregate ordinal variables without converting them into cardinal numbers [GL09]

Since nonadditive integrals encompass OWA operators and many OWA variants, it may seem that it would be best to refer to OWA operators as nonadditive integrals. However, OWA operators have a clear interpretation, and they provide meaningful families of operators [Gra11]. One of the main advantages of OWA operators is the possibility of using them for defining quantifiers. This is based on Zadeh's idea of linguistic quantifiers [Zad83]. Examples of these quantifiers include 'almost all', 'few', 'many', 'most', ... In the context of decision making, OWA operators are used to aggregate the satisfaction level of a candidate solution of different decision criteria into a single satisfaction level. These aggregations can be defined by using a linguistic quantifier. For example, the linguistic quantifier 'most' can be constructed by selecting the weight vector where all but the last weights are non-zero. Similarly, the linguistic quantifier 'few' can be defined by the weight vector where the values of the weights are concentrated on the first weights. Finally, the linguistic quantifiers 'any', and 'for all' would correspond to the min and max OWA operators.

The main challenge when defining an OWA operator lies in the selection of the weight vector. It is for this reason that much research has been devoted to the study of optimal strategies for selecting the appropriate weights. In [Yag93], Yagger defined the *natural* method of generating the OWA weight vector from linguistic quantifiers. Weights are calculated as: $w_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n})$, where $Q : [0,1] \to [0,1]$ is a regular nondecreasing quantifier. It can be seen in 2.1 how the quantifiers *For all* and *There exists* are represented with a function.



(a) Obtaining weights from a quantifier

(b) The quantifier *For all*

(c) The quantifier *There exists*

Figure 2.1: Examples of quantifiers for natural OWAs

Another approach for generating the weights of an OWA operator is to use a parameterised family of OWA operators. These are families that are specified by a few parameters which can be used for generating the weights. The first parameterised family was introduced by O'Hagan [O'H87], and it was called the Maximum Entropy OWA. It has a parameter α that specifies the desired level of orness. The parameterised family contains the OWA operators with the highest level of dispersion for any given α. Yager introduced a family of OWA operators [YF94]: the S-OWA. It has two variants, an orlike variant and an andlike variant. Orlike S-OWA is specified by a parameter $\alpha \in [0,1]$

$$w_i = \begin{cases} \frac{1}{n}(1-\alpha)+\alpha, & \text{if } i = 1 \\ \frac{1}{n}(1-\alpha), & \text{if } i = 2,\dots,n \end{cases}$$

When $\alpha = 0$ the resulting OWA operator is the arithmetic mean, whereas when $\alpha = 1$ the obtained OWA is the maximum. The orness of this family is determined by $\frac{1}{2}(1+\alpha)$, ranging from 0.5 to 1. Similarly, the weights of the andlike S-OWA are as follows

$$w_i = \begin{cases} \frac{1}{n}(1-\beta), & \text{if } i = 1,\dots,n-1 \\ \frac{1}{n}(1-\alpha)+\beta, & \text{if } i = n \end{cases}$$

When $\beta = 0$ the resulting OWA operator is the arithmetic mean, whereas when $\beta = 1$ the obtained OWA is the minimum. The orness of this family is determined by $\frac{1}{2}(1-\beta)$, ranging from 0.5 to 1.

In addition to parameterised families of OWA operators, one of the major strategies for generating the weights is learning the weights from data. In [FY98], Filev and Yager introduced a method utilising this approach. Given tuples with the input values to be aggregated, along with the estimated aggregated values, the aim is to minimise the error between the estimated aggregated value and the result of applying the OWA operator to the input values. This is achieved by using the gradient descent technique for optimising the the instantaneous error.

Many aggregation operators have emerged as variations of the OWA operator. The T-norm OWA (TOWA) operator was proposed by Yager [Yag05], and it is based on the idea of merging OWA operators and t-norms. The OWA aggregation is reformulated as $A(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i Min[H_i]$, with $H_i = \{a_{index(j)} \mid j \in [1, i]\}$. The TOWA operator is proposed as a generalisation of the OWA operator when a t-norm operator other than the minimum is used. Torra proposed combining the weighted mean and OWA operators into a new aggregation operator called Weighted OWA (WOWA) operator. To specify a WOWA operator, two weight vectors are needed. One vector is associated with the input values, similar to the weighted mean, and the other weight vector associated with the ordered position of the input values, similar to the OWA operator. When aggregating values, both weight vectors are combined into a single weight vector. In [YF99], Yager introduced the induced OWA operator. It is based on the idea of generalising the ordering step in the traditional OWA operator. In the induced OWA operator, each input values is associated with an order inducing variable, and in the ordering step input values are not ordered based on their own value but on the value of their order inducing variable. Formally, $A(\langle u_1, a_1\rangle, \ldots, \langle u_n, a_n\rangle) = W^T B$, where b_i is the a value of the OWA pair having the ith largest u value.

Finally, there are some variations of the OWA operator, that instead of aggregating real numbers, aggregate fuzzy sets. In [MS00], Mitchell et al. introduced algorithms for directly and indirectly sorting fuzzy numbers, which are a particular case of fuzzy sets. This strategy is based on a sorting algorithm for crisp numbers using a permutation matrix, and fuzzifying the equality and minimum operations. Using these algorithms, they extended the OWA operator and the induced OWA operator by replacing the sorting step with the new sorting algorithms and fuzzifying the summation. In [ZCJG08], Zhou et al. introduced an OWA operator called type-1 OWA operator. In the type-1 OWA operator, both the input values, the weigths and the aggregated value are fuzzy sets. The membership function of the aggregated value is the following:

$$\mu_G(y) = \sup_{\sum_{k=1}^{n} \bar{w}_i a_{\sigma(i)} = y} \left(\mu_{W_1}(w_1) * \cdots * \mu_{W_n}(w_n) * \mu_{A_1}(a_1) * \cdots * \mu_{A_n}(a_n)\right)$$

where $a_i \in U$, $w_i \in [0,1]$, $\bar{w}_i = w_i / \sum_{j=1}^{n} w_j$ and $\sigma : \{1,\dots,n\} \rightarrow \{1,\dots,n\}$ is a permutation where $a_{\sigma(i)}$ is the ith largest element in $\{a_1,\dots,a_n\}$. Additionally, in [ZCJG11] Zhou et al. introduced an optimised version of the algorithm based on α-cuts that aggregates fuzzy numbers specifically.

The previous OWA operator variations that aggregate fuzzy sets are based on fuzzifying crisp operations. The idea of extending crisp operations to fuzzy sets is called the extension principle, and it was first introduced by Zadeh [Zad75]. The extension principle allows a mapping to be extended from points in U to fuzzy sets of U. If a fuzzy set is denoted as $A = \mu_1 u_1 + \cdots + \mu_n u_n$, where the membership value of u_i is μ_i, then the extension principle states that $f(A) = f(\mu_1 u_1 + \cdots + \mu_n u_n) \equiv \mu_1 f(u_i) + \cdots + \mu_n f(u_n)$. In addition, if the function to be fuzzified is a n-ary function, then the fuzzified version must operate on a fuzzy relation A. In these cases, the membership function of A is usually constructed from the membership functions of the individual fuzzy sets: $\mu_a(u_1,\dots,u_n) = \mu_{A_1}(u_1) * \cdots * \mu_{A_n}(u_n)$.

2.2 Techniques for the detection of compensatory movements

After experiencing a stroke, one of the most prevalent and difficult consequences is the limitation in upper limb (UL) function, significantly impacting the patient's independence in daily activities and potentially resulting in enduring disability [CL00]. When stroke patients attempt to move and encounter the movement limitations, the typical response is to adjust using the motor strategies that are accessible. Compensatory movements are pathological motor strategies that are developed by stroke patients in substitution for healthy movement patterns. Compensatory movements are characterised by decreased movement speed, increased movement variability, increased movement segmentation, and spatial and temporal incoordination between adjacent arm joints [CL00]. Compensatory movements can present themselves in multiple ways. One of the most common compensatory movements are pathological synergies between muscles. Gross extensor synergies consist in shoulder extension and adduction combined with elbow extension, forearm pronation and wrist flexion. Gross flexor synergies consist in shoulder flexion and abduction combined with elbow flexion, forearm supination and wrist extension (see Figure 2.2). There is some controversy among therapists in regards to whether gross flexor and extensor synergies precede the recovery of more complex motor movements. Some therapists argue that basic limb synergies should be encouraged during the early recovery stages the stroke [Twi51], while others favour early attempts to develop normal motor function [CS87].

An additional compensatory movement is excessive trunk displacement (see Figure 2.3). Although healthy subjects may displace the trunk for trunk-assisted reaching, the amount of trunk displacement is related to the severity of the motor deficit. It has been proven that training the reaching ability with different forms of trunks restrain leads to decreased compensatory trunk movement and improved UL kinematics [LLPB15]. Finally, another

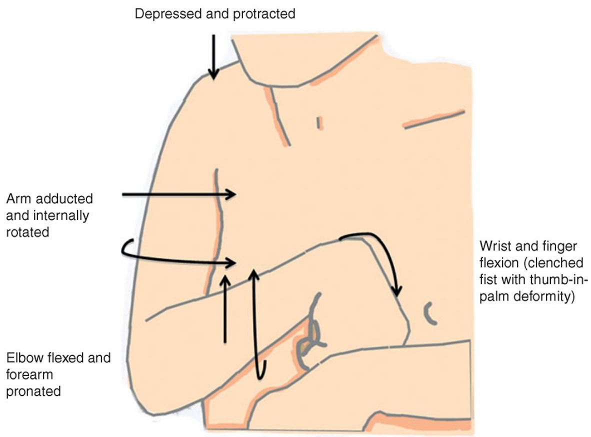


Figure 2.2: Gross flexor synergy diagram. Image obtained from [TV14].

common compensatory strategy is the fixation of specific body segments. Fixation of the pelvis, the lumbar spine, the scapula or the thorax may reduce the number of motor elements to be controlled to achieve the motor task.

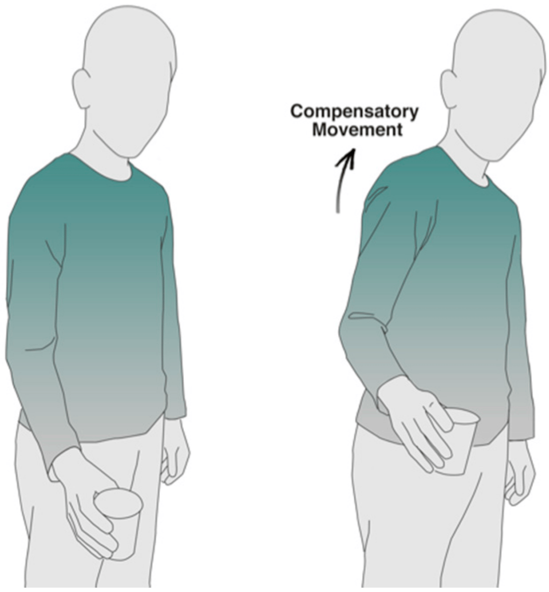


Figure 2.3: Excessive trunk displacement. Image obtained from [WDBT$^+$17].

There have been multiple effort for increasing the effectiveness and reducing the costs of stroke treatment with the use of IT [NPH13]. One of those efforts consists in using computer vision to discriminate between UL functional improvements due to true motor recovery and those due to compensatory movements. When applied in the context of a rehabilitation facility, detection of compensatory movements can potentially enhance the effectiveness of therapy by complementing therapists in the detection of compensatory strategies. When applied to home rehabilitation, it can augment rehabilitation independent of the constant care of therapists, therefore facilitating a more autonomous use of home rehabilitation [WFY$^+$22].

In [LLPB15], Levin et al. tracked patients' movements using a 3-camera optoelectronic system. Patients were asked to point at specific targets without touching them while seated. Reflective markers were placed on the shoulder, elbow, and wrist to measure movements, while trunk displacement was tracked with additional markers on a rigid exoskeleton. The onset and offset of hand and truck movements was analysed offline. The study focused on analysing arm-plane movement as a measure of upper limb flexor synergy. Additionally, movement quality was assessed by using 5 spatial kinematic variables, one temporal variable, and movement error.

In [CLZ+19], Cai et al. used multiple sensors for the detection of compensation movements. The patients' pressure distribution was recorded while they performed seated reaching tasks. In order to record pressure distribution, a pressure distribution mattress was used. Additionally, trunk activity was monitored by tracking 9 trunk muscles using surface electromyography signals. Five features were extracted from the pressure distribution data, and the surface electromyography signals were preprocessed and aggregated using the root mean square. Two classification algorithms, k-nearest neighbours and Support Vector Machine (SVM), were employed to categorise movement patterns based on the extracted features.

In [RWGB17], Ranganathan et al. used wireless wearable devices to identify compensatory movements. Patients performed tasks simulating everyday activities that targetted different joint motions in the shoulder, elbow, and wrist. Three wearable devices, equipped with accelerometers and gyroscopes, were worn in the trunk, upper arm and forearm. Standard deviations of trunk acceleration and elbow angular velocity were calculated to characterize individual movement strategies for each task. Two classification algorithms, naïve Bayes and sequential minimal optimisation, were tested for the identification of compensation movements.

In [ZLL+18], Zhi et al. used a Microsoft Kinect™ v2 sensor for the automatic detection of compensation movements during rehabilitation exercises. The Microsoft Kinect™ sensor provides a colour stream and a depth stream. The Toronto Rehab Stroke Posture dataset was used, which is a dataset containing 3D skeletal joint positions of compensatory movements. Additionally, an expert rater labeled color frames for compensations based on visual examination. The Kinect™ skeletal tracking information was preprocessed using a Savitzky–Golay filter to reduce the effect of unstable tracking. Frames were classified as *compensation* or *no compensation* by using a SVM, and a recurrent neural network with long short-term memory.

In [DEP+23], Das et al. presents a pilot study on the online detection of compensatory strategies in human movement using supervised classification. Their approach involves employing a linear classifier based on energy-based features to distinguish between healthy and compensatory movements and identify the specific joints involved in compensation. The authors employed a marker-based motion capture system to track the movements of patients while they performed various reaching tasks. A dataset was consisting of instances with both healthy and compensatory movements was created for training the classifier. Notably, the results demonstrate the superiority of the proposed approach over deep learning classifiers that do not incorporate energy-based features.

They propose using a linear classifier from energy-based features to discriminate between healthy and compensatory movements and identify the compensating joints. The results show that the proposed approach outperforms deep learning classifiers that do not use energy-based features. various reaching tasks They used a marker-based motion capture system for tracking the movements of patients, and they created a dataset with healthy and compensatory movements.

2.3 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) is a subfield of AI focused on making the behaviour of AI systems more intelligible [PCRM+20]. This is achieved by providing explanations about the decisions taken by AI systems. XAI models are called *glass box models*, in contrast to *black box models* because they offer transparency and interpretability in their decision-making processes. It is worth noting that there is a trade-off between the learning performance of AI techniques and their inherent explainability: the higher the learning performance, the lower the inherent explainability (see Figure 2.4) [GSC+19]. One of the main reasons why XAI is important is because trust in the decision-making process of AI is a mayor factor in the introduction of AI systems in critical fields. Additionally, XAI systems may be able to maintain accountability in AI systems, meaning that the system can be held responsible for its actions [LGHJC22]. One such critical field where XAI is particularly crucial is defence [SLNS18]. Defence systems that incorporate AI techniques have to make decisions that may lead to loss of human lives. In particular, there is a heated debate regarding the use of autonomous weapons, which are weapons that do not need of human intervention to operate and engage targets with force [Etz18]. The most common type of autonomous weapon are anti-vehicle and antipersonnel mines. These operate based on a trigger system, and may not need of AI techniques. Missile defence systems and sentry systems do rely on AI for target detection and aiming, but are generally limited to defence. Finally, there is loitering munition, also known as suicide drones. They are generally used offensively, and also rely on AI for autonomous operation. This type of autonomous weapons has recently raised criticism due to their heavy use in Russia's 2022 invasion of Ukraine [Kun23].

Another critical field where XAI is critical is law [WV18]. Unlike in the case of defence, human life itself is generally not threatened. However, there is the risk of AI systems issuing unfair rulings. One of the main concerns with the application of AI in court is that AI systems may perpetuate biases and inequities present in the existing legal system. Additionally, discrimination by AI systems is more abstract, subtle and difficult to detect than discrimination

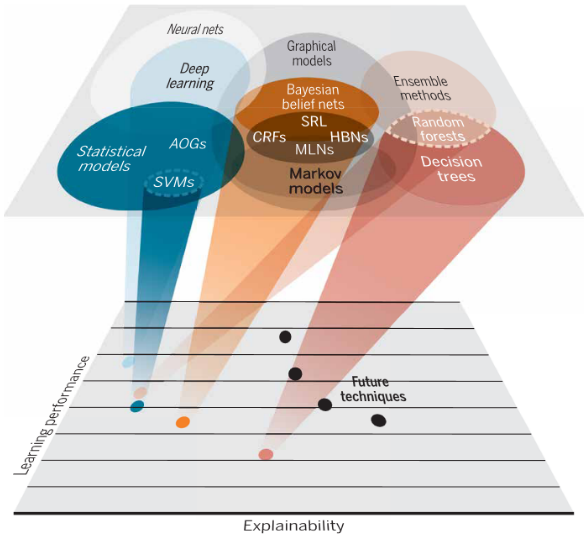


Figure 2.4: The learning performance and explainability of various AI techniques. Image obtained from [GSC+19].

by humans [WMR21]. XAI is not only concerned about upcoming AI applications such as autonomous trial rulings. There are current applications in the field of law that benefit from XAI. Firstly, AI is being used to assist in legal research and highlight legal issues for attorneys. Machine Learning (ML) is also being used to review contracts en masse. Finally, AI is used for predicting legal outcomes. This allows attorneys to weigh the strength of client arguments and the legal position in a lawsuit [Sur19].

Finally, one of the fields where XAI is promising is in the field of healthcare, since it is a critical field and there are multiple applications for AI in healthcare [CPXB23]. One of those applications is the creation of a DSS for disease diagnosis. DSS may either partially or completely automates the diagnosis process for a selection of diseases to facilitate the work of physicians. Another application is image classification [NDA23]. Medical imaging such as X-ray scans or magnetic resonance imaging can be automatically processed by AI systems for the detection of diseases like pneumonia, or tumours [SBV+22]. Image classification is generally achieved with ML, and the resulting explanations are in the form of heat maps that specify which parts of the image the AI system used for the decision (see Figure 2.5). Another task that benefits from XAI is clinical data interpretation, which goes extends medical images and it include the interpretation of diverse datasets such as electronic health records. Additionally, it can be used for the interpretation of abnormalities in 1-dimensional biosignals such as electrocardiogram signals [LOS+22]. Finally, it is worth noting that the validity of explanations obtained through XAI is a subject of ongoing debate. Some argue that explanations in XAI may contain both useful and non-useful information, and that misguided predictions can be provided with explanations that appear reassuring [GORB21].

XAI methods can be classified according to their scope. If the interpretation is generated as a result of the inherent characteristics of the AI method used, it is considered to be an

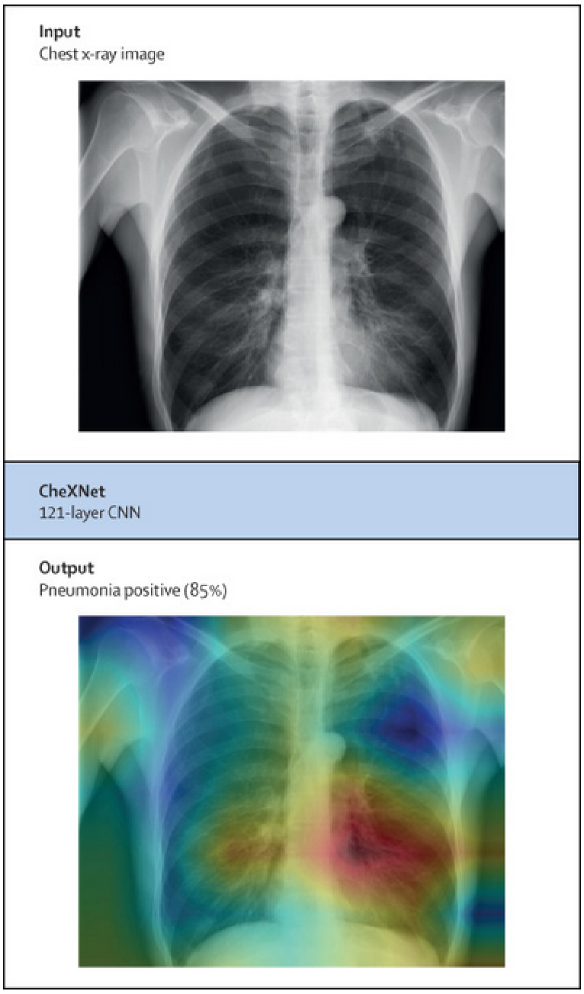


Figure 2.5: XAI applied to medical imaging. Image obtained from [RIZ+17].

intrinsic XAI method. However, if the interpretation is generated as an additional step it is a posthoc method. If the interpretation refers to the whole logic of the model, it is a XAI global method. Conversely, if the explanation is focused on a specific decision, it is classified as a local method Finally, if the XAI method is limited to a particular AI technique, it falls under the category of model-specific method. In contrast, if the XAI method is not tied to a specific model, it is classified as an agnostic technique [PCRM+20].

In addition to classifying the technique used by XAI methods, the explanations generated by the XAI methods can also be categorised. Some explanations are in the form of feature summary, where summary statistics are provided for each feature. This category of explanations includes feature importance. In the case of intrinsically interpretable models, the explanations are model internals. The methods that provide model internals as an explanation are by definition model-specific. Data point explanations achieve interpretability by providing examples that, in order to be useful, need to be meaningful themselves. Finally, some explanations rely on surrogate intrinsically interpretable models, where black box models are approximated with interpretable models [CPC19].

One of the most popular XAI methods is the Local Interpretable Model-agnostic Explanations (LIME) method. LIME is an algorithm that can provide explanations for any classifier or regressor by approximating it locally with an interpretable model. Since it can explain predictions for any classifier or regressor, it is considered a model-agnostic XAI method. It works by selecting a point in the dataset to be interpreted, and perturbing it by making small changes. In this way, a new dataset is created, and predictions for the new dataset are obtained. This new dataset is locally faithful to the classifier, and it is used to generate an interpretable model [RSG16].

Another popular model-agnostic XAI technique is the SHapley Additive exPlanations (SHAP) method. The SHAP method employs Shapley values, which are derived from cooperative game theory. In the context of AI, features are considered players, and the prediction is the outcome of the game. It is based on determining the degree to which each features contributes to the result. First, a single data point is selected to perform the Shapley value calculations. Then, for each feature, the marginal contribution of that feature to each feature subset is calculated. This involves evaluating the model's prediction with and without the feature for each feature subset. The average of all the marginal contributions is the Shapley value [LL17].

As an example of an application of XAI in healthcare, in [IHR$^+$22], Islam et al. enhanced a ML stroke prediction system with XAI techniques. They used electroencephalography signals for the prediction of ischemic stroke. After preprocessing the signal and normalising the features, multiple ML algorithms were tested. In particular, Adaptive Gradient Boosting, XGBoost, and LightGBM were used. LIME and the Eli5 library[1] were used to study feature importance in all three models. Both methods emphasised the spectral delta and theta features as local contributors to stroke prediction. The important features remained generally consistent across all classification methods used.

Finally, it is worth discussing fuzzy logic as a tool for integrating explicability in AI applications, such as DSS. Fuzzy logic has the potential to address the interpretability challenge in XAI. Unlike some existing techniques that struggle with clear crisp boundaries, such as LIME, fuzzy logic can generate if-then rules with linguistic labels [CHSO18]. These rules, expressed in a language easily understood by humans. A straightforward approach to employing fuzzy logic for XAI involves the usage of FISs for the implementation of intelligent applications such as expert systems. In addition, fuzzy logic can be used to provide a transparent and interpretable framework for explaining less interpretable AI models. For instance, fuzzy logic can be integrated with ML algorithms to create hybrid systems that not only offer high-performance capabilities but also enhance explainability [CHSO18].

[1] https://eli5.readthedocs.io/en/latest/overview.html

Chapter 3

Objectives

In this chapter, the objectives of the project will be discussed. Firstly, the general objective will be presented. Then, the general objective will be divided into specific objectives that support the general objective.

3.1 General objective

The focus of this master's thesis is to refine and expand the work done in the bachelor's thesis by making the system more intuitive for therapists, and by increasing the degree to which therapists can delegate tasks to the system. These have been selected as the objective of the master's thesis because the more tasks therapists can delegate to the system, the more quality time they will be able to spend with their patients. Additionally, they would also engender trust in the system among therapists.

During the project, the functional capabilities of the system will be enhanced, placing it within the context of a broader architecture. This new architectural not only will facilitate current improvements, it will also lay the foundation for future developments. This foresight is particularly important because there are plans to commercialise the system. It is precisely due to the commercial nature of the system that features not yet present on other commercial systems are being incorporated.

3.2 Specific objectives

After establishing the general objective of the project, the specific objectives that support it are defined.

- **Study and implementation of techniques for data aggregation in the context of automatic decision-making.** In the DSS, a crucial step involves aggregating multiple partial outputs into a single final output. It is important to consider the data distribution for performing an aggregation that aligns with the strategy of the DSS. To achieve this goal, a comprehensive study of multiple aggregation operators will be conducted, placing a particular emphasis on OWA operators. Subsequently, the aggregation process of the DSS will be improved by implementing the aggregation techniques.

- **Design and development of a module for the system that takes into account the compensatory movements of the patient.** One of the major obstacles to further delegation by therapists is the detection of compensatory strategies by patients. These strategies undermine the effectiveness of rehabilitation and must be considered when adjusting patients' routines. Therefore, a key objective is to integrate a new module for detecting compensatory movements. In addition, the DSS should be adapted to account for the compensation level of patients.
- **Design and implementation of an XAI algorithm focused on generating concise explanations.** One of the factors that facilitates delegation in the system is trust in its automated actions To engender trust in the DSS, it is advantageous for the system to provide explanations for its decisions. In the bachelor's thesis, an XAI module was developed to generate intelligible explanations. However, its main drawback was that the explanations were excessively long, compromising their effectiveness. Therefore, a key focus of the current project is to provide more concise explanations. This will increase therapists' trust in the system, as well make the system's usage more intuitive.
- **Formalisation and modularization of the architecture of the remote rehabilitation system.** The formalisation of an architecture consists in providing a precise and unambiguous description of its structure and behaviour. A formalised architecture will enhance the system's comprehensibility, making it easier to reason about its design and functionality. Therefore, a formalised architecture will provide a foundation for making informed design decisions, facilitating the integration of new developers into the project, and contributing to the overall maintainability of the system.

Chapter 4

Methodology

In this chapter, the methodology employed throughout the project will be presented. First, the development methodology used will be described. Then, the work schedule followed during the project will be presented. Finally, the resources used during the project execution will be enumerated.

4.1 Development methodology

The chosen development methodology plays a crucial role in any software development project, and this is especially true for projects involving AI techniques because they have an inherent degree of uncertainty. It is difficult to predict whether one technique will provide the desired results, and the time it will take to execute tasks. It is for this reason that an agile methodology is preferable to a traditional methodology, and in particular an agile methodology that is specialised for uncertain environments

Agile methodologies prioritise adaptability over predictability, and its main principles were outlined in the *Manifesto for Agile Software Development*[1]:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Adaptive Software Development (ASD) was selected as the development methodology for this project due to its suitability for high-changing and unpredictable projects. Among the major agile methodologies [Awa05], Scrum and Extreme Programming were ruled out because they heavily rely on meetings and roles, which are not suitable for a single-person project. While other developers have been involved in the development of the complete system, no other developer worked concurrently during the execution of this project.. Feature-Driven Development was discarded due to its emphasis on detailed documentation and upfront planning. In contrast, ASD is known for its lightweight nature, low documentation requirements, and adaptability, making it well-suited for projects with dynamic and unpredictable require-

[1] `https://agilemanifesto.org/`

ments. Additionally, it could easily be adapted to single-developer projects.

4.1.1 Adaptive Software Development

ASD is an iterative and incremental software development methodology that prioritises flexibility and responsiveness to change. Introduced by Jim Highsmith and Sam Bayer [HI00], ASD is based on the idea that software development is inherently unpredictable, and requirements are likely to evolve over time. Unlike traditional approaches that emphasise detailed upfront planning, ASD acknowledges the challenges of planning in fast-changing environments and embraces change. The ASD life cycle is characterised by a speculate-collaborate-learn model (see Figure 4.1), where planning is replaced by continuous adaptation. The three phases of the life cycle are non-linear, overlapping, and dynamic. The key principles of each phase are as follows:

- **Speculate.** In the speculate phase, the traditional planning is replaced by defining the project mission. This involves establishing the goals and requirements of the project.
- **Collaborate.** A collaborative environment is a highly recommendable factor for emergent behaviour. However, in this project collaboration is not a viable step because only a single developer is involved in the project.
- **Learn.** The learn phase emphasises acknowledging and responding to mistakes. All stakeholders, including developers and customers, review their assumptions about the project. This phase may result in changes to the requirements as lessons are learned throughout the development process.

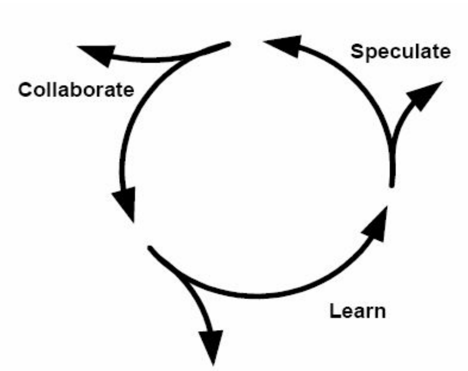


Figure 4.1: ASD life cycle. Image obtained from [Awa05].

4.2 Work distribution

In this section, we outline the distribution of work within the project. Initially, the project objectives were transformed into various work packages. It is worth noting that a fixed plan was intentionally omitted to align with the chosen development methodology. Throughout the project execution, the work packages were adjusted during the speculative phase of ASD as detailed in Section 4.1.1. The following description provides details on the final work packages, including the corresponding time intervals of their execution.

1. **Study of aggregation operators.** To align the aggregation process to the overall strategy of the DSS, it was necessary to study the available aggregation operators in order to select the most adequate alternative. This work package was executed from 19/07/2023 to 12/08/2023
2. **Implementation of aggregation operators.** During this work package, several of the aggregation operators that had been discovered were tested to validate their correctness and efficiency. While the first alternatives that were considered had scalability issues, which led to the exploration of alternative options. This work package was executed from 05/08/2023 to 16/08/2023
3. **Investigation of strategies for compensation movement detection.** Detecting compensation strategies solely through a video feed is a non-trivial challenge. This work package involved an in-depth exploration of compensation strategy types, and the automated approaches for their detection. This work package was executed from 13/08/2023 to 07/09/2023
4. **Implementation of compensation detection.** Once various approaches for compensation detection had been researched, a module for the detection of compensation movements was developed. This work package was executed from 28/08/2023 to 14/09/2023
5. **Implementation of compensation aggregation.** Using the insights gained during the study of aggregation operators, a new aggregation operator was implemented for aggregating the compensation level at different time intervals. This work package was executed from 12/09/2023 to 27/09/2023
6. **Integration of compensation in the DSS.** After implementing the compensation detection module, it was necessary to integrate its result into the logic of the DSS. This involved defining a new fuzzy variable, and expanding the set of fuzzy rules. This work package was executed from 28/09/2023 to 11/10/2023
7. **Validation of the suggested routines.** Over a significant phase of the project, there was constant validation of the routines recommended by the DSS, not only to validate that the new functionality worked properly, but also to ensure that it did not break the previous functionality. Since there is no simple way to automate the validation, it had to be done manually. This work package was executed from 15/08/2023 to 23/10/2023
8. **Study of strategies for making explanations more concise.** Multiple strategies for making explanations more concise were evaluated before arriving to the approach that was ultimately implemented. This work package was executed from 17/09/2023 to 07/10/2023

9. **Implementation of the improved explanation generation.** During this work package, not only was the XAI module enhanced to provide more concise explanations, but the system was modified to adapt to these changes. This included changes in the interface. This work package was executed from 12/10/2023 to 21/10/2023

Chapter 5

Architecture

This chapter outlines the system's architecture, which consists of multiple interrelated modules. The modules been organised into a multi-tier architecture. Firstly, a general description of the architecture is provided. Then, a formalisation of the the architecture of the system is presented. Finally, each one of the new modules added in this master's thesis, are described in detail, along with a technical justification of the decisions taken during their development. The decisions were made following an analysis of the issues that appeared, carefully considering the advantages and disadvantages of potential solutions.

5.1 General Overview

The purpose of the system is to enable home rehabilitation without the need of any specialised equipment, only requiring a standard webcam connected to a laptop for its use. Its primary focus is on physical rehabilitation for stroke patients. The system is a commercial system that is currently being developed. It is a web-based system with a client-server architecture. Specifically, there are two distinct client applications. One of those client applications is targeted towards therapists, while the other is for patients. The purpose of the patient application is primarily to guide the patient's movement when they perform the rehabilitation exercises autonomously. This is achieved by displaying the waypoints that patients need to reach with their joints using AR techniques. The purpose of the therapist application is to manage patients, assign them rehabilitation routines and review their progress. In order to reduce the workload of therapists, and increase the quality time they spend with patients, a DSS module is included in the designed architecture. This module is in charge of using the tracked progress of patients to automatically adjust their rehabilitation routine. While this process is automatic, therapists need to validate the adjusted routine before it is implemented. The author of the project has participated in the development of the system, specialising in the creation of both the backend and the client interface designed for therapists.

The architecture of the system is a layered architecture [Sch96]. It consists on dividing the architecture into multiple tiers or layers. This architectural approach was selected for its low coupling and high cohesion. The architecture is comprised of three layers: the presentation layer, the business layer, and the persistence layer. Since the application is divided into two clients and a server, layers are distributed between these three components. Each one of the clients features a presentation layer and a business layer, while the server features a business layer and a persistence layer. A graphical description of the system architecture can be seen in Figure 5.1. In the following subsections, each layer will be described in greater detail. To provide a comprehensive overview of the architecture, with a specific emphasis on elements relevant to this project, certain modules, such as the authentication module, have been omitted from the diagram. In the therapist application, modules were defined based on their relevance to the the DSS, with the Routines GUI Module, and the Adjust Routine Screen being the main focus. For the patient application, modules were selected to offer overall context for the compensation detection module in the server, and to present the data flow of the application in a general manner. In the server, the module definitions were structured to provide a more detailed description of the automatic routine adjustment process, with particular emphasis on the new developments introduced in this project. The connections in the business layer were defined to show the flow of information from raw data to adjusted routines with explanations.

5.1.1 Persistence layer

The persistence layer is responsible for managing the storage and retrieval of data generated during the execution of the system. In the context of this system, the persistence layer is in charge of managing the data regarding patients, exercises, routines, the data generated during the execution of the routine by patients, and the routines that are automatically suggested. This layer is only present in the server. During this project, the work on this layer consisted in adapting the database schema to the new needs of the system. This included adding new columns, and modifying the data stored in *jsonb* columns. *jsonb* is the binary equivalent of the JavaScript Object Notation (JSON) data structure, which enables to store unstructured data in relational databases. It is worth noting that, while the changes made to the database schema were not complex, they required detailed knowledge over the internal design of the database to ensure that changes are scalable, and consistent with the rest of the schema.

In order to store data in a non-volatile way, a Database Management System (DBMS) was used. The chosen DBMS was PostgreSQL[1]. PostgreSQL was chosen because it is an open source relational DBMS, and a relational schema enforces data integrity. In addition, it is an established DBMS with a large user base, and extensive documentation. In addition to a DBMS, an Object Relational Mapper (ORM) was used. An ORM is a programming technique that enables retrieving and manipulating data from a relational database using an object-oriented paradigm [KS09]. It is a desirable tool because it abstracts many of the details that are involved in the interaction with a DBMS, including SQL injection, managing connections with the database and parsing responses to queries. The ORM used in the project is

[1] https://www.postgresql.org/

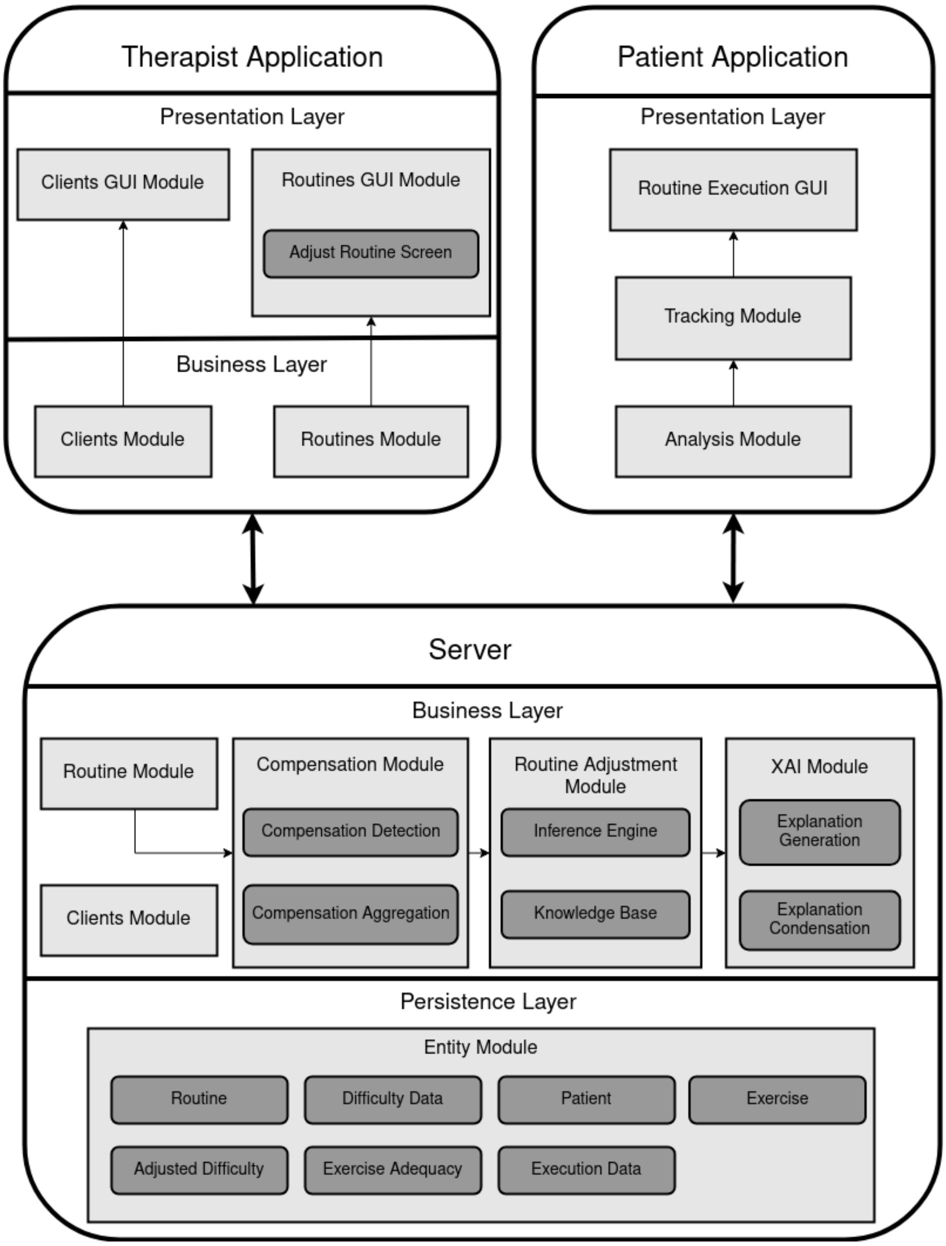


Figure 5.1: General overview of the architecture of the system.

TypeORM[2]. The main reason why TypeORM was chosen is because it is natively compatible with Typescript, providing seamless integration between the programming language and the relational database. This feature is not present in many ORMs in the JavaScript ecosystem. Additionally, TypeORM includes many useful features, such as migration support and a query builder.

The following are the most relevant entities for the project. They are described graphically using an Entity Relationship (ER) diagram in 5.2.

- **Patient**. This entity represents patients in the system.
- **Routine**. This entity represents the routines of patients, including its start and end

[2] `https://typeorm.io/`

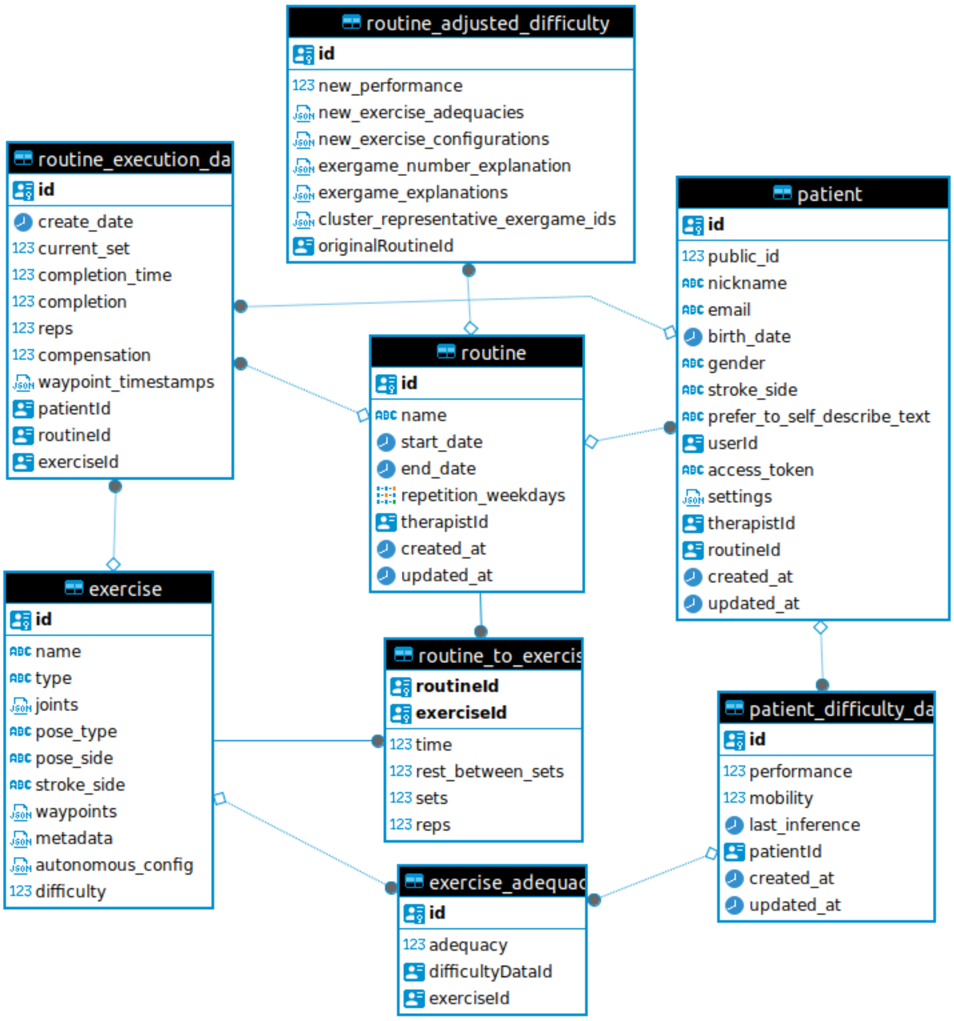


Figure 5.2: Entity-Relationship Diagram of the relevant entities in the system.

dates.

- **Exercise**. In the database, the information stored about exercises is the waypoints that describe how the exercise should be performed, and the difficulty of the exercise.
- **Routine to Exercise**. This entity represents an exercise in a routine, and includes information such as the scheduled number of sets and repetitions.
- **Routine Execution Data**. Each instance stores the execution data of a single set. It includes the estimated compensation level.
- **Patient Difficulty Data**. This entity represents data related to the patient that is considered when adjusting the routine.
- **Exercise Adequacy**. This entity represents how adequate is any given exercise for each patient.
- **Routine Adjusted Difficulty**. This entity contains the result of the automatic adjustment of the routine by the DSS. It is stored the database until the therapist validates

the suggested routine.

5.1.2 Business layer

In general, the business layer is responsible for managing the business rules and logic of an application. In the particular context of this system, the business layer is in charge of the logic for managing how relevant entities for the system should be updated, the rehabilitation application, and the automatic updating of routines. This layer is distributed between the server and the therapist application. Most of the work during this project has been focused in this layer. This includes compensation detection, enhancements to the XAI algorithm, and the new aggregation strategies introduced in this master's thesis.

Typescript[3] was the main language used in both clients and in the server. Typescript is a superset of Javascript that includes types that are checked at transpilation time. Typescript was chosen because its static type checking enables the detection of potential errors early in the development process. Additionally, it is a very versatile language that can be used both in the frontend and in the backend. To execute Typescript code in the backend, a Javascript runtime is required, and Node.js[4] was used. Node.js is a JavaScript runtime built on the V8 JavaScript engine that allows the execution of JavaScript and Typescript code on the server side. It was chosen because, while there are other alternatives such as Deno[5], they are not as established and do not support the npm (Node Package Manager) ecosystem and its many libraries. In addition, NestJS[6] was used as a framework for building the server. It was chosen because of its maturity, modularity and scalability. In order to implement the DSS, the R[7] programming language was used. R is a programming language and software environment for statistical computing. It was chosen for its ample support for fuzzy logic and other AI techniques. The sets[8] library was chosen for the implementing the FIS, and the fpc[9] library was used for executing clustering algorithms. In addition, RStudio[10] was used as an auxiliary Integrated Development Environment (IDE) for executing and visualizing the execution results of the R code.

The purpose of the DSS integrated in the system is to automatically adjust the routine of patients. This would enable therapists to either spend more quality time with their patients or to attend to a larger number of patients. The DSS simultaneously determines which is the adequate number of exercises for a patient, identifies which exercises should be included in the rehabilitation routine, and and establishes the parameters for each exercise, including the scheduled number of repetitions, sets, and time. To implement the DSS, a Fuzzy Inference

[3] https://www.typescriptlang.org/
[4] https://nodejs.org/en
[5] https://deno.com/
[6] https://nestjs.com/
[7] https://www.r-project.org/
[8] https://cran.r-project.org/web/packages/sets/
[9] https://cran.r-project.org/web/packages/fpc/
[10] https://posit.co/download/rstudio-desktop/

System (FIS) was used [Zad65]. Fuzzy logic was chosen for three major reasons. Firstly, no labelled data is required to implement a FIS. Several AI techniques, including ML and particularly Artificial Neural Network (ANN), require labelled data during the training phase, which is not available in this project. Secondly, fuzzy logic is able to manage uncertainty. It is common in the medical field that not all relevant variables are readily available or easily measurable. This is particularly important in this system, since it is based on computer vision, and measurements are not as precise as specialised sensors such as gyroscopes would be. Finally, fuzzy logic is intrinsically explainable. This is a major advantage when trying to integrate XAI in the system. It is worth mentioning that the fuzzy method used in the system is the Mamdani method [MA75].

The FIS is composed of fuzzy rules that dictate the behaviour of the FIS. Each rule has an antecedent and a consequent. Consequents are fuzzy variables, and antecedents may either be fuzzy variables, or expressions composed with t-norms and t-conorms. Fuzzy variables themselves are defined by a set called the universe of discourse and multiple fuzzy sets. Every fuzzy set is characterised by a membership function that describes the degree of membership, ranging from 0 to 1, for each value in the universe of discourse [Zad65]. In this system specifically, each fuzzy set is either *VL* for Very Low, *L* for Low, *M* for Medium, *H* for High, and *VH* for Very High. A graphical description of a sample fuzzy variable can be seen in Figure 5.3. Some examples of fuzzy rules are as follows:

- mobility is H && compensation is H → rep_incr is VL
- reps is VH && sets is M → set_incr is VH
- mobility is M && time is VH && completion is VH → adequacy_incr is M

The way in which fuzzy inference is used for generating the personalised routine is described in further detail in Algorithm 1. The DSS also has a XAI module that generates explanations for the decisions that it makes. Its algorithm will be explored in detail in Subsection 5.3.3.

5.1.3 Presentation layer

The purpose of the presentation layer is to present information to the user while also collecting user input and transmitting it to the business layer. In the system, the presentation layer is present in the therapist application as well as the patient application. During this project, the work in this layer has been focused on adapting the *Adjust Routine* screen in accordance to the new changes to the business layer. In particular, the changes to the XAI module make explanations clearer, but corresponding adjustments in how this information is presented to the user are also required. Originally, when a routine was adjusted automatically, each exercise was presented as an expansion panel. When collapsed, each expansion panel showed the updated configuration for that exercise, including the number of repetitions, sets, and time. However, when an expansion panel was expanded, the explanation for that exercise

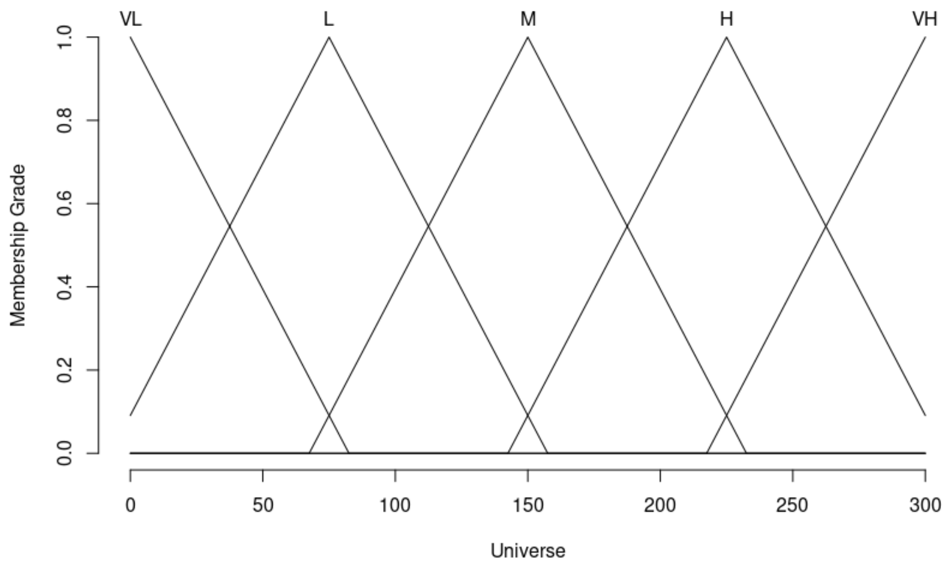


Figure 5.3: Sample fuzzy variable defined by a fuzzy partition.

was revealed. If the exercise was not present in the previous routine, then default parameters are selected, and no explanation is present. If that is not the case, then each explanation is presented as a row. Explanations are rules that were triggered during the inference of the FIS. Since if-then structures are intrinsically interpretable, they are valid explanations. To facilitate readability, explanations are presented visually to the therapist. Since it is an if-then structure, an arrow is used to separate antecedents and consequents. As they follow an if-then structure, an arrow is employed to separate antecedents and consequents. In both sides, variables are represented as sliders, where the minimum and maximum values depend on the variable being represented.

With the new changes addressed in this master's thesis, each explanation is not assigned to a single exercise. Instead, exercises are grouped into clusters, and each cluster is assigned a single set of explanations. Clustering allows us to group similar exercises together and reduce the number of explanations (see Subsection 5.3.3 for more details). It is for this change that the *Adjust Routine* screen had to be redesigned. Now, the configuration of exercises for the new routine is decoupled from the explanations. The first part of the page includes a table with the new configuration, where each row represents an exercise and each column a parameter of the exercise. Then, below the routine configuration, each cluster of exercises is presented as an expansion panel, similarly to how individual exercises were presented previously. Additionally, rules that share the same antecedents are combined into a single rule with multiple consequents. This approach enhances readability and reduces the total number of explanations to be read by the therapist.

In order to implement each of the web clients, a framework was used to facilitate the

Algorithm 1 Generation of the personalised routine.

Input exercises history, patient data, exergames parameters
Output routine, final performance

$performance\ accumulator \leftarrow$ empty list
$exergame\ number \leftarrow$ fuzzy inference of exergame number(patient data)
for each exercise history in exercises history, exergame parameters in exergames parameters **do**
    $exergame\ increments \leftarrow$ fuzzy inference of exergame data(exercise history)
    $exergame\ parameters \leftarrow exergame\ parameters + exergame\ increments$ ▷ The increments are applied to the parameters of the exercise
    $performance\ increment \leftarrow$ fuzzy inference of performance increment(exercise history)
    $performance\ accumulator \leftarrow$ append $performance\ increment$ to $performance\ accumulator$
end for
$sorted\ exergames \leftarrow$ Sort exergames according to their new adequacy
$routine \leftarrow$ Pick the top $exergame\ number$ exergames from $sorted\ exergames$
$final\ performance \leftarrow$ aggregate $performance\ accumulator$

process. In the case of the therapist application, Angular[11] was used. There are multiple web frameworks in the Javascript ecosystem. The most popular ones besides Angular are React[12] and Vue.js[13], but there is a wide variety to choose from. Angular was chosen because it is a mature framework, it is well documented, and, since it is an opinionated framework, it contains many features needed for web development out of the box. In order to implement the patient application, however, traditional web frameworks were not adequate. This is because the patient application had specific needs: for a repetition to be detected by the application, patients need to hit a set of circles, called waypoints, placed along the expected trajectory of the movement. Phaser[14] was the chosen framework for the implementation of the patient application. Phaser is a Hypertext Markup Language (HTML) game framework, and it was chosen because it facilitated many of the implementation details, such as collision detection, animations, and camera access. Finally, in order to track the pose of patients as they perform the exercises, the Mediapipe[15] library was used. Mediapipe provides on-device lightweight ML models for multiple tasks. In particular, the pose landmark detection was used for tracking the live video stream from the webcam. It was chosen because it provides an easy interface to the ML models, and because the patient application needs to be executed in a variety of devices that generally do not have powerful hardware for machine learning tasks. Additionally, being developed by a company such as Google contributes to long-term stability and support.

[11] `https://angular.io/`
[12] `https://react.dev/`
[13] `https://vuejs.org/`
[14] `https://phaser.io/`
[15] `https://developers.google.com/mediapipe`

5.2 Formalisation

As stated above, the architecture is divided into three layers (see Figure 5.1), that are spread across the two client applications and the server. The *Clients GUI Module* (Therapist application, presentation layer) is responsible for providing therapists with an interface for performing Create, Read, Update and Delete (CRUD) operations over the patients. Similarly, the *Routines GUI Module* (Therapist application, presentation layer) is responsible for providing an interface for CRUD operations over rehabilitation routines. An important component of this module is the *Adjust Routine Screen*, where routines can be automatically adjusted by the DSS. The *Clients Module* and *Routines Module* (Therapist application-Server, business layer) are distributed between the therapist application and the server. They handle CRUD operations, and are responsible for tasks like input validation and error handling.

All modules in the patient application correspond to the presentation layer, and there are three of them. The *Routine Execution GUI* module provides an interface for patients to perform rehabilitation exercises, and guides the patients' movements to ensure exercises are performed correctly. To guide user movements, waypoints are used to indicate the path that their movements must follow. Each waypoint $w_i \in W$ is defined as a R^2 vector, which describes the coordinates of the waypoint on the screen. Each coordinate of the vector is in the range $U = [0, 1]$.

The *Tracking Module* is responsible for recognising patient's movements in real time to support the execution of rehabilitation exercises. It receives as input a live video feed, and for every frame, it generates a pose. Formally, the tracking module is a function T defined mathematically as $T : VF \rightarrow PF$, where $VF = \{(vf_1, \ldots, vf_k) | vf_i \in U^{n \times m \times 3}\}$ is is the set of all possible ordered sets of image frames, where each frame is a $n \times m$ 3-channel image, and PF is the set of all possible ordered sets of poses. Poses are defined as mappings from joints to tuples describing the coordinates of said joint in the pose, and its level of visibility: $p_i : J \rightarrow P$, with $P = \{\langle co_i, v_i \rangle \mid co_i \in U^3, v_i \in U\}$. Also, $J = \{j_1, \ldots, j_{33}\}$ is the set containing all the joints tracked by the system.

The *Analysis Module* is responsible for processing the data obtained through the tracking module and determining how the exercise is actually being performed. Formally speaking, it is defined as a function $A : EPF \rightarrow ED$, where $EPF = \{\langle e_i, pf_i \rangle \mid e_i \in E, pf_i \in PF\}$ is the set containing all of exercise-pose feed pairs. $ED = \{\langle s_i, r_i, t_i \rangle \mid s_i, r_i \in \mathbb{N}, t_i \in \mathbb{R}\}$ is the set containing all possible *execution data*, which are 3-tuples describing the number of sets, the number of repetitions, and the time spent executing the exercise. E is the set of all exercises, defined as $e_i = \langle EJ(e_i), FJ(e_i), EW(e_i) \rangle$. $EJ : E \rightarrow J$ is a function that maps each exercise with the exercise joint being tracked during the exercise execution. $FJ : E \rightarrow \wp(J)$ is a function that maps each exercise with the set of fixed joints that cannot be moved during the exercise execution. For example, in Figure 6.1, the exercise joint is the wrist, and the fixed joint is the elbow. Finally, $EW : E \rightarrow \{(w_1, \ldots, w_n) \mid w_i \in W\}$ maps each exercise with the the ordered set of exercise waypoints that mark the trajectory to be followed by the exercise joint.

The *Compensation Module* is in charge of assessing the compensation level from the tracking data. It can be formulated as $C = \langle CD, CA \rangle$, where CD and CA represent the two steps performed by the compensation module. $CD : PF \to UC$ is the compensation detection step, and it has been formalised as a function that maps a stream of poses to the un-aggregated compensation, where $UC = \{(\langle c_1, cs_1 \rangle, \ldots, \langle c_n, cs_n \rangle) \mid c_i \in [0, 100], cs_i \in U\}$ is the set of all possible ordered sets of $\langle c_i, cs_i \rangle$ tuples. c_i represents the compensation level in time interval i, while cs represents the confidence score in that same time interval. $CA : UC \to [0, 100]$ is the aggregation operator for aggregating multiple compensation values into a single one. See Subsection 5.3.2 for more details.

The *Routine Adjustment Module* is responsible for automatically adjusting the routine of patients and suggesting a new one. It is the core of the DSS, and it can be formalised as $RAM = \langle IE, KB \rangle$, where IE is the inference engine, and KB is the knowledge base. The knowledge base can be defined as $KB = \langle FR, FV \rangle$, where FR are the fuzzy rules and FV are the fuzzy variables. Fuzzy variables are defined as $FV = \{\langle UD_i, FS_i \rangle \mid i \in \{1, \ldots, n\}\}$, where UD_i is the universe of discourse of fuzzy variable i, and FS_i is the set containing all fuzzy sets for variable i. Each fuzzy set $fs_{ij} \in FS_i$ is defined as a variable that maps each value from the universe of discourse to a value from 0 to 1: $fs_{ij} : UD_i \to U$. Fuzzy rules are defined as $FR = \{\langle AC_i, cq_i \rangle \mid AC_i \subseteq FV, cq_i \in FV, i \in \{1, \ldots, n\}\}$, where AC_i are the antecedents for rule i, and cq_i is the consequent for rule i. Both antecedents and consequents are fuzzy variables. The inference engine has been formalised as a function $IE : IEI \to RAD$. The inference engine input can be defined as $IEI = \{\langle red_i, KB \rangle \mid red_i \in RED\}$, where KB is the knowledge base, as previously defined, and RED is the set containing all possible routine execution data. The routine execution data itself can be defined as a mapping from exercises to a tuple containing the execution data and the compensation level: $red_i : E \to \{\langle ed_i, c_i \rangle \mid ed_i \in ED, c_i \in [0, 100]\}$. On the other hand, the routine adjustment data is defined as $RAD = \{\langle np_i, nea_i, rc_i \rangle \mid np_i \in [0, 100], nea_i \in NEA, rc_i \in RC\}$. np_i represents the new performance of the patient, $nea_i : E \to [0, 100]$ represents the new adequacy levels for all the exercises, and $rc_i : E \to \{\langle nr_i, ns_i, nt_i \rangle \mid nr_i, ns_i, nt_i \in \mathbb{R}\}$ represents the new routine configuration, which specifies the new number of scheduled sets, repetitions, and time for each exercise.

The XAI *Module* generates explanations for the decisions taken by the *Routine Adjustment Module*. Explanations are in the form of rules that were activated by the inference engine. The XAI *Module* can be formalised as $XAI = \langle EG, EC \rangle$. $EG : IEI \to \{\langle RE_i, ee_i \rangle \mid RE_i \subseteq EX, ee_i \in EE\}$ represents the explanation generation step of the module, where RE_i are the explanations specific to the routine, and $ee_i : E \to \wp(EX)$ are the explanations specific to each exercise in the routine. $EX = \{\langle AC_i, CQ_i \rangle \mid ACE_i, CQE_i \subseteq EXV\}$ is the set of all possible explanations, which are similar to rules except for the fact that explanations may have more than one antecedent. In addition, antecedents and consequents of explanations are not comprised only by a fuzzy variable, but instead by a 3-tuple containing a fuzzy variable, a value for the variable, and the fuzzy set within that variable: $EXV = \{\langle fv_i, v_i, fs_i \rangle \mid fv_i \in FV, v_i \in UD_i, fs_i \in FS_i\}$. $EC : EE \to \{\langle cl_i, eec_i \rangle \mid cl_i \in CL, eec_i \in EEC\}$ represents the explanation condensation step of the XAI module, where explanations specific to exercises are grouped to make the explanations more concise. $cl_i : E \to \mathbb{N}$ represents the grouping of exercises into clusters, while $eec_i : \mathbb{N} \to \wp(EX)$ maps each cluster to its explanations. See Subsection 5.3.3 for more details.

Finally, the *Entity Module* is the only module in the persistence layer, and it is located in the server. It is responsible for managing the non-volatile storage and retrieval of system data. Its components are the various entities that represent relevant information for the system Each entity is described in further detail in Subsection 5.1.1.

5.3 Evolution of Architecture

In this section, the major developments executed during this project are presented. Firstly, the inclusion of new aggregation strategies for both inputs and outputs of the FIS will be discussed. Secondly, the compensation detection module added in this master's thesis will be described, as well as its integration with the DSS. Finally, the enhancements to the XAI module will be analysed. For each one of the developments, the decisions that were taken are discussed in detail, along with the advantages and disadvantages of each alternative.

5.3.1 New aggregation strategies

As discussed in Section 2.1, aggregation operators constitute a fundamental aspect in the realms of AI and decision-making. In the case of the DSS of this system, aggregation is necessary both for input values, and for output values. Regarding input values, aggregation is required as there is data spanning multiple days for each exercise. This need is particularly evident for compensation data, since there are data points at intervals of a few seconds.

For input values, the preferred aggregation method for most variables is the arithmetic mean. However, two variables, compensation, and performance increment, have data distributions such that the arithmetic mean may not be the most suitable operator. For example, if most compensation values during an exercise execution were low and only a few of them were high, then it would be preferable to give more weight to the high compensation values rather than to treat all values equally. Similarly, exercises with a low performance increment should be given higher weight in the aggregation process. For the compensation, the values to be aggregated represent groups of frames, and not every group of frames needs to include a compensatory movement for the aggregated value to be considered as compensation. In the case of the performance increment, a conservative approach is preferable since it involves aggregation across multiple exercises. Unlike inputs, which are exercise-specific, the performance increment describes the status of the patient on a more global level. Consequently, an increase in performance level is justified only if the overall trend suggests improved performance. Building on the previous example, assigning higher weight to exercises with lower performances results in a slower increase in the difficulty of the rehabilitation routine. This approach enhances the long-term sustainability of the rehabilitation process.

It is worth noting that there are constraints associated with both cases. The number of elements to be aggregated is undetermined, with the time taken by patients for exercise performance unknown in the case of compensation. Similarly, for performance increments, patients may be assigned varying numbers of exercises. Additionally, the aggregation operator should preferably be explainable, since the XAI module operates on the assumption that the underlying logic of the DSS is inherently explainable.

There are various aggregation operators provide a more nuanced approach than the arithmetic mean. The Choquet Integral and the Sugeno Integral are notable of aggregation operators. These integrals offer greater expressiveness, serving as a generalized version of many other aggregation operators. However, their main drawback lies in their reduced understandability compared to OWA operators. As it was discussed in Section 2.1, the persistent use of OWA operators over time is due to their inherent explainability. Despite the enhanced expressiveness of fuzzy integrals, the clarity and interpretability of OWA operators have led to their continued preference.

Many variations of the OWA operator were also considered for the aggregation of the performance increment and the compensation. The WOWA operator is not applicable to either case. The WOWA operator assigns weights to both ordered positions and input values, and in the context of this project, input values are irrelevant. Exercises are indistinguishable when aggregating the performance increment, and the same principle applies to groups of frames and the compensation level. The same line of reasoning extends to the weighted average.

TOWA operators do not have any particular disadvantage for these aggregation scenarios, but they do not provide any particular advantage either. As shown in [Yag05], given a weight vector W, TOWA operators are more restrictive than the equivalent OWA operators. This is because the minimum is the largest t-norm, and replacing the minimum with other t-norm results in lower aggregated values. However, since similar results can be achieved by changing the weight vector of a regular OWA operator, TOWA operators were discarded.

Another promising variation of OWA operators are type-1 OWA operators. Type-1 OWA operators aggregate fuzzy sets rather than crisp values. This approach is particularly relevant for the aggregation of performance increments, since they are the result of a fuzzy inference. Therefore, it would be possible to perform the aggregation before the defuzzification process, consequently reducing the information loss that takes place when defuzzifying. Three main approaches were considered for type-1 OWA operators. The first approach, proposed in [ZCJG08], is the one that can be considered to be the most *pure* among the three. This is because it is able to aggregate any arbitrary fuzzy set so long as it has been discretized. It is also worth noting that in this approach weights are also fuzzified. However, the main drawback of this approach is its low scalability. Since type-1 OWA operators has been defined according to Zadeh's extension principle, its calculation involves calculating the supremum of $\mu_{W_1}(w_1) * \cdots * \mu_{W_n}(w_n) * \mu_{A_1}(a_1) * \cdots * \mu_{A_n}(a_n)$ for every variation of weights, and elements from the domain of discourse. Its runtime complexity is expressed as $P(m, n) = \frac{m!}{(m-n)!}$, where m represents the total number of distinct elements available for selection, and n corresponds to the number of elements to be selected, matching the number of input elements to be aggregated. The value of m is determined during the discretization of the universe of discourse set, in the case of input elements, and the $[0, 1]$ set in the case of weights. Given that the number of elements after discretization tends to be considerably larger than the number of input elements, the number of calculations has a factorial growth. Several strategies were explored to reduce this complexity, including evaluating a sample of the total variations. However, these attempts were not effective because the aggregated value tends to 0 if the number of variations is not high enough. Additionally, the fact that weights are also fuzzified increases the complexity of defining the weights, and it drastically reduces the explainability of the DSS. Two additional approaches were considered for type-1 OWA operators. Firstly, the approach proposed in [ZCJG11] was considered, since it gives equivalent results, and it is allegedly faster than the previous one. However, it can only be applied to fuzzy numbers, which are a a subset of all possible fuzzy sets, and performance increments may not necessarily be a fuzzy number because it is the output of a fuzzy inference involving multiple fuzzy rules. Secondly, the approach proposed in [MS00] is also faster and, while it does still require inputs to be fuzzy numbers, it does not fuzzify the weights. Ultimately, all three approaches suffer from poor scalability and require substantial modifications to be applied in this project.

In addition to the consideration of various OWA variations, it is important to choose an appropriate strategy for selecting weights, or a parameterized OWA family altogether. Some OWA families provide a valuable semantic for the chosen weights, which is a crucial aspect given the explainability aspect of the DSS. S-OWA, for instance, are easy to compute and provide a significant semantic value. Depending on whether it is an *orlike* or an *andlike* S-OWA, it provides a valuable distinction by giving more relevance to either the first or last value. The Maximum Entropy OWA provides an interesting semantic, that being of the family with the highest entropy (or equivalently the lowest dispersion) for any given level of orness. However, its computation is more complex compared to S-OWA. Finally, it is also possible to derive the weights from labelled data, but this approach is not viable in this project because there is no labelled data available.

Following a comprehensive evaluation of multiple alternatives, the chosen approaches for aggregation are as follows. For aggregating performance increments, a S-OWA was selected. Since performance increments are not guaranteed to be a fuzzy number before fuzzification, none of the type-1 OWA operators were applicable. The semantics provided by S-OWA contribute to the interpretability of the FIS. Additionally, the semantics facilitates its usage by developers of the DSS. In the case of compensation, an approach inspired by one of the type-1 OWA operators was adopted. Although compensation values themselves are crisp numbers, the uncertainty introduced by the tracking module allowed for the generation of confidence scores for each compensation value. Using these confidence scores, the compensation level was transformed into a fuzzy number. Furthermore, it was necessary to manage the excessive time complexity of type-1 OWA operators (see Figure 5.4). This methodology is elaborated further in Subsection 5.3.2.

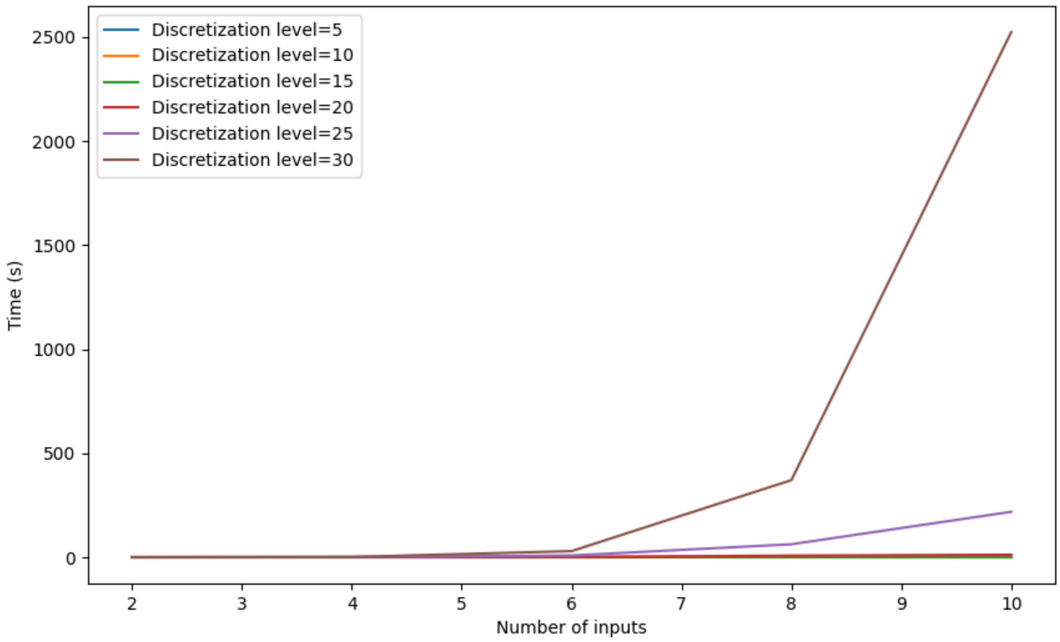


Figure 5.4: Execution time of type-1 OWA operator.

5.3.2 Compensation movements detection

Various compensation strategies are employed by patients performing UL rehabilitation, as discussed in Section 2.2. Pathological synergies between muscles, including gross extensor and flexor synergies, are notable examples. Another compensation strategy involves compensatory trunk displacement to overcome limitations in UL mobility. Finally, fixation of body segments, such as the pelvis, lumbar spine, or thorax, constitutes another compensation strategy. This approach reduces the number of motor elements to be controlled to achieve a motor task.

To detect compensatory movements, it is crucial to consider the input interface used by patients to perform exercises. As seen in Section 2.2, many approaches rely on dedicated sensors with high-fidelity tracking capabilities. However, given that the system is designed for home rehabilitation with an emphasis on ease of use and cost-effectiveness, the only available sensor is a regular webcam. It is necessary to decide the types of compensatory movements that the system will detect, considering its limitations. Tracking fixation of body segments or pathological synergies between UL muscles is challenging due to the detailed tracking requirements, which exceed the capabilities of a standard webcam. For instance, detecting pathological muscle synergies requires tracking the rotation of the upper arm, wrist, and finger movements, as illustrated in Figure 2.2. On the other hand, excessive trunk displacement is a compensatory movement that can be tracked relatively easily using ML models designed for computer vision. Consequently, the decision was made to focus on detecting compensatory movements related to excessive trunk displacement.

To detect compensatory movements, captured poses are accumulated as the exercise is executed, and these are processed as the sets are finished. Online processing is not necessary since the compensation level is used solely to support the DSS. If compensation detection were employed for providing real-time feedback to the patient, then real-time processing would be essential. Since this project uses Mediapipe for tracking patient movements, the compensation detection must rely on the information provided by the library. For each pose tracked from the live video feed, a list of predetermined joints is tracked (see Figure 5.5). The tracked information for each joint includes the x, y, and z coordinates, as well as the estimated level of visibility. The x, y coordinates correspond to the image plane and are considered to be fairly reliable, while the z coordinate represents depth, with the midpoint of the hips serving as the origin. It is worth noting that the values of the z coordinate are not as reliable as the x or y coordinates because the z axis is perpendicular to the image plane. The joints involved in tracking trunk displacement include the left and right shoulders, as well as the left and right hips.

In the compensation detection process, the first step consists in filtering poses by requiring a minimum level of visibility. Next, the angle between the shoulder midpoint and the hips midpoint (ignoring the z-axis due to lack of reliability) is calculated for each pose. The angle is a relevant metric because trunk displacement is often non-uniform, and only the upper trunk is displaced while the lower trunk remains static or moves in the opposite direction. Additionally, the offset of the trunk in each axis between each pose and the previous one is computed. The trunk coordinates are defined as the centroid of the four joints. The visibility of a pose is defined as the minimum visibility value among the four joints. Poses are then grouped into intervals of 10 seconds. The interval length was chosen to be just large enough to allow time for a single repetition performed at a moderate speed. The confidence score for each group of poses is defined as the average visibility of each pose, with invalid

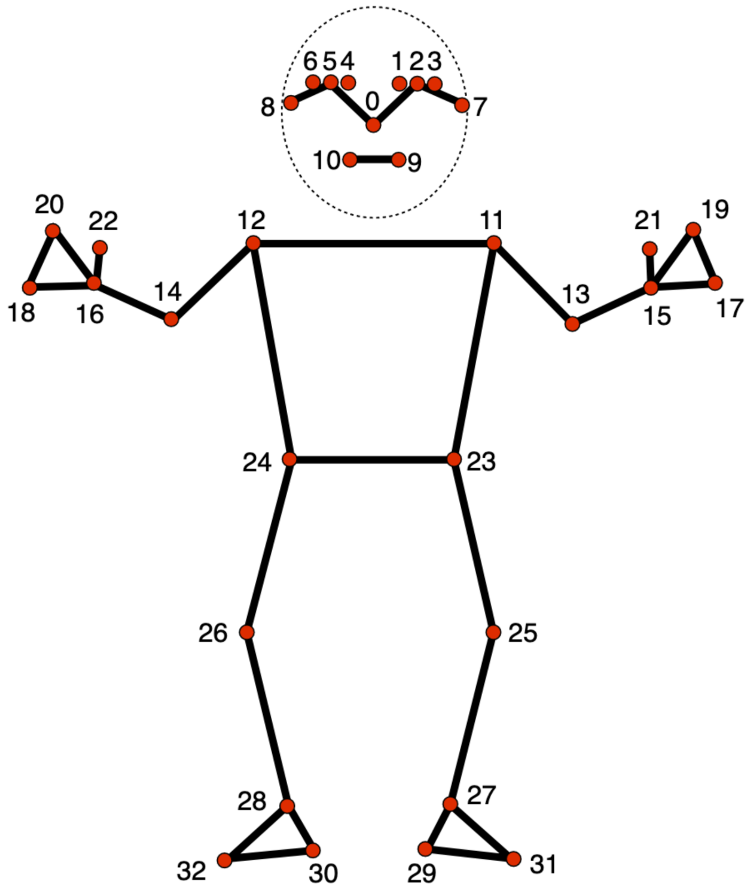


Figure 5.5: Joints tracked by Mediapipe. Image obtained `https://developers.google.com/mediapipe/solutions/vision/pose_landmarker`

poses providing a visibility score of 0. While this approach effectively detects compensatory movements, it is important to note that it relies on the Mediapipe library and does not consider other factors that may affect tracking accuracy. For instance, in unfavourable visibility conditions such as insufficient lighting or patients wearing low-contrast clothing, jittering in the tracking algorithm may occur. Ideally, this should also be considered when determining the confidence score.

To determine the compensation level of trunk displacement, the aggregated variability of each of the four metrics (angle of the shoulder-hips midpoints and displacement along three axes) is calculated. To calculate the displacement, the 5% highest and lowest values are discarded for each metric to avoid the distortion that may be caused by the imprecision of the tracking module. Then, the Mean Absolute Percentage Error (MAPE) over the average is employed to compute the mean deviation in relative terms. It is imperative to use a relative measure because the four metrics have different scales, and otherwise they could not be combined together. Additionally, it is necessary to aggregate absolute deviation to prevent positive and negative increases from cancelling each other out. Given these constraints, MAPE proves to be the most suitable aggregation method. For each time interval, the system generates four compensation values, each with its respective confidence score.

As detailed in Subsection 5.3.1, an adaptation of a type-1 OWA operator has been used for aggregating the compensation values. Specifically, the approach outlined in [MS00] was employed for fuzzy sorting. However, since the time complexity of the complete algorithm is excessive, it was simplified. Instead of using Zadeh's extension principle to extend OWA operators to fuzzy sets, only the sorting strategy is applied. The fuzzy sets, once sorted, are defuzzified, and a regular OWA operator is employed. In this case, an orlike S-OWA operator is used to give more importance to higher values of compensation. This strategy facilitates the incorporation of the confidence score in the aggregation process while maintaining a reasonable time complexity. The compensation values to be aggregated, along with their confidence scores, are transformed into fuzzy numbers by generating an isosceles fuzzy triangle. The membership value is set to 1 for the compensation value, and it decreases as the distance from the compensation value increases. The confidence score dictates how steep the decrement of the membership is as the distance increases. A higher confidence score results in a steeper decrement.

Fuzzy sets are sorted by using a permutation matrix, in which each value ranges from 0 to 1. Although not explicitly stated in [MS00], the multiplication of scalar values in the matrix with fuzzy numbers is carried out by multiplying the scalar value with the membership function. Similarly, the addition of fuzzy numbers is performed by adding their membership functions. Since membership functions inherently range from 0 to 1, it is essential to normalize the permutation matrix before sorting the numbers to ensure that the resulting fuzzy sets maintain this property.

The elements in the permutation matrix, denoted as $S_{i,j}$, can be obtained using the formula: $S_{i,j} = \max\left(1 - |r_j - t_i|, 0\right)$. Here, r_j represents the rank of fuzzy number j, and t_i is a variable used to untie two fuzzy sets when they overlap. The rank r_k is calculated as follows: $r_k = \sum\limits_{j=1}^{n} p'_{jk} + \frac{1}{2}$. Here, p'_{jk} is defined as $p'_{jk} = \frac{p_{jk}}{p_{jk}+p_{kj}}$, and $p_{jk} = \text{SIM}(a_j, X_{jk})$. a_i represent the fuzzy numbers to be aggregated, X_{jk} denotes the fuzzified version of the minimum operator, and SIM stands for a standard similarity measure for fuzzy numbers. The standard similarity measure used is defined in [WDK95], and the fuzzified version of the minimum is expressed as: $X_{jk}(w) = \max\left(\min(a_j(u), a_k(v))\right)$, where $w = \min(u, v)$.

The only variable that has not been defined is t_i, which is implicitly expressed as: $i = \frac{1}{2} + \sum_j c_j(t_i)$. Here, c_j is a hard-limited curve defined as:

$$c_j(x) = \begin{cases} 0 & \text{if } x < r_j - 1 \\ 1 & \text{if } x > r_j + 1 \\ \frac{x - r_j + 1}{2} & \text{otherwise} \end{cases}$$

t_i must be expressed in a explicit way, such that it can be computed. Since c_j is a piecewise function, we need to study $\sum_j c_j(t_i)$ over intervals delimited by the individual limits of each c_j function being summed. On each interval, each individual function may yield 0, 1, or a value in between depending on t_i. Let's define the number of individual functions that yield 0 as a, the number of functions that yield 1 as c, and the rest as b, with the constraint $a + b + c = n$ (where n is the total number of functions). Therefore, on each interval, we have:

$$
\begin{aligned}
i &= \frac{1}{2} + 0a + 1c + \sum_{j_b} c_j(t_i) \\
i &= \frac{1}{2} + c + \sum_{j_b} \frac{t_i - r_j + 1}{2} \\
i - c - \frac{1}{2} &= \frac{1}{2} \sum_{j_b} t_i - r_j + 1 \\
2i - 2c - 1 &= \sum_{j_b} t_i - r_j + 1 \\
2i - 2c - 1 &= bt_i + b - \sum_{j_b} r_j \\
bt_i &= 2i - 2c - 1 - b + \sum_{j_b} r_j \\
t_i &= \frac{2i - 2c - 1 + \sum_{j_b} r_j}{b} - 1
\end{aligned}
$$

For the answer to make sense, the parameter b must be non-zero. Additionally, the uppermost and lowermost intervals, where either all values are 1 or all values are 0, do not yield valid answers and can therefore be omitted. The evaluation of the preceding expression is required for each interval, and if the resulting value falls within the specified interval, then it is the solution

One we have discussed how the compensation level is calculated and aggregated, it is relevant to discuss how it impacts the DSS, and in particular, the FIS. The compensation level serves as an input for the FIS. When defining the fuzzy variable, a distinctive approach was employed compared to other variables. Rather than creating an evenly spaced fuzzy partition using triangles, it was defined using trapezoids concentrated on the lower values (see Figure 5.6). This choice was driven by the fact that compensation values close to 100% are infrequent. It must also be studied which output variables should be affected by the compensation variable. A high compensation value should negatively impact the number of repetitions, since one cause of compensation movements is patient overexertion. Moreover, it should indirectly influence the exergame number, as more exergames tend to result in more repetitions. Additionally, the compensation variable affects the exergame adequacy, because compensation strategies may arise when a particular exergame is not adequate for the patient.

Finally, it is important to follow a strategy when defining the fuzzy rules. Since our goal is to reduce the value of those output variables, only rules whose consequent is Very Low (VL), Low (L), or Medium (M) were added. This ensures that no rule based on compensation contributes to an increase in the number of repetitions.

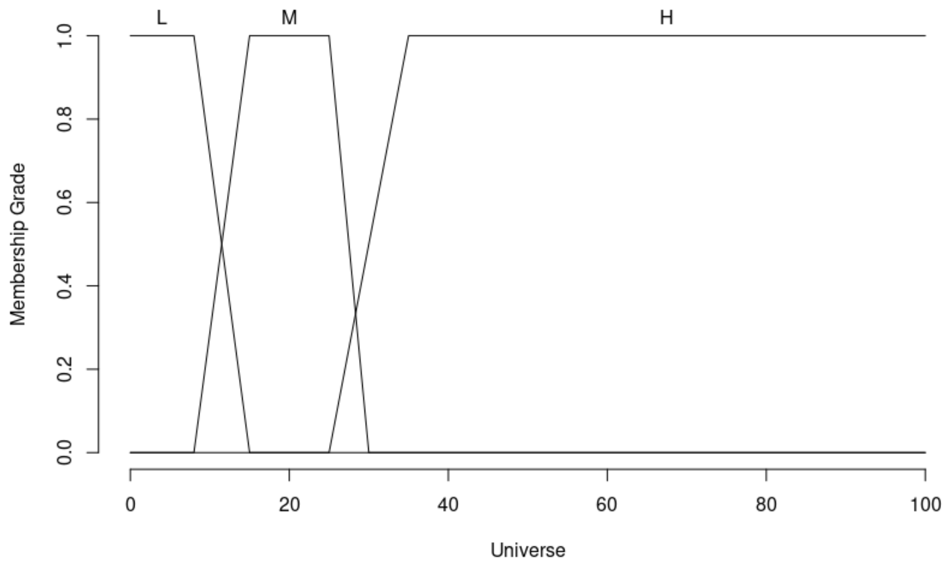


Figure 5.6: Fuzzy definition of the compensation variable defined using trapezoids.

5.3.3 XAI enhancements

The main purpose of the XAI module in this system is to provide explanations for the decisions made by the DSS during the automatic adjustment of patient routines. While the XAI module is able to interpretable explanations that actually reflect the inference process of the DSS by providing activated rules, it struggles with conciseness. Since an explanation is provided for every output variable and for each exercise, the total number of explanation for a given routine is excessive. Considering the number of explanations, need to spend significant time reviewing the suggested routine. Even when explanations are presented visually to therapists, the advantages of automatically adjusting the routine are significantly diminished. Therefore, there arises a necessity to simplify the explanations provided to therapists. The new strategies should not only reduce the number of explanations but also aim to convey a comparable level of information with fewer explanations. Ultimately, the objective is to enhance the information density of explanations, ensuring that important details are maintained while minimising the overall volume of information presented.

The first strategy that was implemented for increasing the density of explanation was grouping exercises that were similar and fusing their explanations. While patient may execute each exercise in different ways, explanations do not differ excessively among them. Therefore, many explanations may be redundant across exercises, allowing for their fusion without significant loss of information. To determine which exercises should be grouped together, clustering was employed. As the number of clusters was unknown in advance, clustering algorithms that require a predetermined number of clusters, such as k-means and fuzzy c-means, were discarded. The two main contenders were DBSCAN and hierarchical clustering. DBSCAN was ultimately chosen for its effectiveness in identifying clusters of varying sizes and densities. The minimum number of samples was set to 1, since 1-sample clusters are valid, and to prevent any sample from being labelled as noise. The radius was determined iteratively by testing different values and selecting the one yielding the most reasonable result. It is worth noting that, in this context, 1-sample clusters are valid, which is not typically the case in other clustering applications. This influenced the selection of the radius. Since that the minimum and maximum values allowed for each variable were available, min-max normalization was used instead of standardization. The selected clustering parameters were exercise-specific variables, both inputs and outputs, involved in the FIS This is because these are the only variables capable of influencing the explanations. Exercises newly added to a patient's routine were grouped by default. A visual representation of a clustering result after performing principal component analysis can be observed in Figure 5.7.

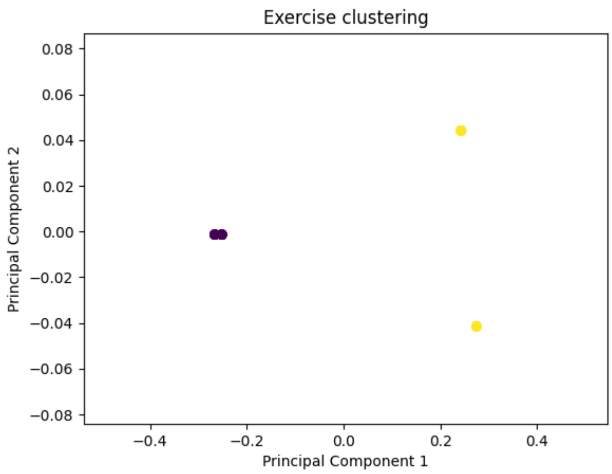


Figure 5.7: Exercise clustering (after performing principal component analysis).

The second strategy, implemented in conjunction with the first, focuses on further reducing the number of rules for each cluster of exercises by fusing explanations. Initially, modifying how explanations are generated was considered. The current approach involves selecting the activated rule with the highest aggregate membership value for each variable. The possibility of accepting multiple rules or none to enhance the relevance of the selected rules was evaluated but ultimately discarded. Ensuring at least one explanation for each variable is crucial to prevent scenarios where too few explanations are provided, and the interpretability of the DSS is reduced. Another approach was the fusion of explanations that shared some antecedents, but not all of them. However, this approach had the risk of introducing unintended meaning in explanations. For instance, an input variable could be the antecedent of an output variable in an explanation when, in reality, they could be unrelated. Consequently, only rules sharing all antecedents are actually fused. While limited, this approach reduces the number of explanations, and conveys the same information. The logic of the XAI module is summarised in Algorithm 2.

Algorithm 2 Generation of the explanation for the decisions of the DSS.

Input rules, output variables, exercises
Output explanations

$explanations \leftarrow empty\ list$
for each exercise in exercises, output variable in output variables **do**
  $max\ relevance \leftarrow -\infty$
  $relevant\ rule \leftarrow null$
  for each rule in rules **do**
    if rule does not affect output variable **then** continue
    end if
    $fuzzy\ set \leftarrow fuzzy\ inference(rule, exercise, output\ variable)$
    $relevance \leftarrow reduce(fuzzy\ set, +)$ ▷ Add membership values to obtain the relevance
    if $relevance > max\ relevance$ **then**
      $max\ relevance \leftarrow relevance$
      $relevant\ rule \leftarrow rule$
    end if
  end for
  $explanations \leftarrow$ append $relevant\ rule$ to $explanations$
end for
$clusters \leftarrow$ execute clustering over $exercises$
$explanations \leftarrow$ select a single representative for each $cluster$ in $clusters$
for each rule 1, rule 2 in explanations **do**
  if rule 1 and rule 2 belong to the same cluster and share all antecedents **then**
    Fuse rule 1 and rule 2
  end if
end for

Chapter 6

Results

Following the exploration of the project's technical aspects and the design decisions that were taken, the purpose of this chapter is to present the results of the development, obtained as a result of deploying a comprehensive remote rehabilitation system based on the proposed architecture. In this chapter, there will be an emphasis on the graphical appearance of the system. In particular, the results related to the DSS will be described in detail, since it was the original focus of the master's thesis. The context in which the project has been executed will be examined. Then, illustrative examples of the DSS results are provided. The purpose of these examples is to perform a preliminary validation to ensure that results are reasonable, as a preliminary step to clinical trials. Various statistical metrics related to its development are also included. Finally, a cost estimation of the project is presented.

6.1 Project Context

As described in Chapter 5, the system is a cloud-based system, and comprises three major components. There is a server, a client application for patients and a client application for therapists. The purpose of the patient client is to guide and evaluate the execution of rehabilitation exercises at home, as well as giving feedback (see Figure 6.1), while the purpose of the therapist application is to manage patients and routines. Therefore, the purpose of the server is to connect the two client applications together. As shown in Figure 6.2, the changes performed by the therapist are reflected in the patient's application. The source code of the project is organised into three repositories. The *therapist* repository contains both the server as well as the therapist application. The *webapp* repository contains the patient application. Finally, the *shared* repository is a git submodule, and it contains common code that is shared between the two repositories. The source code files modified during the project's execution is available at `https://pruebasaluuclm-my.sharepoint.com/:u:/g/personal/sergio_martinez34_alu_uclm_es/EYvQ-ik10LhLjPHjEgUAV6kBmY_dLgG-eDuff4X5q_ePaw?e=uBO6Nn`.

6.2 Final Result

The functionality of the DSS is accessed through the *My Routines* page (see Figure 6.3). In this page, all routines present in the system are shown. From this page, the therapist can access the functionality for editing the routines manually, deleting routines or automatically

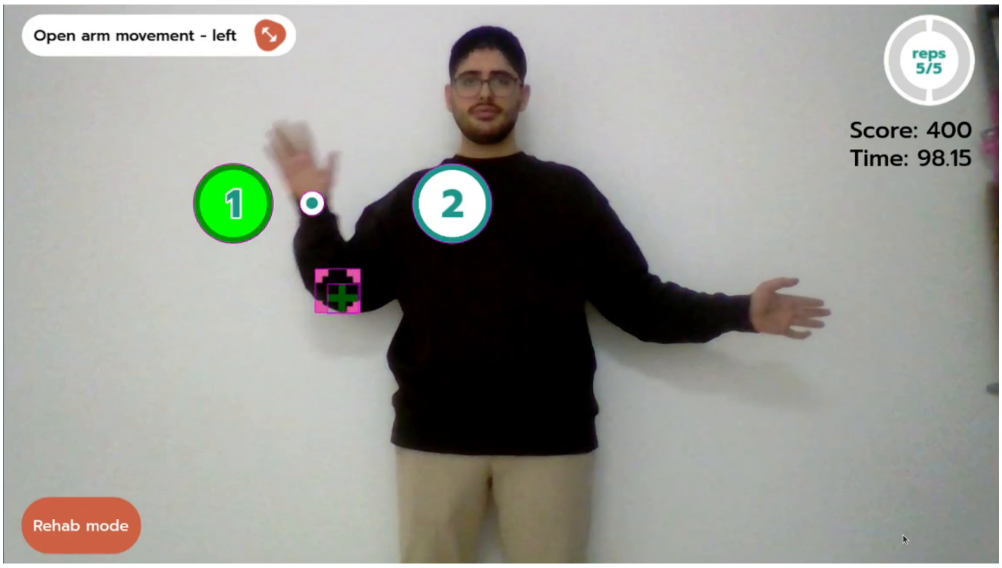


Figure 6.1: Patient application guiding the movements of the patient.

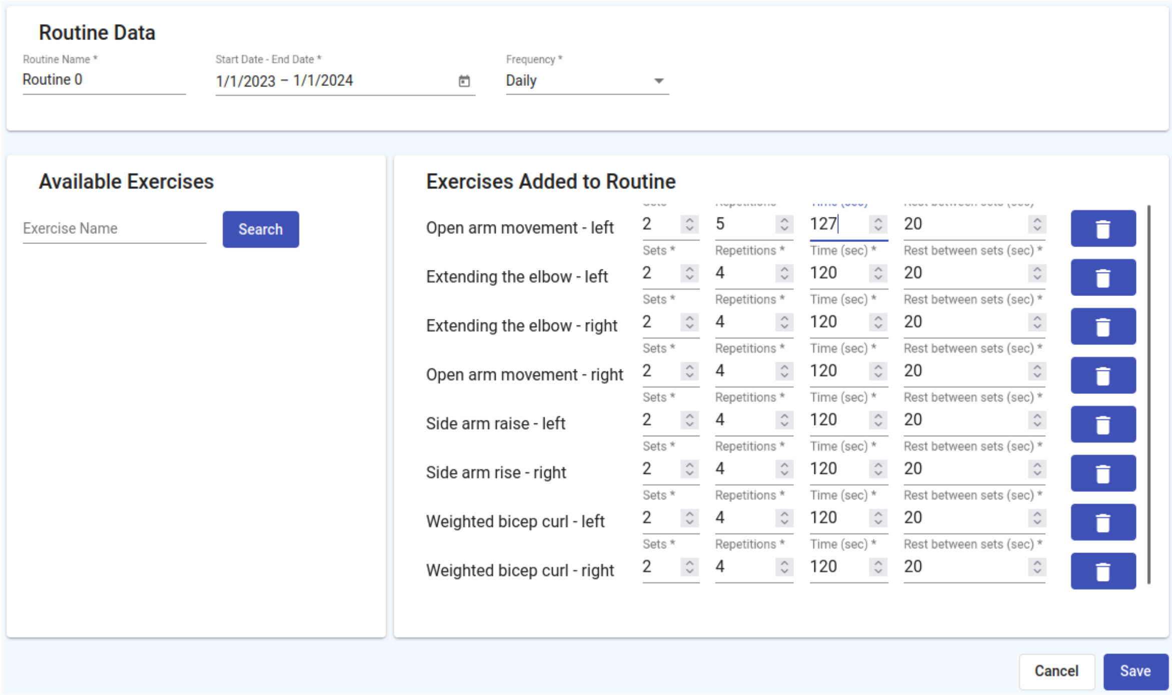
adjusting routines.

Initially, in the *Adjust Difficulty* page, the only element present is a button for requesting the automatic difficulty adjustment (see Figure 6.4). Once the adjustment has been performed, the suggested routine is presented to the therapist.

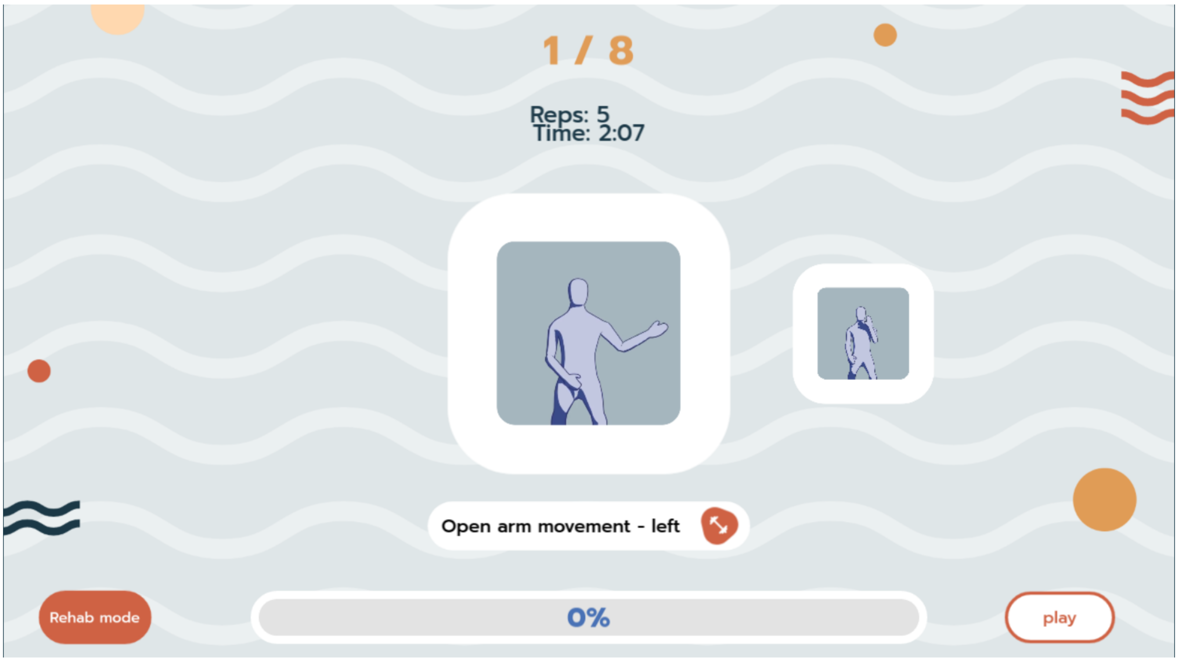
Originally, since each exercise was assigned a set of explanations, each exercise was listed as an expansion panel, with the configuration of each exercise presented within the expansion panel (see Figure 6.5). Currently, the new routine and the explanations are decoupled (see Figure 6.6). Firstly, the new routine configuration is presented in a table format, which is easier to read than the previous approach. Then, a list of explanations is shown. Each expansion panel corresponds to a group of exercises that share the same explanations (see Figure 6.7). Within each group, the exercises that compose the routine are listed, and then the explanations are shown. Explanations are presented in a visual way, where each variable is depicted as a slider. The antecedents precede an arrow, with the consequents following the arrow. This visual presentation enhances clarity and comprehension.

6.3 Inference Results

In this section, multiple examples of the inference results will be presented. Each example will showcase one of the project's enhancements, as discussed in Section 5.3. In all examples, the start date of the routine is the 03-01-2023, and the end date is the 09-01-2023. The routine is scheduled to be executed every Monday, Wednesday and Friday. Each routine consists of 8 exercises, with the daily goal of completing 2 sets of 20 repetitions for each exercise within 120 seconds per set. These routines have been designed based on existing exercises and routines provided by therapists. Tables 6.1, 6.2, and 6.3 display the execution data of the three examples. Each group of rows represents a different execution sample, and



(a) Therapist editing the routine of a patient.



(b) The changes made by the therapist are displayed in the patient application.

Figure 6.2: Connection between the two client applications.

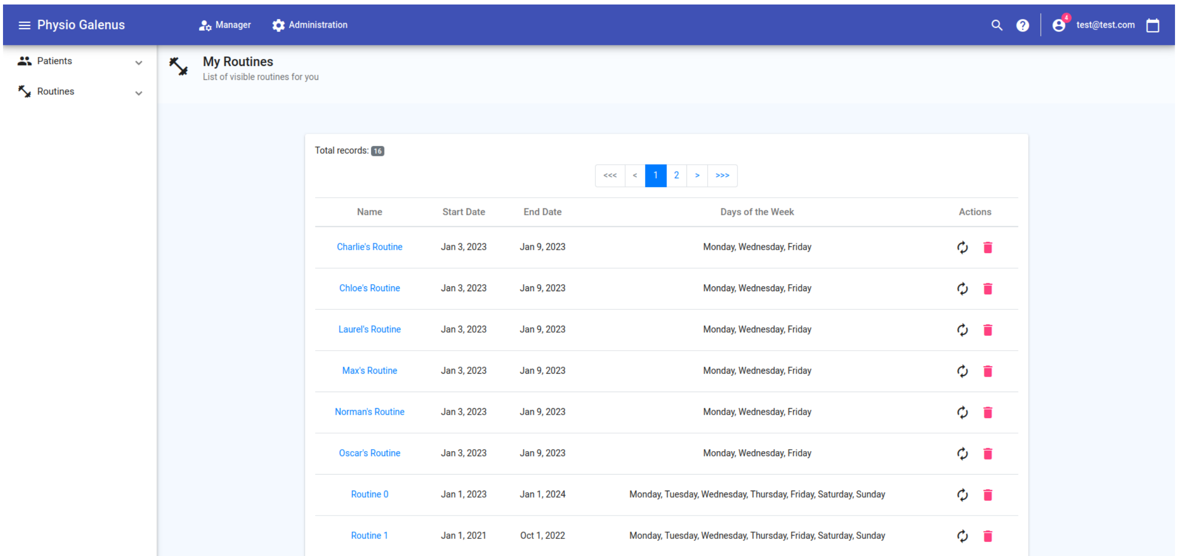


Figure 6.3: *My Routines* page.

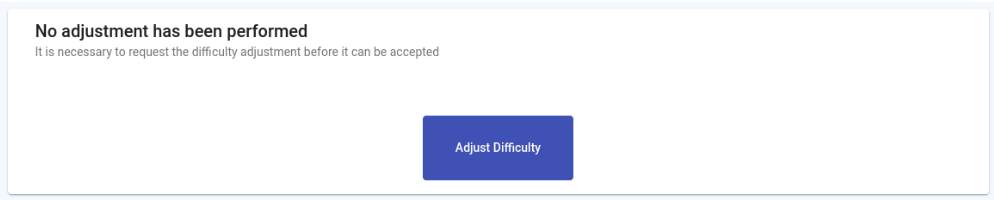


Figure 6.4: *Adjust Difficulty* page before the routine is adjusted.

each exercise is assigned one of these samples. All execution samples are distributed evenly across the 8 exercises. The column *Set Number* indicates which set does the row represent out of the total of all the sets executed during that day. The column *Compensation Level* describes the detected level of compensation throughout the execution of the set.

In the first example (Table 6.1), there are two types of exercises. The first type includes exercises where patients successfully executed all scheduled repetitions in a timely manner. In contrast, the second type involves exercises where patients were unable to complete all repetitions and took too much time. During the adjustment of this patient's routine, a new performance value is calculated for each exercise (see Figure 6.8), which are then aggregated into a single value. This performance variable contextualises the patient's execution data, since the same execution data may be a good or bad performance based on the patient's previous executions. The utilisation of the OWA operator in the aggregation process of the performance increment introduces a more nuanced approach. While the resulting performance increment is -1 when employing the arithmetic mean, it is -3 when employing the OWA, considering a performance range of [-20, 20]. The OWA operator takes into account the distribution of data and penalises the cases where there are exercises with significantly low performance. Utilising the arithmetic mean might suggest a performance increment equivalent to an execution where all exercises have mildly low performance. However, the

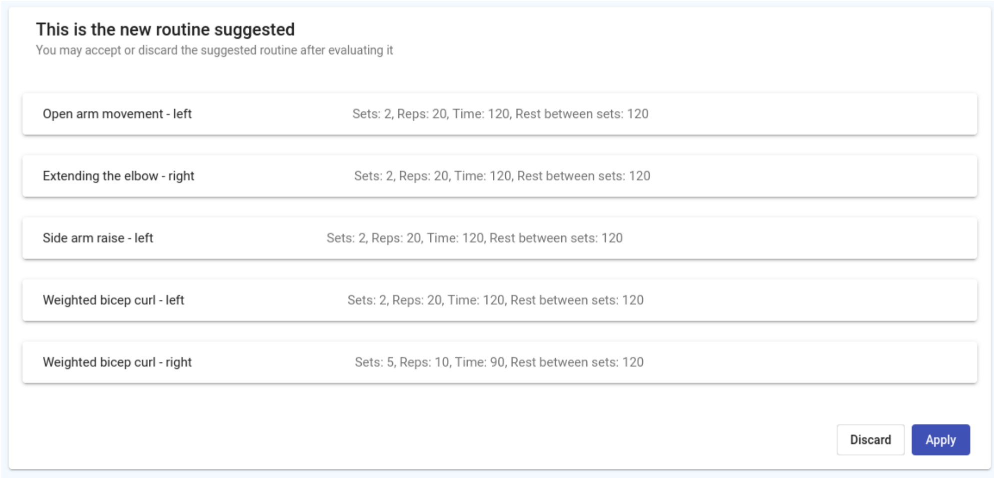


Figure 6.5: Former layout of the *Adjust Difficulty* page.

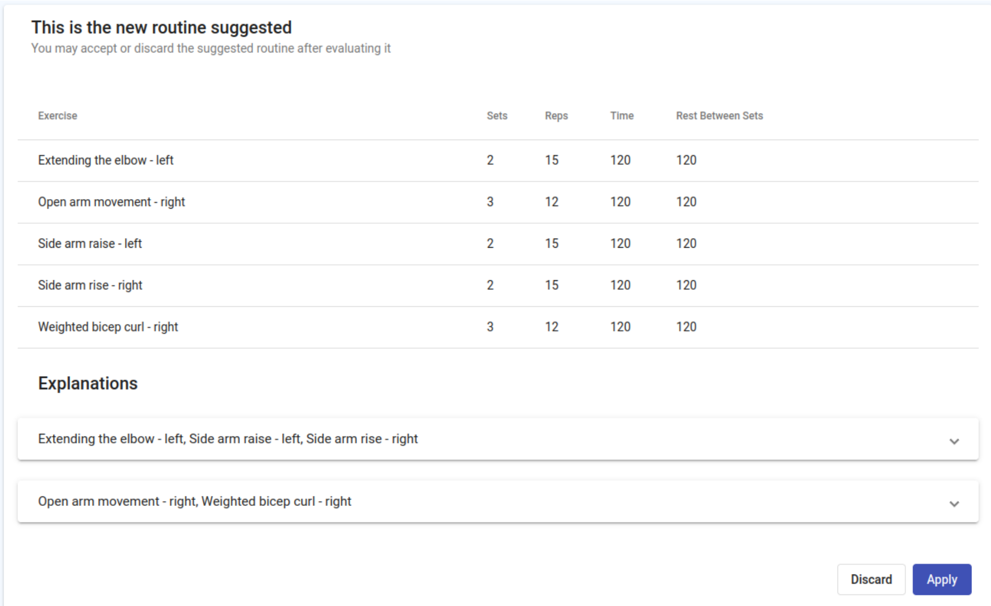


Figure 6.6: Current layout of the *Adjust Difficulty* page.

OWA operator takes into account the data distribution. It is important for rehabilitation evaluations to adopt a conservative approach when assessing performance, giving greater weight to exercises that were executed more poorly. Additionally, it is worth noting that although the absolute value of the increment may seem small, routines are expected to be adjusted multiple times over the course of the rehabilitation, and the effect of multiple adjustments accumulates over time.

In the second example (Table 6.2), an execution is presented where a patient has a high compensation level of 50%. This compensation level is considered elevated, as percentages close to 100% are rare, with most values being close to 0. By performing the adjustment on this patient's routine, the impact that compensation has on the FIS can be observed. It can be seen that the resulting routine has 10 less total repetitions per exercise than if the

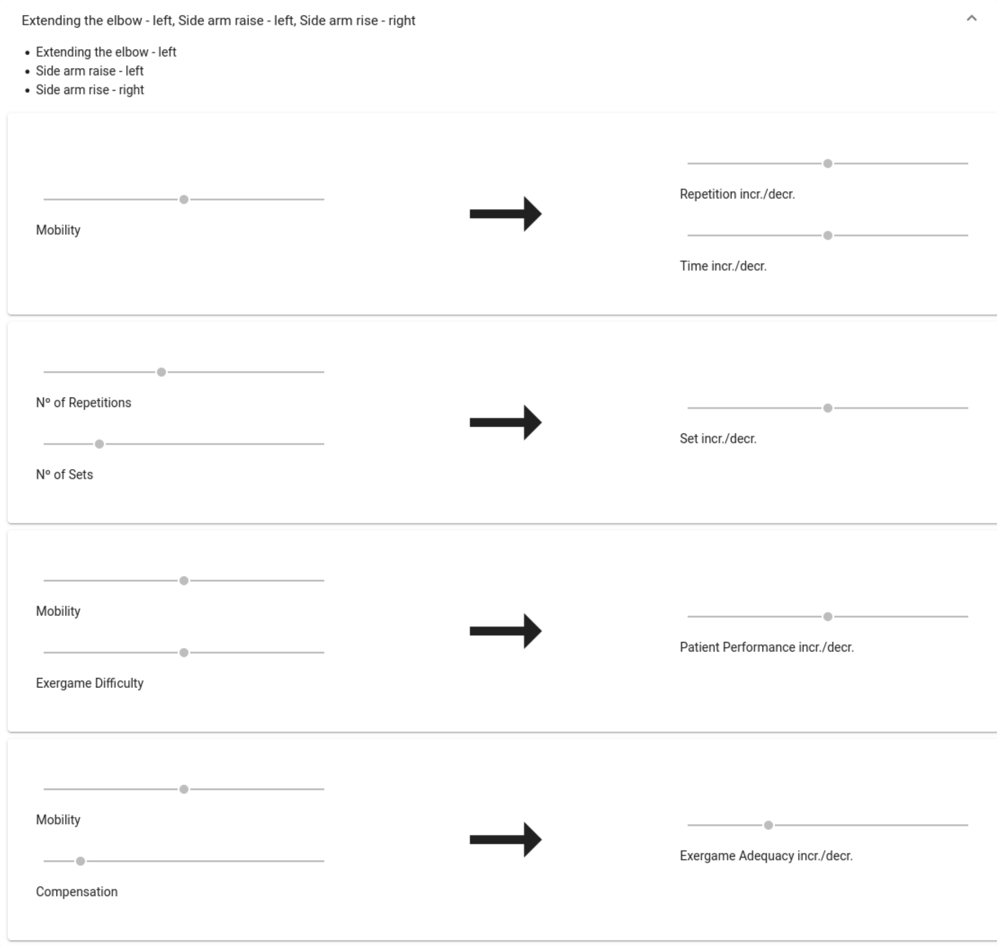


Figure 6.7: Explanations for the adjustments of a cluster of exercises.

compensation level were 0. In addition, the routine includes 2 fewer exercises. Furthermore, the adequacy of exercises is reduced by 26 percentage point in comparison to the scenario without compensation. However, in this specific example where all exercises are the same, this reduction in adequacy is not reflected in the new routine. The resulting routine can be seen in Figure 6.9.

In the third example (Table 6.3), there is a wide variety of exercise executions. The first two types of exercises reflect slightly positive execution data, while the second and third types display a slightly negative execution data. However, neither type of exercises is identical, representing a more realistic scenario compared to the other examples where exercises are too uniform. In the other examples, since each exercise type is drastically different, then the explanations of each exercise type are simply grouped together. In contrast, when adjusting this example data, it can be seen that the system is capable of grouping together the first and second types of exercises, as well as the third and forth types. This highlights the XAI module's capacity to discern patterns and group exercises in a meaningful way. The resulting routine can be seen in Figure 6.6.

Table 6.1: Routine Execution Data (Example 1)

Date	Set Number	Time (s)	Repetitions	Compensation Level (%)
Exercise Type 1				
2023-01-03	1	30	20	0
2023-01-03	2	30	20	0
2023-01-05	1	30	20	0
2023-01-05	2	30	20	0
2023-01-07	1	30	20	0
2023-01-07	2	30	20	0
Exercise Type 2				
2023-01-03	1	120	5	0
2023-01-05	1	120	5	0
2023-01-07	1	120	5	0

Table 6.2: Routine Execution Data (Example 2)

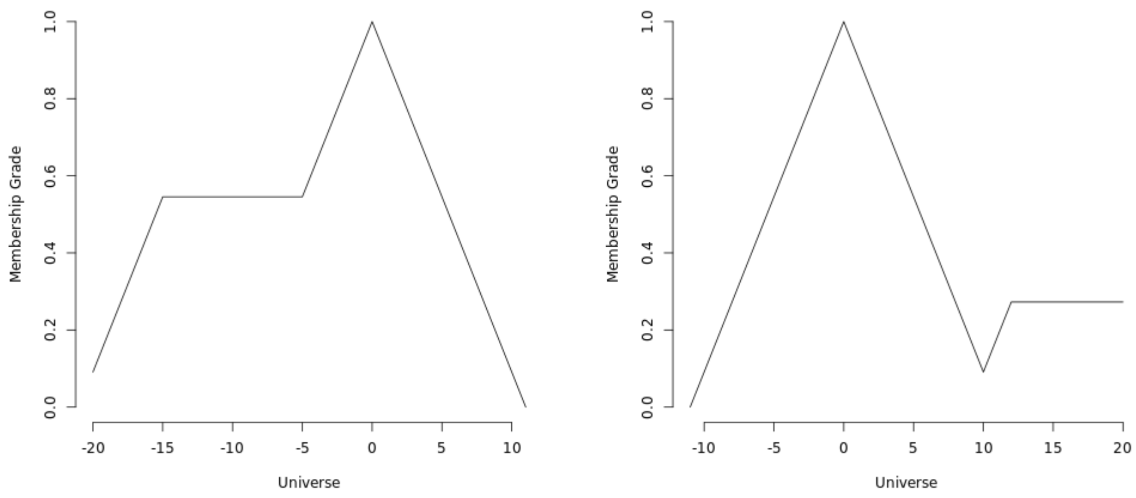
Date	Set Number	Time (s)	Repetitions	Compensation Level (%)
2023-01-03	1	60	10	50
2023-01-03	2	60	10	50
2023-01-05	1	60	10	50
2023-01-05	2	60	10	50
2023-01-07	1	60	10	50
2023-01-07	2	60	10	50

6.4 Project Statistics

In this section, the statistics generated from the work in the three repositories will be presented. The *therapist* repository is the one containing the server, as well as and the therapist application. The *webapp* repository contains the patient application. The *shared* repository holds common data between the other two repositories and acts as a git submodule. Statistics were extracted using the `git log -shortstat`, which can work with local Git repositories. The changes in each repository were merged, and the number of cumulative changes is displayed in Figure 6.10. It is worth noting that the late integration of changes in the repositories partially skews the statistics.

6.5 Project Costs and Resources

In this section, the cost estimation of the project is presented. The total cost of the project is composed of the cost of the hardware resources, and the development costs. No expenses were necessary for software resources. The hardware resources include two laptops, one for the patient, and one for the therapist. Their costs were estimated at 400€ each. No external sensors, such as a Kinect™, gyroscopes or accelerometers were required for this project. All pieces of software employed during the project were either open source, or did not require a



(a) Exercise Type 1.

(b) Exercise Type 2.

Figure 6.8: Performance increment before defuzzification for the routine specified in Table 6.1.

paid license.

To estimate the developer's costs, the time dedicated to the project was analysed, which spanned from August 2023 to October 2023, with an estimated total workload of 125 hours. The estimated time does not take into account the time spent during the creation of this document. The hourly salary was estimated using employment platforms such as InfoJobs[1], and to a lesser extent Indeed[2]. The hourly rate was was approximated to be of 37.50€. As a result, the total cost of the project is 5487.50€. This cost only covers development and excludes additional expenses such as documentation, maintenance, deployment, or technical support.

[1] `https://www.infojobs.net/`

[2] `https://indeed.com/`

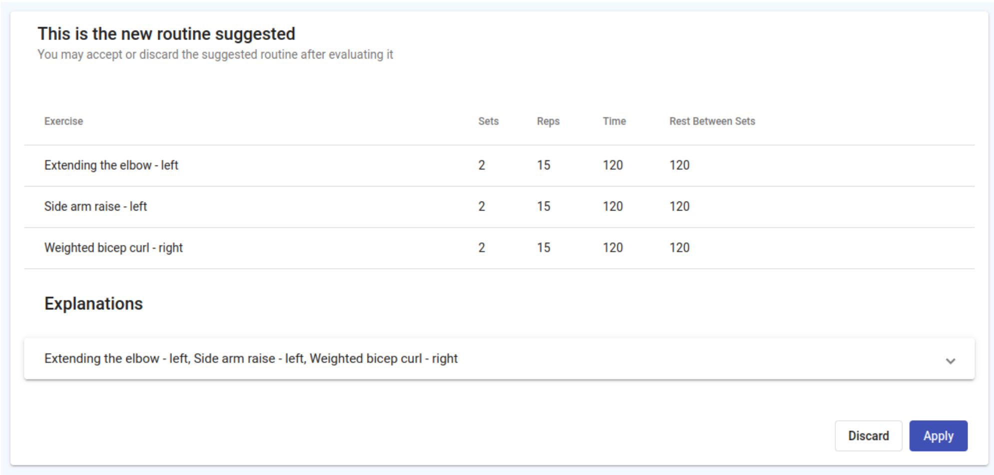


Figure 6.9: Suggested routine after adjusting the routine specified in Table 6.2.

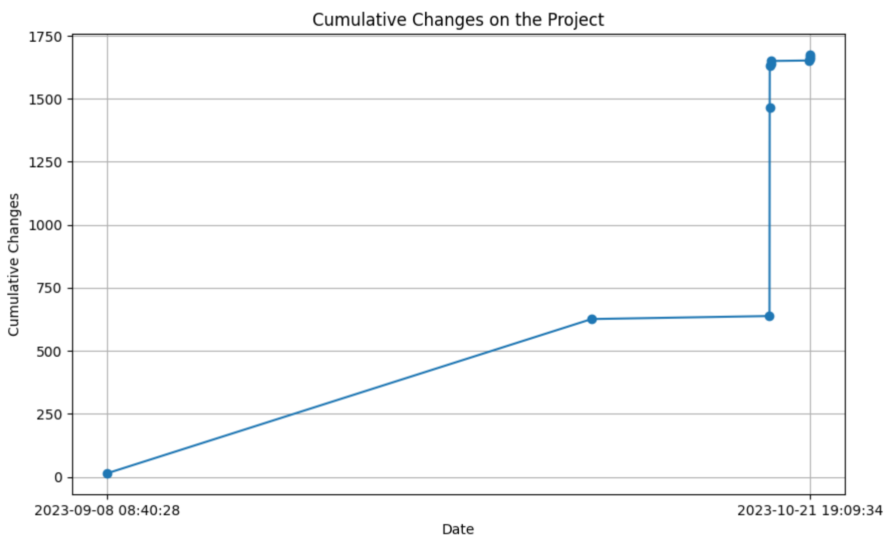


Figure 6.10: Graph with the cumulative changes across the three repositories.

Table 6.3: Routine Execution Data (Example 3)

Date	Set Number	Time (s)	Repetitions	Compensation Level (%)
Exercise Type 1				
2023-01-03	1	60	12	10
2023-01-03	2	60	8	11
2023-01-05	1	75	10	12
2023-01-05	2	60	11	10
2023-01-07	1	50	9	10
2023-01-07	2	60	14	25
Exercise Type 2				
2023-01-03	1	60	11	10
2023-01-03	2	60	9	11
2023-01-05	1	60	10	12
2023-01-05	2	70	12	10
2023-01-07	1	65	9	10
2023-01-07	2	60	14	25
Exercise Type 3				
2023-01-03	1	55	15	10
2023-01-03	2	75	20	8
2023-01-05	1	60	14	5
2023-01-05	2	85	18	6
2023-01-07	1	65	17	6
2023-01-07	2	55	17	9
Exercise Type 4				
2023-01-03	1	55	16	10
2023-01-03	2	75	20	8
2023-01-05	1	60	14	5
2023-01-05	2	85	17	6
2023-01-07	1	65	16	13
2023-01-07	2	55	17	9

Chapter 7

Conclusions and future work

In this chapter the reached objectives of the project will be discussed. Then, the addressed competences will be reviewed. Finally, potential future work and system improvements will be proposed.

7.1 Reached objectives

The general objective can be considered to have been completed. The degree to which therapists can delegate tasks to the system has been augmented by the introduction of compensatory movement detection and its integration into the decision logic of theDSS. Additionally, the usability of the system has been enhanced by refining the explanation generation process, making explanations more intuitive and concise. Below, the discussion will focus on the extent to which the partial objectives have been achieved.

- **Study and implementation of techniques for data aggregation in the context of automatic decision-making.** This objective can be considered complete, since multiple aggregation operators were studied to improve the decision making process of the system. As detailed in Section 2.1, various aggregation operators were studied. Then, their advantages and disadvantages were compared against the specific needs of the project, as discussed in Subsection 5.3.1. Ultimately, a suitable aggregation operator was chosen and implemented in the system.
- **Design and development of a module for the system that takes into account the compensatory movements of the patient.** An algorithm for detecting compensation movements was designed and developed, taking into consideration the various types of compensation movements performed by stroke patients and the capabilities of the available hardware for detection. Then, the insights derived from the study of aggregation operators, as discussed in the project, were utilised for the aggregation of compensation levels. Finally, the compensation level was integrated into the logic of the DSS, specifically within the FIS.
- **Design and implementation of an XAI algorithm focused on generating concise explanations.** This objective can be considered complete. The conciseness of the explanations provided by the XAI module was improved in two major ways. Firstly,

exercises were grouped together using clustering techniques to minimise redundancy. Given the expected overlap among exercises executed by the same patient, this approach allows to reduce the number of explanations provided to therapists without a major loss in the information being transmitted. Secondly, explanations with identical antecedents were grouped together. While the effectiveness of this approach in increasing conciseness was lower than the first one, it did not introduce any information loss.

- **Formalisation and modularization of the architecture of the remote rehabilitation system.** The formalisation of the architecture of the remote rehabilitation system can be considered complete. The process began with the definition of each module within the system, including the interactions between these modules. To offer a precise and unambiguous description of the architecture's structure, mathematical expressions were employed to formalise each individual module. This formalisation contributes to a more rigorous understanding of the system's components and their interactions, facilitating future development, design decisions, and system maintenance.

7.2 Addressed competences

In this section the addressed specific competences will be discussed.

- **[CE4] Ability to model, design, define architecture, implement, and manage, computer applications, systems, and services.** This competence was mainly addressed by the partial objective *Formalisation and modularization of the architecture of the remote rehabilitation system*. This process required not only defining the architecture of the system, but also modelling each one of the modules mathematically to provide an unambiguous description of the system. Additionally, this competence was addressed comprehensively during the design phase of the compensatory movement detection and the evolution of the XAI module. Since these developments were performed over an existing system, a thorough understanding of the existing architecture was crucial to manage and integrate the new changes effectively and in a maintainable way.
- **[CE12] Ability to apply mathematical, statistical, and artificial intelligence methods to model, design, and develop applications, and systems.** This competence was directly addressed in many of the partial objectives of the project. The study and application of aggregation operators, particularly OWA operators, is a prime example of this competence, since it involved a deep understanding of multiple mathematical and AI methods. Furthermore, the evolution of the XAI module also contributes to this competence. XAI is a subfield of AI, and one of the techniques used for improving explanation conciseness involved unsupervised learning. Finally, the inclusion of compensation movements into the DSS involved the application of mathematical and artificial intelligence methods. Various mathematical techniques were employed for

the detection of compensation movements, and the integration with the FIS required extensive knowledge of fuzzy logic.

7.3 Future work

One significant improvement that should be implemented in the system is the inclusion of functional exercises. While the existing exercises are enough for the system in its current state, expanding the repertoire of exercises, particularly incorporating functional exercises, will be crucial for the potential commercialisation of the tool. However, there are challenges associated with the incorporation of functional exercises, especially for the lower limbs. At the start of rehabilitation, patients typically execute easier exercises that generally target the upper limb. The difficulty of lower-limb exercises arises from the involvement of multiple joints, which are significantly more demanding than exercises involving a single joint. The increased challenge should not be overlooked when including lower-limb exercises. In addition, the current system is more oriented towards upper limb exercises, and adjustments are needed to accommodate a broader range of lower-limb exercises. Functional exercises, which often involve additional items such as water bottles, spoons, hair brushes, and stairs, present unique challenges. These items can either occlude the patient from the camera's view or complicate the positioning of the webcam. Moreover, certain functional exercises, like moving a water bottle, may require the patient's arm to be perpendicular to the camera's image plane, making tracking more difficult and less precise.

An essential aspect of future work is the validation of the system with therapists and patients. While the output of the DSS has undergone thorough review, validating the DSS with real patients and experienced therapists is crucial. Experts in the field can identify subtle details that may have not been apparent during the development process, and potential improvements to the DSS. This validation process would be a necessary step for certifying the system as a medical tool, which is a desirable objective when commercialising a medical tool. This future work is planned to be executed during the *FPI Industrial* granted to the author.

An important improvement for the system involves considering indications provided by therapists In the current version of the DSS, the adjustment of routines is an automatic process where the only input therapists can provide is either accepting or discarding the routine. Allowing therapists to have a greater level of influence would enable better management of edge cases and patient-specific circumstances. For instance, if a therapist deems it necessary for a patient to perform a particular exercise based on the patient's unique circumstances, the DSS should incorporate that exercise into the routine, disregarding the system's own assessment of exercise adequacy. Additionally, in cases where a patient has a condition, such as obesity, where excessive exercise could potentially harm the joints, therapists should have the possibility to limit the workload accordingly.

Another interesting possibility for future work would be making the DSS more proactive. The existing version of the DSS operates in a passive mode, where suggestions are provided only upon the therapist's request. This approach was initially chosen due to the potential risks associated with the autonomous adjustment of patient routines. However, if these risks are managed adequately, there is an opportunity to shift towards a more proactive system, where the DSS does not need explicit requests from therapists. This proactive approach has the potential to further alleviate therapists from routine tasks, allowing them to dedicate more quality time to direct patient care.

An additional improvement would be providing therapists with insights into patients' progress. This could be achieved by exposing relevant variables used in the decision-making process of the DSS that are currently hidden to the therapist, or by tracking new data. Furthermore, it could be possible to extract information from the extracted data, enhancing support for therapists' decision-making.

Finally, another important future task involves the deployment of the system in a production environment. Despite using Docker[1] to manage the environment, deploying the system to a production setting is a crucial step. This process will involve testing for compatibility issues, as well as verifying the scalability of the system.

7.4 Personal conclusion

The completion of my master's thesis marks a significant milestone in my academic journey, symbolising the culmination of the knowledge and skills acquired during this intensive program. I have been able to put into practice the newly acquired knowledge relating to AI, as well as the non-technical skills that I have developed throughout the master's course. In contrast to the bachelor's thesis, the master's thesis has a broader scope, and there is more emphasis on a future-oriented vision.

This has been a great challenge because it has allowed me to apply the knowledge and skills that I have acquired to a new aspect of a real problem that i was already familiar with. Providing solutions for major social issues such as stroke is a rewarding endeavour, and working on a field that I was not fully familiar with, such as healthcare, is an enriching experience.

Finally, it is worth mentioning that the end of this project is not the end of the system, since there are plans in progress for its commercialisation. I intend to continue contributing in this process through the *FPI Industrial* that I was granted.

[1] `https://www.docker.com/`

References

[AMR+00] Craig Anderson, Cliona Ni Mhurchu, Sally Rubenach, Michael Clark, Carol Spencer, y Adrian Winsor. Home or Hospital for Stroke Rehabilitation? Results of a Randomized Controlled Trial: II: Cost Minimization Analysis at 6 Months. *Stroke*, 31(5):1032–1037, Mayo 2000. url: `http://dx.doi.org/10.1161/01.STR.31.5.1032`.

[Awa05] MA Awad. A comparison between agile and traditional software development methodologies. *University of Western Australia*, 30:1–69, 2005.

[AWS+99] Eric Lewin Altschuler, Sidney B Wisdom, Lance Stone, Chris Foster, Douglas Galasko, D Mark E Llewellyn, y Vilayanur Subramanian Ramachandran. Rehabilitation of hemiparesis after stroke with a mirror. *The Lancet*, 353(9169):2035–2036, 1999.

[BdAS+20] Leonard Baatiema, Ama de-Graft Aikins, Fred S. Sarfo, Seye Abimbola, John K. Ganle, y Shawn Somerset. Improving the quality of care for people who had a stroke in a low-/middle-income country: A qualitative analysis of health-care professionals' perspectives. *Health Expectations*, 23(2):450–460, Enero 2020. url: `http://dx.doi.org/10.1111/hex.13027`.

[Cho54] Gustave Choquet. Theory of capacities. En *Annales de l'institut Fourier*, volume 5, páginas 131–295, 1954.

[CHSO18] Ravikiran Chimatapu, Hani Hagras, Andrew Starkey, y Gilbert Owusu. *Explainable AI and Fuzzy Logic Systems*, página 3–20. Springer International Publishing, 2018. url: `http://dx.doi.org/10.1007/978-3-030-04070-3_1`.

[CL00] M. C. Cirstea y M. F. Levin. Compensatory strategies for reaching in stroke. *Brain*, 123(5):940–953, Mayo 2000. url: `https://doi.org/10.1093/brain/123.5.940`.

[CLZ+19] Siqi Cai, Guofeng Li, Xiaoya Zhang, Shuangyuan Huang, Haiqing Zheng, Ke Ma, y Longhan Xie. Detecting compensatory movements of stroke survivors using pressure distribution data and machine learning algorithms. *Journal of NeuroEngineering and Rehabilitation*, 16(1), Noviembre 2019. url: https://doi.org/10.1186/s12984-019-0609-6.

[CMM02] Tomasa Calvo, Gaspar Mayor, y Radko Mesiar. *Aggregation Operators: New Trends and Applications*. Physica-Verlag HD, 2002. url: https://doi.org/10.1007/978-3-7908-1787-4.

[CPC19] Diogo V. Carvalho, Eduardo M. Pereira, y Jaime S. Cardoso. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics*, 8(8):832, Julio 2019. url: https://doi.org/10.3390/electronics8080832.

[CPXB23] Ahmad Chaddad, Jihao Peng, Jian Xu, y Ahmed Bouridane. Survey of Explainable AI Techniques in Healthcare. *Sensors*, 23(2):634, Enero 2023. url: http://dx.doi.org/10.3390/s23020634.

[CS87] Janet Carr y Roberta B Shepherd. A motor relearning programme for stroke. *(No Title)*, 1987.

[DEP+23] Neha Das, Satoshi Endo, Sabrina Patel, Carmen Krewer, y Sandra Hirche. Online detection of compensatory strategies in human movement with supervised classification: a pilot study. *Frontiers in Neurorobotics*, 17, Julio 2023. url: http://dx.doi.org/10.3389/fnbot.2023.1155826.

[DGB+21] Alexander W. Dromerick, Shashwati Geed, Jessica Barth, Kathaleen Brady, Margot L. Giannetti, Abigail Mitchell, Matthew A. Edwardson, Ming T. Tan, Yizhao Zhou, Elissa L. Newport, y Dorothy F. Edwards. Critical Period After Stroke Study (CPASS): A phase II clinical trial testing an optimal time for motor recovery after stroke in humans. *Proceedings of the National Academy of Sciences*, 118(39), Septiembre 2021. url: http://dx.doi.org/10.1073/pnas.2026676118.

[Dob04] Bruce H Dobkin. Strategies for stroke rehabilitation. *The Lancet Neurology*, 3(9):528–536, Septiembre 2004. url: http://dx.doi.org/10.1016/S1474-4422(04)00851-8.

[Don18] Eric S Donkor. Stroke in the 21st Century: A Snapshot of the Burden, Epidemiology, and Quality of Life. *Stroke Research and Treatment*, 2018:3238165, Nov 2018.

[Dun94] Pamela W Duncan. Stroke Disability. *Physical Therapy*, 74(5):399–407, Mayo 1994. url: http://dx.doi.org/10.1093/ptj/74.5.399.

[Etz18] Amitai Etzioni. Pros and Cons of Autonomous Weapons Systems (with Oren Etzioni). En *Library of Public Policy and Public Administration*, páginas 253–263. Springer International Publishing, 2018. url: https://doi.org/10.1007/978-3-319-69623-2_16.

[FBN+22] Valery L Feigin, Michael Brainin, Bo Norrving, Sheila Martins, Ralph L Sacco, Werner Hacke, Marc Fisher, Jeyaraj Pandian, y Patrice Lindsay. World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. *International Journal of Stroke*, 17(1):18–29, Enero 2022. url: http://dx.doi.org/10.1177/17474930211065917.

[FY98] Dimitar Filev y Ronald R. Yager. On the issue of obtaining OWA operator weights. *Fuzzy Sets and Systems*, 94(2):157–169, Marzo 1998. url: https://doi.org/10.1016/s0165-0114(96)00254-0.

[GL09] Michel Grabisch y Christophe Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1):247–286, Octubre 2009. url: https://doi.org/10.1007/s10479-009-0655-8.

[GORB21] Marzyeh Ghassemi, Luke Oakden-Rayner, y Andrew L Beam. The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, 3(11):e745–e750, Noviembre 2021. url: https://doi.org/10.1016/s2589-7500(21)00208-9.

[Gra11] Michel Grabisch. OWA Operators and Nonadditive Integrals. En *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*, páginas 3–15. Springer Berlin Heidelberg, 2011. url: https://doi.org/10.1007/978-3-642-17910-5_1.

[GS11] Dayna Griffiths y Jonathan Sturm. Epidemiology and Etiology of Young Stroke. *Stroke Research and Treatment*, 2011:1–9, 2011. url: http://dx.doi.org/10.4061/2011/209370.

[GSC+19] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, y Guang-Zhong Yang. XAI—Explainable artificial intelligence. *Science Robotics*, 4(37), Diciembre 2019. url: https://doi.org/10.1126/scirobotics.aay7120.

[HI00] James A Highsmith III. Adaptive software development: a collaborative approach to managing complex systems, 2000.

[HvLGZ02] Henk T. Hendricks, Jacques van Limbeek, Alexander C. Geurts, y Machiel J. Zwarts. Motor recovery after stroke: A systematic review of the literature. *Archives of Physical Medicine and Rehabilitation*, 83(11):1629–1637, Noviembre 2002. url: `http://dx.doi.org/10.1053/apmr.2002.35473`.

[IHR+22] Mohammed Saidul Islam, Iqram Hussain, Md Mezbaur Rahman, Se Jin Park, y Md Azam Hossain. Explainable Artificial Intelligence Model for Stroke Prediction Using EEG Signal. *Sensors*, 22(24):9859, Diciembre 2022. url: `https://doi.org/10.3390/s22249859`.

[KLM+18] Elżbieta Kuźma, Ilianna Lourida, Sarah F. Moore, Deborah A. Levine, Obioha C. Ukoumunne, y David J. Llewellyn. Stroke and dementia risk: A systematic review and meta-analysis. *Alzheimer's & Dementia*, 14(11):1416–1426, Agosto 2018. url: `http://dx.doi.org/10.1016/j.jalz.2018.06.3061`.

[KLO+20] Åsa Karlsson, Nina Lindelöf, Birgitta Olofsson, Monica Berggren, Yngve Gustafson, Peter Nordström, y Michael Stenvall. Effects of Geriatric Interdisciplinary Home Rehabilitation on Independence in Activities of Daily Living in Older People With Hip Fracture: A Randomized Controlled Trial. *Archives of Physical Medicine and Rehabilitation*, 101(4):571–578, Abril 2020. url: `http://dx.doi.org/10.1016/j.apmr.2019.12.007`.

[KS09] Mike Keith y Merrick Schnicariol. *Object-Relational Mapping*, página 69–106. Apress, 2009. url: `http://dx.doi.org/10.1007/978-1-4302-1957-6_4`.

[Kun23] Dominika Kunertova. Drones have boots: Learning from Russia's war in Ukraine. *Contemporary Security Policy*, 44(4):576–591, Octubre 2023. url: `https://doi.org/10.1080/13523260.2023.2262792`.

[KX20] Daniel Kuriakose y Zhifeng Xiao. Pathophysiology and Treatment of Stroke: Present Status and Future Perspectives. *International Journal of Molecular Sciences*, 21(20):7609, Oct 2020.

[LBK11] Peter Langhorne, Julie Bernhardt, y Gert Kwakkel. Stroke rehabilitation. *The Lancet*, 377(9778):1693–1702, Mayo 2011. url: `http://dx.doi.org/10.1016/S0140-6736(11)60325-5`.

[LGHJC22] Gabriel Lima, Nina Grgić-Hlača, Jin Keun Jeong, y Meeyoung Cha. The Conflict Between Explainable and Accountable Decision-Making Algorithms. En *2022 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Junio 2022. url: `https://doi.org/10.1145/3531146.3534628`.

[LL17] Scott M Lundberg y Su-In Lee. A Unified Approach to Interpreting Model Predictions. En I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, y R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. url: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[LLPB15] Mindy F. Levin, Dario G. Liebermann, Yisrael Parmet, y Sigal Berman. Compensatory Versus Noncompensatory Shoulder Movements Used for Reaching in Stroke. *Neurorehabilitation and Neural Repair*, 30(7):635–646, Octubre 2015. url: https://doi.org/10.1177/1545968315613863.

[LOS+22] Hui Wen Loh, Chui Ping Ooi, Silvia Seoni, Prabal Datta Barua, Filippo Molinari, y U Rajendra Acharya. Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022). *Computer Methods and Programs in Biomedicine*, 226:107161, Noviembre 2022. url: https://doi.org/10.1016/j.cmpb.2022.107161.

[LSRC14] Sarah Llanque, Lynette Savage, Neal Rosenburg, y Michael Caserta. Concept Analysis: Alzheimer's Caregiver Stress: Alzheimer's Caregiver Stress. *Nursing Forum*, 51(1):21–31, Mayo 2014. url: http://dx.doi.org/10.1111/nuf.12090.

[MA75] E.H. Mamdani y S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, Enero 1975. url: http://dx.doi.org/10.1016/S0020-7373(75)80002-2.

[Mar00] Jean-Luc Marichal. On Choquet and Sugeno integrals as aggregation functions. *Fuzzy measures and integrals-theory and applications*, páginas 247–272, 2000.

[MNR+21] Tommi Mikkonen, Jukka K Nurminen, Mikko Raatikainen, Ilenia Fronza, Niko Mäkitalo, y Tomi Männistö. Is machine learning software just software: A maintainability view. En *Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference, SWQD 2021, Vienna, Austria, January 19–21, 2021, Proceedings 13*, páginas 94–105. Springer, 2021.

[MS93] Toshiaki Murofushi y Michio Sugeno. Some quantities represented by the Choquet integral. *Fuzzy sets and systems*, 56(2):229–235, 1993. url: https://doi.org/10.1016/0165-0114(93)90148-B.

[MS00] H. B. Mitchell y P. A. Schaefer. On ordering fuzzy numbers. *International Journal of Intelligent Systems*, 15(11):981–993, 2000. url: `https://doi.org/10.1002/1098-111x(200011)15:11<981::aid-int1>3.0.co;2-z`.

[MVI03] Eduardo Martínez-Vila y Pablo Irimia. The Cost of Stroke. *Cerebrovascular Diseases*, 17(Suppl. 1):124–129, Diciembre 2003. url: `http://dx.doi.org/10.1159/000074804`.

[NDA23] Sajid Nazir, Diane M. Dickson, y Muhammad Usman Akram. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in Biology and Medicine*, 156:106668, Abril 2023. url: `http://dx.doi.org/10.1016/j.compbiomed.2023.106668`.

[NPH13] Hyo Suk Nam, Eunjeong Park, y Ji Hoe Heo. Facilitating Stroke Management using Modern Information Technology. *Journal of Stroke*, 15(3):135, 2013. url: `https://doi.org/10.5853/jos.2013.15.3.135`.

[NR02] Kenji Narushima y Robert G. Robinson. Stroke-related depression. *Current Atherosclerosis Reports*, 4(4):296–303, Julio 2002. url: `http://dx.doi.org/10.1007/s11883-002-0009-3`.

[O'H87] Michael O'Hagan. Fuzzy decision aids. En *21th Asilomar Conference on Signals, Systems and Computers*, volume 2, páginas 624–628. IEEE and Maple Press, 1987.

[PCRM$^+$20] Seyedeh Neelufar Payrovnaziri, Zhaoyi Chen, Pablo Rengifo-Moreno, Tim Miller, Jiang Bian, Jonathan H Chen, Xiuwen Liu, y Zhe He. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *Journal of the American Medical Informatics Association*, 27(7):1173–1185, Mayo 2020. url: `https://doi.org/10.1093/jamia/ocaa053`.

[RIZ$^+$17] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, y Andrew Y. Ng. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, 2017. url: `https://arxiv.org/abs/1711.05225`.

[RPOD95] I. E. Rolfe, S.-A. Pearson, D. L. O'Connell, y J. A. Dickinson. Finding solutions to the rural doctor shortage: the roles of selection versus undergraduate medical education at Newcastle. *Australian and New Zealand Journal of Medicine*, 25(5):512–517, Octubre 1995. url: `http://dx.doi.org/10.1111/j.1445-5994.1995.tb01497.x`.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, y Carlos Guestrin. "Why Should I Trust You?". En *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Agosto 2016. url: `https://doi.org/10.1145/2939672.2939778`.

[RWGB17] Rajiv Ranganathan, Rui Wang, Rani Gebara, y Subir Biswas. Detecting Compensatory Trunk Movements in Stroke Survivors using a Wearable System. En *Proceedings of the 2017 Workshop on Wearable Systems and Applications*. ACM, Junio 2017. url: `https://doi.org/10.1145/3089351.3089353`.

[SBV+22] Deepti Saraswat, Pronaya Bhattacharya, Ashwin Verma, Vivek Kumar Prasad, Sudeep Tanwar, Gulshan Sharma, Pitshou N. Bokoro, y Ravi Sharma. Explainable AI for Healthcare 5.0: Opportunities and Challenges. *IEEE Access*, 10:84486–84517, 2022. url: `https://doi.org/10.1109/access.2022.3197671`.

[Sch96] Douglas C. Schmidt. *Architectural Patterns*. Wiley, 1996.

[SFA10] Joseph R. Shiber, Emily Fontane, y Ademola Adewale. Stroke registry: hemorrhagic vs ischemic strokes. *The American Journal of Emergency Medicine*, 28(3):331–333, Marzo 2010. url: `http://dx.doi.org/10.1016/j.ajem.2008.10.026`.

[SLNS18] Peter Svenmarck, Linus Luotsinen, Mattias Nilsson, y Johan Schubert. Possibilities and challenges for artificial intelligence in military applications. En *Proceedings of the NATO Big Data and Artificial Intelligence for Military Decision Making Specialists' Meeting*, páginas 1–16, 2018.

[Sug74] M. Sugeno. *Theory of Fuzzy Integrals and Its Applications*. PhD thesis, Tokyo Institute of Technology, Tokyo, 1974.

[Sur19] Harry Surden. Artificial Intelligence and Law: An Overview. *Georgia State University Law Review*, 35, 2019. U of Colorado Law Legal Studies Research Paper No. 19-22. url: `https://ssrn.com/abstract=3411869`.

[SW00] Sue Stephenson y Rose Wiles. Advantages and Disadvantages of the Home Setting for Therapy: Views of Patients and Therapists. *British Journal of Occupational Therapy*, 63(2):59–64, Febrero 2000. url: `http://dx.doi.org/10.1177/030802260006300203`.

[TN07] Vicenç Torra y Yasuo Narukawa. *Modeling decisions: information fusion and aggregation operators*. Springer Science & Business Media, 2007.

[TV14] Robert W. Teasell y Ricardo Viana. Evidence-based benefit of rehabilitation after stroke. En *Textbook of Neural Repair and Rehabilitation*, páginas 601–614. Cambridge University Press, Abril 2014. url: https://doi.org/10.1017/cbo9780511995590.049.

[Twi51] Thomas E Twitchell. The restoration of motor function following hemiplegia in man. *Brain*, 74(4):443–480, 1951.

[War98] CP Warlow. Epidemiology of stroke. *The Lancet*, 352:S1–S4, Octubre 1998. url: http://dx.doi.org/10.1016/S0140-6736(98)90086-1.

[WDBT+17] Qi Wang, Liesbet De Baets, Annick Timmermans, Wei Chen, Luca Giacolini, Thomas Matheve, y Panos Markopoulos. Motor Control Training for the Shoulder with Smart Garments. *Sensors*, 17(7):1687, Julio 2017. url: http://dx.doi.org/10.3390/s17071687.

[WDK95] Xuzhu Wang, B. De Baets, y E. Kerre. A comparative study of similarity measures. *Fuzzy Sets and Systems*, 73(2):259–268, 1995. url: https://www.sciencedirect.com/science/article/pii/016501149400308T.

[WFY+22] Xiaoyi Wang, Yan Fu, Bing Ye, Jessica Babineau, Yong Ding, y Alex Mihailidis. Technology-Based Compensation Assessment and Detection of Upper Extremity Activities of Stroke Survivors: Systematic Review. *Journal of Medical Internet Research*, 24(6):e34307, Junio 2022. url: https://doi.org/10.2196/34307.

[WMR21] Sandra Wachter, Brent Mittelstadt, y Chris Russell. Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI. *Computer Law & Security Review*, 41:105567, Julio 2021. url: https://doi.org/10.1016/j.clsr.2021.105567.

[WV18] Bernhard Waltl y Roland Vogl. Explainable artificial intelligence the new frontier in legal informatics. *Jusletter IT*, 4:1–10, 2018.

[Yag88] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988. url: https://doi.org/10.1109/21.87068.

[Yag93] Ronald R. Yager. Families of OWA operators. *Fuzzy Sets and Systems*, 59(2):125–148, Octubre 1993. url: https://doi.org/10.1016/0165-0114(93)90194-m.

[Yag05] Ronald R. Yager. Extending multicriteria decision making by mixing t-norms and OWA operators. *International Journal of Intelligent Systems*, 20(4):453–474, 2005. url: `https://doi.org/10.1002/int.20075`.

[Yag17] Ronald R. Yager. OWA Aggregation of Probability Distributions Using the Probabilistic Exceedance Method. En *Fuzzy Sets, Rough Sets, Multisets and Clustering*, páginas 277–289. Springer International Publishing, 2017. url: `https://doi.org/10.1007/978-3-319-47557-8_16`.

[YF94] Ronald R Yager y Dimitar P Filev. Parameterized and-uke and or-like owa operators. *International Journal of General System*, 22(3):297–316, 1994.

[YF99] R.R. Yager y D.P. Filev. Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29(2):141–150, Abril 1999. url: `https://doi.org/10.1109/3477.752789`.

[Zad65] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, Junio 1965. url: `https://doi.org/10.1016/s0019-9958(65)90241-x`.

[Zad75] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, 8(3):199–249, 1975. url: `https://doi.org/10.1016/0020-0255(75)90036-5`.

[Zad83] Lotfi A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9:149–184, 1983.

[ZCJG08] Shang-Ming Zhou, Francisco Chiclana, Robert I. John, y Jonathan M. Garibaldi. Type-1 OWA operators for aggregating uncertain information with uncertain weights induced by type-2 linguistic quantifiers. *Fuzzy Sets and Systems*, 159(24):3281–3296, Diciembre 2008. url: `https://doi.org/10.1016/j.fss.2008.06.018`.

[ZCJG11] Shang-Ming Zhou, Francisco Chiclana, Robert I. John, y Jonathan M. Garibaldi. Alpha-Level Aggregation: A Practical Approach to Type-1 OWA Operation for Aggregating Uncertain Information with Applications to Breast Cancer Treatments. *IEEE Transactions on Knowledge and Data Engineering*, 23(10):1455–1468, Octubre 2011. url: `https://doi.org/10.1109/tkde.2010.191`.

[ZLL+18] Ying Xuan Zhi, Michelle Lukasik, Michael H. Li, Elham Dolatabadi, Rosalie H. Wang, y Babak Taati. Automatic Detection of Compensation During Robotic Stroke Rehabilitation Therapy. *IEEE Journal of Translational Engineering in Health and Medicine*, 6:1–7, 2018. url: `https://doi.org/10.1109/jtehm.2017.2780836`.

Este documento fue editado y tipografiado con LaTeX empleando
la clase **esi-tfg** (versión 0.20181017) que se puede encontrar en:
`https://bitbucket.org/esi_atc/esi-tfg`