Personalized Interaction Models of Socially Assistive Robots for Monitoring and Guiding Rehabilitation Exercises

Diogo da Costa Gregório

Electrical and Computer Engineering

Supervisor(s): Professor Min Lee   Professor Plinio Moreno López

January 2023

Chapter 1

Introduction

Socially Assistive Robots (SARs) are robotic systems designed to assist and interact with humans in various settings. In recent years, there has been growing interest in using SARs for healthcare applications, including rehabilitation exercises. SARs can provide personalized interaction models that monitor and guide patients during their rehabilitation exercises, leading to improved patient outcomes.

Personalized interaction models involve the customization of the SAR's interaction with the patient based on their specific needs and preferences. By using sensors and machine learning algorithms, the SAR can adapt to the patient's performance, providing feedback and guidance in real-time. This approach can enhance the patient's engagement and motivation, leading to better compliance with their rehabilitation program.

In this context, SARs can be particularly beneficial for patients who require long-term rehabilitation or who have limited access to healthcare services. Furthermore, the use of SARs can reduce the workload of healthcare professionals, allowing them to focus on more complex tasks and providing better care to patients [1].

Overall, the development of personalized interaction models for SARs in rehabilitation exercises is an exciting area of research with significant potential to improve patient outcomes and increase access to healthcare services.

1.1 Motivation

Rehabilitation is an essential aspect of healthcare that aims to restore patients' physical, cognitive, and psychological function. However, many patients struggle with adherence to their rehabilitation program (around 42% according to [2]), leading to poor outcomes and increased healthcare costs. Socially Assistive Robots (SARs) have shown great potential in assisting patients with their rehabilitation exercises, but the development of personalized interaction models is necessary to maximize their effectiveness.

Personalized interaction models involve customizing the SAR's interaction with the patient based on their specific needs and preferences. By leveraging sensors and machine learning algorithms, the SAR can adapt to the patient's performance, providing feedback and guidance in real-time. This approach can enhance the patient's engagement and motivation, leading to better compliance with their rehabilitation program and ultimately better outcomes.

There is currently a significant gap in the literature regarding the development of personalized interaction models for SARs in rehabilitation exercises. Therefore, there is a need for further research in this area to understand how these models can be optimized to improve patient outcomes and increase access to healthcare services. The aim is to contribute to this growing field of research by investigating the effectiveness of personalized interaction models in SARs for monitoring and guiding rehabilitation exercises.

The results of this research will have implications for the design and development of SARs in healthcare, as well as the delivery of rehabilitation services. Additionally, it will provide valuable insights into how personalized interaction models can be leveraged to improve patient outcomes and increase access to healthcare services.

1.2 Objectives

Following the motivation, this paper identifies factors that may influence the success and effectiveness of personalized interaction models in SARs for monitoring and guiding rehabilitation exercises as well as its feasibility and impact in patient adherence regarding engagement, effectiveness and motivation in comparison to traditional rehabilitation interventions.

The main objective is to use reinforcement learning and apply both machine learning and rule-base modeling to automatically assess a patient's rehabilitation exercise and provide personalized corrective feedback by analizing the data from the input generated.

1.3 Report Outline

This report is organized as follows: Chapter 1 presents the introduction, motivation and the outline of the report. Chapter 2 briefly explains the model of reinforcement learning, states some related work and studies previously produced. Chapter 3 displays the state of the art of SARs and the models that were used. Finally, in Chapter 4 conclusions are derived from previous work and the future work objectives and plan are indicated.

Chapter 2

Theoretical Background

The present chapter concentrates on exposing the theory behind the reinforcement learning model, show some related work that was taken into account during the realization of the project and specify the study designs on stroke rehabilitation.

2.1 Reinforcement Learning

Reinforcement learning (RL) [3] is a powerful framework that can be utilized in the development of personalized interaction models for socially assistive robots in monitoring and guiding rehabilitation exercises. RL is a subfield of machine learning where an agent learns to make sequential decisions through interactions with an environment to maximize a notion of cumulative reward.

In the context of personalized interaction models for rehabilitation exercises, RL can be applied to enable the robot to learn optimal strategies for monitoring and guiding the exercises based on individual patient characteristics, preferences, and progress. Here are some key aspects and potential applications of RL in this domain:

- **State Representation**: RL requires an appropriate representation of the environment state. In the case of rehabilitation exercises, the state could include information about the patient's current physical condition, range of motion, joint angles, muscle activity, exercise history, and any other relevant contextual factors. This state representation enables the robot to perceive and understand the current situation.

- **Action Space**: The action space represents the set of actions that the robot can take in response to a given state. In the context of rehabilitation exercises, actions could include providing verbal instructions, demonstrating the correct form, adjusting the resistance level, providing feedback or encouragement, and adapting the exercise program based on the patient's progress, with this case being more focused on providing feedback.

- **Reward Design**: The reward function defines the objective or goal of the RL agent. In the case of personalized interaction models for rehabilitation exercises, the reward function should be designed to encourage desired behaviors, such as adherence to the exercise program, correct execution of exercises, and improvements in the patient's performance or health outcomes. The reward function should be carefully designed to provide meaningful feedback and align with the goals of the rehabilitation process.

- **Exploration and Exploitation**: RL algorithms typically involve a trade-off between exploration (trying out different actions to gather information) and exploitation (leveraging existing knowledge to maximize rewards). Balancing exploration and exploitation is crucial in personalized interaction models to ensure that the robot can learn from interactions with the patient while also providing effective guidance and support.

- **Personalization and Adaptation**: One of the main advantages of RL in personalized interaction models is its ability to adapt to individual patients. By continually monitoring and evaluating the patient's progress and responses, the robot can dynamically adjust its behavior and strategies to suit the specific needs and preferences of each patient. This adaptability allows for a personalized and tailored rehabilitation experience.

- **Policy Learning**: RL algorithms learn policies that map states to actions based on the observed rewards. By optimizing the policy, the robot can learn to make informed decisions about how to monitor and guide rehabilitation exercises effectively. Techniques such as value iteration, Q-learning, or policy gradient methods can be employed to train the RL agent.

- **Reinforcement Learning Challenges**: While RL offers great potential, there are challenges in its application to personalized interaction models for rehabilitation exercises. These challenges include sample efficiency (learning from limited interactions), handling high-dimensional state and action spaces, addressing safety concerns, and ensuring ethical considerations in the human-robot interaction.

By employing RL techniques in the development of personalized interaction models, socially assistive robots can offer adaptive, tailored, and engaging support to individuals undergoing rehabilitation exercises. RL enables the robot to learn from the patients and doctors' interactions, improve its guidance strategies over time, and contribute to more effective and personalized rehabilitation outcomes.

2.1.1 Markov Decision Process

In reinforcement learning (RL), Markov Decision Processes (MDPs) [4] [5] are fundamental models used to formalize and solve sequential decision-making problems. MDPs provide a framework for studying the interaction between an agent and an environment, where the agent learns to make decisions based on feedback in the form of rewards.

The RL problem is first formulated as an MDP. This involves defining the set of states S, actions A, transition probabilities $P(s'|s,a)$, immediate rewards $R(s,a,s')$, the discount factor γ, the start state s_0 and the horizon$H$, with a goal function (i.e. equation 2.2). In an MDP, the objective is to find an optimal policy π that maximizes the expected sum of rewards: $\pi^* : S \times 0 : H \rightarrow A$, in other words, the agent

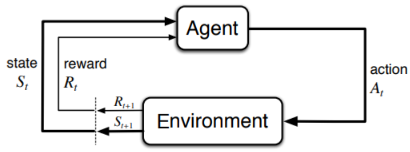

Figure 2.1: Markov Decision Process.

follows a policy, that can be stochastic or deterministic, which is a mapping from states to actions that guides its decision-making with the objective is to find an optimal policy that maximizes the expected cumulative reward over time.

$$max_{\pi} E\left[\sum_{t=0}^{H} \gamma^t R(S_t, A_t, S_{t+1})|\pi\right] \tag{2.1}$$

Regarding agent-environment interaction, the RL agent interacts with the environment in a sequence of discrete time steps. At each time step, the agent observes the current state, selects an action, and receives a reward from the environment.

The value function V is used to evaluate the quality of states or state-action pairs under a given policy. It represents the expected cumulative reward starting from a particular state and following the policy. The value function is typically estimated or iteratively updated using techniques such as dynamic programming or function approximation. The optimal value function V^* **??** will then represent the sum of discounted rewards when starting from state s and acting optimally.

$$max_{\pi} E\left[\sum_{t=0}^{H} \gamma^t R(S_t, A_t, S_{t+1})|\pi, s_0 = s\right] \tag{2.2}$$

Based on V^*, the current state and the policy being followed the agent selects actions. This action selection process can be deterministic (e.g., choosing the action with the highest value) or stochastic (e.g., sampling actions according to their probabilities). To find the optimal value for state s when there is a horizon H it is applied the Bellman equations (2.3 & 2.4), which state that at convergence it is found the optimal value function V^* for the discounted infinite horizon problem.

$$\forall S \in S : V^*(s) = max_A \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V^*(s')] \tag{2.3}$$

$$\forall S \in S : \pi^*(s) = argmax_{a \in A} \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V^*(s')] \tag{2.4}$$

To learn and explore, the agent interacts with the environment, by receiving rewards, and updating its policy or value function. RL algorithms employ various learning methods such as model-free, such as **Q-learning**. Exploration strategies are also employed to encourage the agent to explore different actions and states to discover the optimal policy. The Q-learning method uses Q-values $Q(s, a)$ 2.5 which represent the expected utility starting in s, taking action a, and thereafter acting optimally.

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma max_{a'} Q_k(s',a')) \tag{2.5}$$

To replace expectation by samples, the Q-learning method starts by considering the old estimate $Q_k(s,a)$, then it considers the new sample estimate and incorporates it into a running average:

$$Q_{k+1}(s,a) \leftarrow (1-\alpha)Q_k(s,a) + \alpha[target(s')] \tag{2.6}$$

With the ultimate goal of RL being to find the policy that maximizes the expected cumulative reward over time, by iteratively improving the policy or value function, the agent aims to converge to an optimal or near-optimal solution.

To finalize, MDPs provide a mathematical framework that enables RL algorithms to learn and make decisions in complex, uncertain environments. They facilitate the optimization of long-term rewards by considering the dynamics of the system and the trade-off between immediate and future rewards. RL algorithms utilize MDPs to model, solve, and optimize sequential decision-making problems in a wide range of domains, including robotics, game playing, autonomous systems, and recommendation systems.

2.1.2 Machine Learning Model

The approach used in [1] uses a supervised learning algorithm with training data from all patients except one (Leave-One-Subject-Out) for testing to predict the quality of motion or compute the score of being correct on a performance component, $P_{PM} = P(Y=1|X)$, where X refers a feature vector of an exercise motion and $Y \in [0,1]$ describes the correctness on a performance component of a motion. In this approach it is used several supervised learning algorithms such as Decision Trees (DT), Linear Regression (LR), Support Vector Machine (SVM), a Neural Network (NN), and a Long Short Term Memory (LSTM) network using the 'Scikit-learn' and the 'PyTorch' libraries.

Decision Trees

A Decision Tree (DT) is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

In this approach it was used **Classification and Regression Trees** (CART) [6] which is a decision tree learning algorithm designed to build binary trees by repeatedly dividing the training data set into smaller subsets based on the value of one of the input features to build pruned trees for DT. This improves the trees' generalization abilities and helps avoiding overfitting so it doesn't perform worse on new, unseen data.

Pruned decision trees pursue a balance between accuracy and complexity by removing unnecessary branches or nodes that do not provide the power to classify instances. This results in more generalized

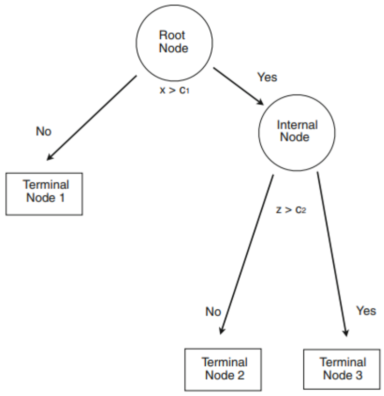

Figure 2.2: A simple CART tree structure.

models that can better handle unseen data, making them more reliable for making predictions or classifications in real-world scenarios.

By using this approach the objective is to predict the value of a target variable and to use the DT model's output to predict the quality of motion or compute the score of being correct on a performance component.

Linear Regression

Linear Regression [7] is a supervised learning algorithm that models the relationship between a dependent variable and one or more independent variables by finding the best fit linear equation to the observed data.

To find the best-fitting line, linear regression uses a technique called ordinary least squares (OLS), which minimizes the sum of the squared differences between the predicted and actual values. The OLS method calculates the optimal values for the coefficients by solving a system of equations.

In OLS, the linear equation is represented as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{2.7}$$

where y is the dependednt variable to be predicted, β_n are the parameters to be estimated, x_n are the independent variables and ϵ is the error term representing the deviation between the predicted and actual values. By minimizing the sum of squared errors (2.8), OLS will find the best-fitting line:

$$\sum (y - \hat{y})^2 \tag{2.8}$$

In this approach, it was applied **L1, L2 regularization** or a linear combination of L1 and L2 (ElasticNet with 0.5 ratio) to Linear Regression (LR) models. L1 regularization, also known as Lasso regression (Least Absolute Shrinkage and Selection Operator), shrinks the less important feature's coefficient to zero thus, removing it all together enableing a better feature selection when there is a huge number of features while L2 regularization, or Ridge regression, adds "squared magnitude" of coefficient as penalty term to the loss function. In equations (2.9) and (2.10) are representations of both L1 and

L2 regulariazation elements in a cost function, respectively, with the key difference between each the penalty term .

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.9}$$

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.10}$$

In summary, this LR model uses techniques that provide methods to control the complexity of a model and prevent overfitting by adding penalty terms to the loss fucntion based on the absolute values (L1) or squared values (L2) of the coefficients, so it improves the model's generalization and performance.

Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm used for both classification and regression tasks. It is a powerful and versatile algorithm that aims to find an optimal hyperplane or decision boundary that best separates the data points of different classes [8].

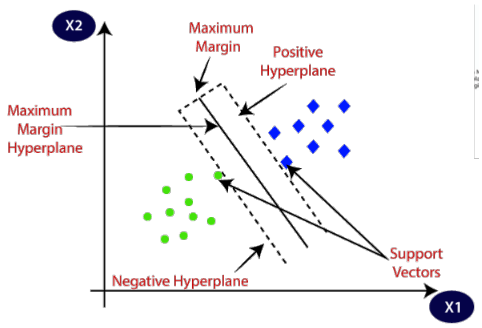

Figure 2.3: Illustration of how a SVM algorithm functions.

In classification tasks, SVM seeks to find a hyperplane that maximally separates the classes by using the closest data points of each class, thereby creating a clear margin between them. These closest data points are called support vectors, hence the name "Support Vector Machine.". In regression tasks, SVM aims to find a hyperplane that best fits the data points while allowing a certain amount of tolerance or error. The goal is to minimize the error or maximize the margin within the specified tolerance.

In this approach, it was applied Support Vector Machine (SVM) models with either linear, polynomial or Radial Basis Function (RBF) kernels and a penalty parameter C = 1.0.

With a **linear kernel**, such as dot product, the model intends to find a linear decision boundary that separates the classes in the feature space and uses the C parameter to control the penalty for misclassified data points. In the approach used it is chosen a C=1.0 which is considered as a middle of the range value as a smaller C value allows for missclassification, resulting in a larger margin, and a larger C value enforces strict classification leading to a smaller margin.

Polynomial kernels uses its function to map input data into a higher-dimensional space, where a linear decision boundary is found, by computing the the similarity between two data points based on the polynomial degree and coefficient parameters. The Cparameter wworks here in similar fashion to the linear SVM, where it determines the trade-off between margin width and classification errors.

To finalize this section, the **RBF SVM** measures the similarity between data points based on their radial distance from a reference point, as its kernel allows for non-linear decision boundaries, making it suitable for capturing complex relationships in the data. As for the C parameter, it works with the smae interpretation as in the linear and polynomial SVMs.

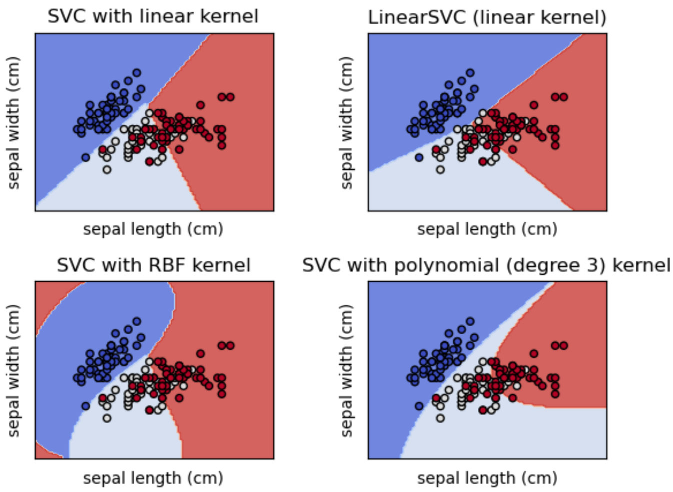

Figure 2.4: Example of how the SVM algorithm approaches classification tasks with each type of kernel.

Overall, SVM has several advantages, such as the ability to handle high-dimensional data, good generalization performance, and resistance to overfitting however, SVM's computational complexity increases with the number of data points, and it may be sensitive to the choice of hyperparameters. Additionally, interpreting the SVM model and understanding the significance of individual features can be challenging compared to some other algorithms.

Neural Networks

Neural networks, also known as artificial neural networks (ANNs)[9] or simply neural nets, are a class of machine learning models inspired by the structure and functioning of biological neural networks in the human brain. They are powerful and flexible algorithms capable of learning complex patterns and relationships from data.

A neural network consists of interconnected nodes, called neurons or artificial neurons, organized into layers. The three main types of layers in a neural network are the input layer, hidden layers, and output layer. The input layer receives the input data, which is then propagated through the hidden layers, where the actual computation and learning take place. The output layer provides the final prediction or output of the neural network.

Each neuron in a neural network receives inputs from the previous layer, applies a transformation or activation function to the inputs, and generates an output. The output of each neuron is then passed as inputs to the neurons in the next layer. The connections between neurons are assigned weights, which determine the strength and importance of each connection.

Neural networks can be applied to various machine learning tasks, such as classification, regression, and clustering. They excel at tasks involving complex patterns, non-linear relationships, and large amounts of data. With the availability of deep neural networks, which have multiple hidden layers, known as deep learning, neural networks have achieved remarkable success in areas such as computer vision, natural language processing, speech recognition, and many other domains.

One notable characteristic of neural networks is their ability to automatically extract features and representations from raw data, reducing the need for manual feature engineering. This makes them well-suited for tasks where the underlying patterns are not easily discernible or require a high level of abstraction.

During the training process, the neural network adjusts the weights based on the input data and the desired output. This adjustment is done using a learning algorithm, often a variant of gradient descent, which minimizes the difference between the predicted output and the true output. By iteratively adjusting the weights, the neural network gradually improves its ability to make accurate predictions or classifications.

In this approach, it is used Neural Networks (NN) models to predict the quality of motion or compute the score of being correct on a performance component. NNs are trained while grid-searching over various architectures (i.e. one to three layers with 32, 64, 128, 256, 512 hidden units) and different learning rates (i.e. 0.0001, 0.005, 0.001, 0.01, 0.1).

Grid search [10] is a systematic approach to hyperparameter tuning that helps identify the optimal combination of hyperparameters for a given model. It explores different architectural choices and learning rates to find the configuration that maximizes the model's performance on the validation set. By considering a wide range of possibilities, grid search helps to uncover the best hyperparameters for a particular task and dataset. For this task the evaluation was based on the F1-score metric.

Long Short Term Memory

Long Short-Term Memory (LSTM) [11] is a type of recurrent neural network (RNN) architecture that is designed to effectively capture and model long-term dependencies and patterns in sequential data. It was specifically developed to address the limitations of traditional RNNs, which struggle with learning and retaining information over long sequences.

LSTM introduces memory cells, which are capable of selectively retaining and forgetting information based on the input and the internal dynamics of the network. The memory cells are composed of different components, including input gates, forget gates, and output gates. These gates control the flow of information within the LSTM unit, allowing it to learn and remember relevant information over time.

The key feature of LSTM is its ability to store and access information for extended periods, enabling it to capture long-term dependencies in sequential data. The input gate determines how much new information is incorporated into the memory cell, while the forget gate determines how much of the existing memory is discarded. The output gate regulates the information that is passed on to the next time step or used as the final output of the LSTM unit.

The LSTM architecture is particularly useful in tasks involving sequential data, such as natural language processing, speech recognition, machine translation, and time series analysis. It has shown great success in these domains, surpassing the performance of traditional RNNs.

One advantage of LSTM is its ability to handle the vanishing gradient problem, which is a common issue in training deep neural networks. The design of LSTM allows it to propagate error signals over long sequences without significant degradation, making it easier to train deep recurrent networks.

In the approach [1], it is used Long Short Term Memory (LSTM) models in the form of two different architectures: many-to-one and many-to-many. In both architectures, 0.5 dropout is applied to LSTM layers, and various fully connected layers are explored for LSTMs (one to three layers with 32, 64, 128, 256, or 512 hidden units) with different learning rates.

The many-to-one architecture (2.5) is used for the "ROM" (Range of Motion) and "Smoothness" performance components. This architecture leverages sequential kinematic features to assess performance components at the end of an exercise.

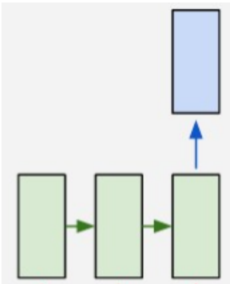

Figure 2.5: Many-to-one architecture

The many-to-many architecture (2.6) is used for the "Compensation" performance component. This architecture utilizes kinematic features at every frame of an exercise for frame-level assessment. In other words, it takes in a sequence of kinematic features as input and outputs a sequence of values that represent the quality of motion at each frame of an exercise. Figure 3b in the paper shows a diagram of this architecture.

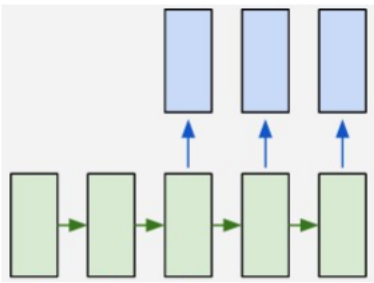

Figure 2.6: Many-to-many architecture

Overall, these two architectures are used to leverage sequential kinematic features and achieve personalized assessment during upper-limb rehabilitation exercises using LSTMs.

2.2 Related Work

The related work section of the paper provides a comprehensive overview of the research on socially assistive robotics in healthcare services, with several studies that demonstrate the potential of robotic exercise coaching systems to supplement rehabilitation processes.

Socially assistive robotics refers to the use of robots to provide assistance and support to individuals in a social context. This can include tasks such as providing feedback, coaching, or companionship. For instance [12],[13], studies that deployed a robotic exercise coaching system in a rehabilitation process to automatically monitor rehabilitation exercises and provide subjects feedback without the presence of a therapist. This studies found that such systems can be effective in providing feedback to subjects and monitoring their progress, with early results already demonstrating the promises of socially assistive robotics, a new interdisciplinary research area with large horizons of fascinating and much needed research. Even as socially assistive robotic technology is still in its early stages of development, the next decade promises systems that will be used in hospitals, schools, and homes in therapeutic programs that monitor, encourage, and assist their users. This is an important time in the development of the field, when the board technical community and the beneficiary populations must work together to shape the field toward its intended impact on improved human quality of life.

It is discussed how socially assistive robotics has shown great potential to supplement healthcare services through social interaction [14]. For example, a robotic exercise coaching system can be deployed in a rehabilitation process to automatically monitor rehabilitation exercises and provide subjects feedback without the presence of a therapist.

Fasola and Mataric's study should be highlighted, which compared elderly people's engagement with physically embodied robots versus virtually embodied agents as exercise partners. The study found that elderly people considered physically embodied robots more engaging and acceptable as exercise partners than virtually embodied agents.

Furthermore, there is research showing that diverse populations, including post-stroke patients, elderly people, and children, can successfully engage in exercise sessions with robotic exercise coaching systems. This suggests that such systems have broad applicability across different populations.

Also, regarding the ethics of the integration of social robots not only in rehabilitation but in healthcare as a whole, there was a study made [15] in which the goal was mainly to produce a commentary that is useful in the field of research without neglecting other aspects such as bioethics and economics. The study concluded that some potential ethical concerns include issues related to privacy, autonomy, and the potential for social isolation. Also, for example, the patient may prefer human interaction for emotional support. It also notes that social robots have the potential to be a reliable mediator/facilitator between humans in the field of rehabilitation and assistance as they can provide consistent care, reduce the workload of human caregivers, and potentially improve patient outcomes.

Most approaches done previously rely on a pre-defined motion or a generic threshold, which might not be applicable for patients with various characteristics. By focusing on personalized assessment, integrating machine learning and rule-based models, using LSTMs, and evaluating complex performance metrics, this approach is more suitable for use in a clinical setting and more effective at providing personalized feedback to patients during rehabilitation exercises.

2.3 Study for Stroke Rehabilitation

To automatically assess a patient's rehabilitation exercise and generate personalized corrective feedback it was presented an interactive approach that combines machine learning and rule-based models [1]. For this there was a study conducted on a dataset of three stroke rehabilitation exercises from 15 post-stroke subjects [16].

For each subject, it was recorded videos of the subjects performing the exercises both with and without compensation motion:

- **Exercise 1- Drinking water:** in this exercise, the patient is instructed to raise their wrist to their mouth as if they were drinking water. This exercise helps to improve wrist and forearm movement, which is important for performing tasks such as holding a cup or utensil.
- **Exercise 2- Light switch:** in this exercise, the patient is instructed to pretend to touch a light switch on the wall. This exercise helps to improve shoulder and elbow movement, which is important for performing tasks such as reaching for objects on a high shelf.
- **Exercise 3- Cane usage:** in this exercise, the patient is instructed to extend their elbow while seated, as if they were using a cane for support while walking. This exercise helps to improve elbow movement and stability, which is important for performing tasks such as carrying objects or pushing a wheelchair.

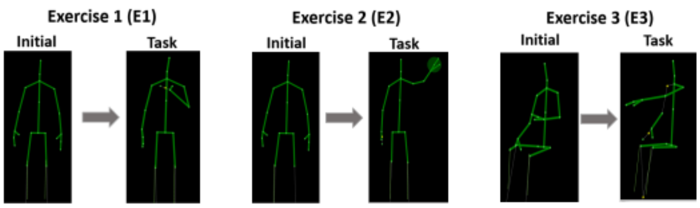

Figure 2.7: Exercise 1 (E1): 'Bring a Cup to the Mouth'; Exercise 2 (E2): 'Switch a Light On'; Exercise 3 (E3): 'Move a Cane Forward'

[1]

This exercises were then evaluated based on three common performance components of stroke rehabilitation exercises: Range of Motion (ROM), Smoothness, and Compensation [17].

Range of motion refers to how closely a patient performs the target position of a task-oriented exercise. In other words, it measures how well the patient is able to move their limbs through the full range of motion required for the exercise. This component is important because it helps to ensure that patients are performing the exercise correctly and effectively.

Smoothness refers to the degree of trembling and irregular movement of joints while performing an exercise. In other words, it measures how smoothly the patient is able to move their limbs during the exercise. This component is important because it helps to ensure that patients are not compensating for weakness or impairment by using other muscles or joints.

Compensation refers to whether a patient performs any compensated movements to achieve a target movement during an exercise. In other words, it measures whether the patient is using alternative movements or strategies to complete the exercise due to weakness or impairment in certain muscles or joints. This component is important because it helps therapists identify areas where patients may need additional support or intervention.

In addition, it was compared the approach to two other methods: a threshold-based method and an SVM-based method without ensemble voting and it was revealled that this approach outperformed both methods in terms of accuracy and F1 score.

Chapter 3

State of the Art

This chapter describes the state of the art of personalized interaction models of SARs for monitoring and guiding rehabilitation exercises by providing an overview of stroke rehabilitation and describing the key concepts of SARs. It will also be discussed the machine learning techniques used as well as Markov decision processes (MDP) and reinforcement learning (RL) ending up by highlighting the current state of the art in this field and what future directions this research might go into.

3.1 Current state of SARs

Socially Assistive Robots is a growing field that aims to develop robotic systems that can assist people in performing rehabilitation exercises and monitoring their progress.

Its systems are designed to monitor and track user progress by adjusting exercise programs based on performance and provide feedback to users and healthcare providers. SARs are also designed to provide social and emotional support to users, in order to improve the user's motivation and also adherence to the exercise programs, reducing the machine to person barrier.

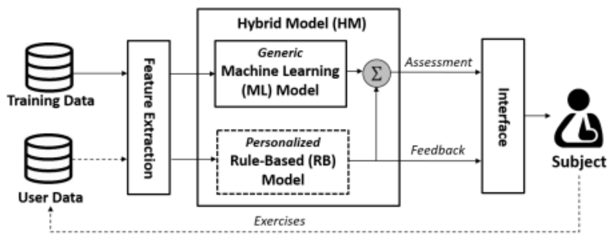

Figure 3.1: Flow diagram of an interactive approach of a socially assistive robot for personalized physical therapy.

[1]

Due to the ability of SARs being tailored to the individual users's needs and preferences, which makes them a promising approach for personalized rehabilitation and exercise programs, they have been tested in various clinical environments, such as spinal cord injury, Parkinson and stroke rehabilitation, showing results in terms of performance with improved outcomes and affordability by reducing healthcare costs.

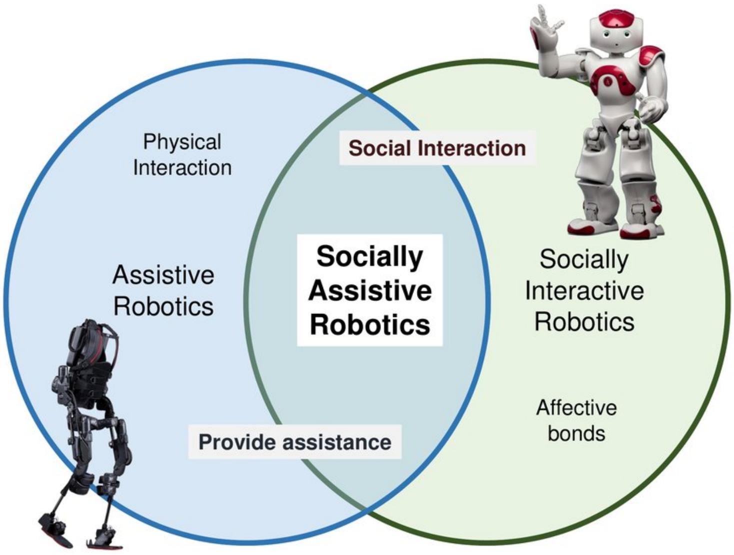

Figure 3.2: Defining Socially Assistive Robotics.
[18]

Nowadays the research that is being done in this field is mostly focused on developing better systems that are more intelligent and advanced in terms of adapting to user needs instantaneously and providing the most effective support specifically to each patient, with the biggest challenges regarding the subject of SARs including designing SAR systems that are affordable, easy to use, and accessible to a wide range of users, as well as addressing ethical and privacy concerns related to the use of robotic systems in healthcare [13].

3.2 Personalized Interaction models in SAR

3.2.1 Feature Extraction

The feature extraction process involves capturing skeletal data using a Kinect v2 sensor (Microsoft, Redmond, USA) and extracting various kinematic features from this data to model classifiers of performance components.

According to [19] the kinematic features extracted and utilized come from an exercise trial that will be explained below but, firstly, to describe the kinematic features there are some brief notations that need to be included so the process of extraction is better understood: F, sampling frequency (30Hz); t, time index; j, joint index; p_t, joint position; x, y, z, cartesian coordinates; ja, joint angle feature; rt, relative trajectory feature; pt, projected trajectory feature; sp, speed feature; ac, acceleration feature; jk, jerk feature. As well there are superscirpts such as max for maximum and avg for average.

ROM

For the **ROM**, it was computed the ja_t(3.1), this is the angle among three joints for a series of movements such as flexion and extension of the arms for example, and the normalized rt_t (3.2), nrt_t, this being how far a certain joint is moved away from the initial position applying the Euclidean distance, for example for the distance between the head and wrist joints. To compute the joint angle between the three joints it is also computed the projected trajectory P_t, which computes the projected trajectory between each joint's coordinates.

$$\begin{aligned} ja_t(j1,j2,j3) &= arcos\left(\frac{P_t(j1,j2)_t(j2,j3)}{|P_t(j1,j2)_t(j2,j3)|}\right) \\ P_t(j1,j2) &= (p_t(j1,x)-p_t(j2,x))+(p_t(j1,y)-p_t(j2,y))+(p_t(j1,z)-p_t(j2,z)) \end{aligned} \tag{3.1}$$

$$rt_t(b,s) = \sqrt{\sum_{c\in C}(p_t(b,c)-p_t(s,c))^2} \tag{3.2}$$

The normalization of the relative trajectory (3.3) is done by defining several normalized features to compensate individual's physical variability and can be used to do such things as segment a starting and ending frames of an exercise.

$$nrt_t(b,s) = \frac{|rt_t2(b,s)-rt_1(b,s)|}{rt_1(b,s)} \tag{3.3}$$

For this approach, the ja_t features will be destined to compute the joint angles of shoulder and elbow joints, the nrt_t features focus on measuring the distance of elbow and wrist joints regarding the head joint, and also the normalized projected trajectories features, npt_t (3.4), that will compute the distance of wrist with respect to the head and shoulder joints in x, y, z coordinates.

$$npt_t(j1,j2,c) = \frac{|pt_t(j1,j2,c)-pt_1(j1,j2,c)|}{pt_1(j1,j2,c} \tag{3.4}$$

Smoothness

As for "**Smoothness**", according to [1], the features are computed on wrist and elbow joints and are only regarding speed features (i.e. speed (sp_t(3.5)), acceleration (ac_t(3.6)), jerk (jk_t(3.7)), Mean Arrest Period Ratio (MAPR)($mapr_t$(3.10)), zero-crossing ratios (zc_t(3.11)), and normalized speed and jerk ($nsp_t(3.8)$ and njk_t(3.9)). As it is showned in the equations below, regarding the speed, acceleration and jerk features, all can be computed using the relative trajectory feature.

$$\begin{aligned} sp_t(j) &= F\times(rt_t(b,j)), \quad if \quad t>1. \\ &= 0 \quad otherwise \end{aligned} \tag{3.5}$$

$$ac_t(j) = F \times (sp_t(j) - sp_{(t-1)}(j)), \quad if \quad t > 1. \\ = 0 \quad otherwise \tag{3.6}$$

$$jk_t(j) = F \times (ac_t(j) - ac_{(t-1)}(j)), \quad if \quad t > 1. \\ = 0 \quad otherwise \tag{3.7}$$

In regards to the normalization of the speed and jerk features, it is utilized an average and maximum of speed or jerk until a selected frame to compute it and it is the division of an average speed/jerk by the maximum.

$$nsp_t(j) = \frac{sp_t^{avg}(j)}{sp_t^{max}(j)} \tag{3.8}$$

$$njk_t(j) = \frac{jk_t^{avg}(j)}{jk_t^{max}(j)} \tag{3.9}$$

The MAPR represents the portion of frames when speed exceeds 10% of the sp_t^{max}. It is computed by calculating the speed of each joint at each time frame using the Euclidean distance between consecutive joint positions, then setting a threshold for speed, determine the number of frames where the speed exceeds it and divide the count by the total number of frames. It is expected that subjects with limited functional abilities will have more stationary movements and higher values of MAPR than health subjects.

$$mapr_t(f_t, j) = \frac{1}{t} \sum_{s=1}^{t} \mathbb{I}_A(ft_s(j)), \\ A = \{ft_s(j) > ft_t^{max}(j) \times 0.1\}, ft_s(j) \in \{sp_s(j), jk_s(j)\} \tag{3.10}$$

To finalize this section, the zero-crossing ratio is used to quantify the frequency of sign changes in the acceleration or jerk of a subject's movement during an exercise trial. To compute zc_t first it is calculated the acceleration or jerk of each joint at each time frame using the difference between consecutive velocities or accelerations. Then it is counted he number of times that the sign of acceleration or jerk changes from positive to negative or vice versa to end by dividing this count by the total number of frames. Post-stroke survivors with limited functional abilities are expected to have higher zero-crossing ratio of acceleration or jerk than healthy subjects.

$$zc_t(f_t, j) = \frac{1}{t-1} \sum_{s=2}^{t} \mathbb{I}_{\mathbb{R}<0}(ft_s(j) ft_{(s-1)}(j)), \\ ft_s(j) \in \{ac_s(j), jk_s(j)\} \quad for\, t > 1 \tag{3.11}$$

Compensation

For this component, it is calculated the distances between joint positions of head, spine, shoulder in x, y, z axis from the initial to current frame by computing the ja_t (3.1) and the $dpt_t(|pt_t|)$.

With the main objective of being able to identify compensated movements, the joint angle features calculate the tilted angle of a spine, the elevated angle of a shoulder, and shoulder abduction angle, and the projected trajectories features measure the distance between the initial and current joint positions of head, spine, and shoulder joints in x, y, z coordinates.

Summarized Feature Vector and Matrix

Before the computation of the summarized feature vector and matrix, to reduce noise of acquiring joint positions from a kinect sensor similar to [19] it was applied a moving average filter with the window size of five frames (size of 2 seconds with a 50% overlap between adjacent windows).

The summarized feature matrix **F** $= \{f_1, ..., f_T\} \in R^{T\times d}$, given an exercise motion, with T number of frames and d features and statistics (e.g. maximum, minimum, range, average, and standard deviation) is computed of a feature matrix over all frames of the exercise to summarize a motion into a summarized feature vector, $X \in R^{5d}$. This summarized feature vector was utilized for the assessment on **ROM** and **Smoothness** performance components. Additionally, a feature matrix was utilized for frame-level assessment on **Compensation** performance component.

This approach allowed to capture temporal information about the exercise motion and extract features that reflect its dynamics.

3.3 Rehabilitation and exercise monitoring

Socially assistive robots (SAR) have shown potential for delivering rehabilitation and exercise interventions that are both effective and engaging for patients. Personalization is a critical component of SAR interventions, as it can help to tailor the intervention to the individual needs, preferences, and abilities of each patient. To achieve this, personalized interaction models have been developed that allow SAR to adapt their behavior to the specific characteristics of the patient they are interacting with.

There are many types of personalized interaction models in SAR for rehabilitation and monitorization of exercises, such as sensor-based models, machine learning models, cognitive models, rule-based models and user-based models.

For this specific case the models used are **machine learning** and **rule-based** models to automatically have an assessment of the patient's rehabilitation exercise and give corrective feedback based of their movement. These models will be explained with more detail later.

3.3.1 Rule-Based Model, RB

A rule-based model is a type of artificial intelligence model that, when presented with an input or new scenario, applies a set of explicitly defined rules to generate an output or make a decision or prediction.

It is mostly used where the decision-making process can be easily codified, such as in diagnostic, expert and decision support systems as its simplicity and ease to be interpreted make it very usefol for classification, pattern recognition and knowledge representation tasks.

In [1], the RB model is used as a starting point to the automated assesssment capability of a robotic exercise coaching system and to provide feedback feedback to the patient in real-time to inicate wether they are doing the exercise in the correct way or not during the session.

The RB model is not very effective for tasks that require complex decision-making or learning from data, hence it is paired with machine-learning forming an hybrid approach where the ML model is used to automatically learn insightson a large amount of data and augment the RB model with new insights on data from machine learning models. This makes the SAR able to do multivariate analysis on a patient's exercise while also having the transparency and personalized interaction of a robotic exercise coaching system.

Initial Development

For the initial development there were maid semi-structured interviews with two therapists (ϕ = 5.0, δ = 1.05 years of experience in post-stroke rehabilitation) to set the rules based on their expert knowledge, by providing feedback on the correct execution of each exercise as well as specifying the range of acceptable values of each kinematic parameter. This knowledge is formalized as 15 independent $if-then$ rules and can be specified as in [20]:

$$\hat{Y} = \begin{cases} 1 & if\, p^{max}(wr, c_y) \geq p^{max}(spsh.c_y) \\ 0 & else \end{cases} \tag{3.12}$$

where $\hat{Y}$ denotes the predicted label on a performance component. This rule simply checks the maximum position of a wrist joint, $p^{max}(wr, c_y)$, in comparison to that of a spine shoulder joint, $p^{max}(spsh, c_y)$, in the y-coordinate of c in the set $C \in c_x, c_y, c_z$ to roughly estimate whether a patient achieves the target position of Exercise 1.

For the prediction with multiple rules, the paper applies a majority voting algorithm and does not apply any tie-breaking method given an odd number of rules. This means that if there are multiple rules that apply to a given exercise, the model will predict the outcome based on the majority of those rules. If there is an even number of rules and there is a tie, then the model will randomly choose one of the tied outcomes.

Also in [1] it is presented the performance of the initial rule-based model (RB-Init) from interviews with therapists and that of the fine-tuned rule-based model (RB-tuned) after accommodating held-out user's unaffected motions to tune threshold values for personalized assessment. The parameters of rule-based models (i.e., the range of the threshold value with 2 or 3) are selected to achieve the best F1-score during validation: 3 is utilized over three performance components of three exercises except for the 'ROM' and 'Smoothness' of both Exercise 1 and 2.

In addition, the RB model also has its rules set based on clinical guidelines that can be found in existing literature on post-stroke rehabilitation and physical therapy.

Performance Score

The performance score (P_{RB}) is computed based on the predicted label on each performance component using the rule-based model and can be computed with the equation below:

$$P_{RB} = \frac{1}{|\mathbb{R}|} \sum_{r \in R} min(\frac{x_r}{\tau_r}, 1) \tag{3.13}$$

The min function is used so the equation assigns the value of 1 if the feature value x_r of a rule r exceeds the threshold τ_r of that r and takes into account the expert knowledge from the therapists, with $\mathbb{R}$ describing the set of rules prompted by them. If the min function were not used in the equation for computing the score of being correct, then it would assign a value of 0 to any feature value that does not exceed the threshold of a rule. This means that even if a patient's performance is close to the target range specified by r, it would still be considered incorrect if it does not exceed the threshold value.

Tunning the RB model

As the initial threshold values of rules are generic, the RB model can be fine-tuned by updating τ_r of the initial rule-based model with held-out patient's unaffected motion to learn a Gaussian probability density function $f(x_r) \sim N(\mu_r, \sigma_r^2)$, with μ_r and σ_r being the mean and the standard deviation of a feature value x_r, and then updating the threshold value with either $2\sigma_r$ or $3\sigma_r$.

This new RB-tuned showed significant improvements in performance by increasing its ability to replicate the therapist's assessment by around 37% from 0.5821 to 0.7957 average F1-scores over all exercises ($p < 0.01$).

3.3.2 Hybrid Model, HM

The hybrid model [21] combines the strengths of both models, the data-driven, machine learning (ML) model and the rule-based (RB) model from therapists, to achieve better performance than either one alone. As the RB model is interpretable, it also provides an opportunity for clinicians to analyze rules and personalize rehabilitation assessment.

To fuse the predictions of the two models, the hybrid model applies a weighted average, ensemble technique [22] [23] where the performance of each model is used as the weight of each model, more specifically the F1-scores of each model in the range of $[0, 1]$. The hybrid model computes the score of being correct, $P_{HM} = P(Y = 1|X)$ as follows:

$$P_{HM} = \frac{\rho_{ml}}{\rho_{ml} + \rho_{rb}} P_{ML} + \frac{\rho_{rb}}{\rho_{ml} + \rho_{rb}} P_{RB} \tag{3.14}$$

In equation (3.14), P_{ML} and P_{RB} are the scores of being correct for the machine learning and rule-based model, respectively, and ρ_{ml} and ρ_{rb} describe the F1-scores of the models.

3.3.3 Ensemble Voting Method for Frame-Level Assessment

For this approach the goal of this method is to provide a more robust assessment of compensation motion by using predictions on multiple consecutive frames, V_f. To reach this goal it is utilized the initial continuous frame-level predictions and it is computed the scores based on preditions made on multiple consecutive frames using an SVM model with RBF kernel.

In the first step of the ensemble voting method it is used a support vector machine (SVM) with radial basis function (RBF) kernel to classify each frame as either compensatory or non-compensatory, with $h(f_t)$ being the predicted frame-level assessment at t frame with an assessment model h and feature vector f_t.

In the second step, as soon as there are V_f initial frame-level predictions available, a score is computed for each frame based on the predictions made by the SVM model on multiple consecutive frames, taking into account the predictions made on the current and previous four frames to compute a score for the current frame. To compute the score, the SVM model outputs a probability value,$Y_t \in \gamma$, indicating the likelihood that the frame is compensatory, then the probability values for the current and previous four frames are used to compute an average probability value, $\hat{Y}_t$, on a compensation motion with the largest number of the predictions. To finalize the score computation, if the average probability value exceeds a threshold value, then the current frame is classified as compensatory, otherwise, it is classified as non-compensatory. The winning prediction at frame t is selected as follows:

$$\hat{Y}_t = \underset{Y \in \gamma}{argmax} \sum_{f_t in \bar{F}} \delta(h(f_t), Y) \tag{3.15}$$

where $\bar{F}$ indicates a set of accumulated V_f feature vectors until t frame and $\delta(h(f_t), Y)$ counts the predicted assessment of Y with the predictions from V_f frames by assigning 1 if $h(f_t) = Y$ and 0 otherwise. To finalize this subject, in case of tied votes, this method assigns $\hat{Y}_t$ with the latest prediction $h(f_t)$, and to provide a more robust frame-level assessment it leverages votes from past V_f - 1 frames to the current t frame.

Chapter 4

Conclusion and Future Work

Finally, this last chapter describes the conclusions derived from the prepared work, the objectives and the plan of the future work to be developed.

4.1 Conclusions

It was presented an interactive approach for a robotic exercise coaching system that integrates a data-driven machine learning model with an interpretable rule-based model and tunes with patient's data for transparent, personalized interaction and it was proposed an ensemble voting method to improve frame-level assessment during upper-limb rehabilitation exercises.

The approach uses long short term memory networks (LSTMs) to measure agreement with ground truth labels during upper-limb rehabilitation exercises and it was evaluated how well it can monitor other complex performance metrics of an exercise, such as smoothness or the occurrence of a compensation motion.

The results showed that the interactive approach achieved good agreement with expert's annotation, demonstrating the feasibility of using this approach for personalized assessment during rehabilitation exercises, by achieving high accuracy in assessing exercise performance and being able to provide personalized feedback to patients based on their unique characteristics and needs.

Using an interactive approach to tune a model with patient's motions for personalized assessment during rehabilitation exercises is crucial as this approach has the potential to be used in a clinical setting, where it can help improve the quality of care for stroke patients undergoing rehabilitation.

Overall, the paper presents a promising approach for automatically assessing exercise performance and providing personalized feedback to patients during rehabilitation exercises.

4.2 Future Work and Work plan

The work done throughout this semester raised some important questions, not only related to SARs but to the field of robotics in itself, like:

- Can the approach be used for other types of rehabilitation exercises, or is it specific to upper-limb rehabilitation exercises?
- How well does the approach presented generalize to new patients and exercises?
- What are the limitations of the approach presented, and how can they be addressed in future work?
- How can the approach presented be integrated into existing clinical workflows, and what are the potential benefits and challenges of doing so?
- What are the ethical considerations associated with using machine learning to assess exercise performance and provide personalized feedback to patients during rehabilitation exercises?

The future work will focus on improving the interpretability of the rule-based model and exploring other machine learning approaches to further improve performance.

Future work could explore how well the approach generalizes to other types of exercises and patients with different conditions. Also, to address some limitations, future work could evaluate the approach on larger and more diverse datasets to assess its generalizability to other exercises and patient populations, explore alternative sensing modalities that could be used in place of motion capture data, such as wearable sensors or video analysis, incorporate additional factors that may affect exercise performance into the model, such as fatigue or pain and develop real-time feedback mechanisms that can provide immediate feedback during exercise performance to improve coaching effectiveness.

For a better understanding of the expected work to be elaborated, Figure 4.1 illustrates the respective Gantt chart.

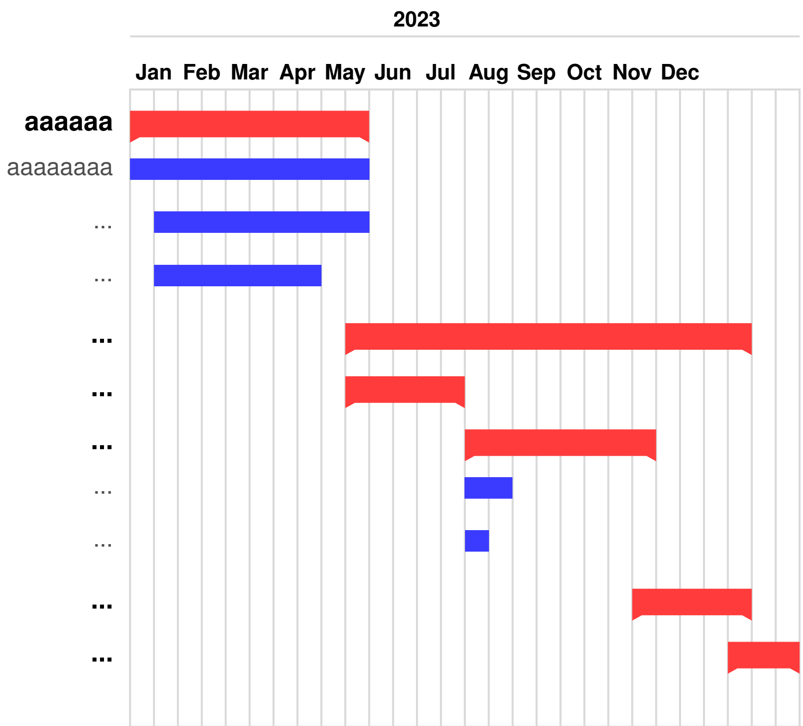

Figure 4.1: Timeline of the expected work.

Bibliography

[1] M. H. Lee, D. P. Siewiorek, A. Smailagic, A. Bernardino, and S. B. Badia. Towards personalized interaction and corrective feedback of a socially assistive robot for post-stroke rehabilitation therapy. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1366–1373, 2020. doi: 10.1109/RO-MAN47096.2020.9223462.

[2] W.-Y. Kuo, C.-Y. Chen, M.-C. Chen, C.-M. Wang, Y.-L. Lin, and J. Wang. Can rehabilitation adherence among stroke patients be measured using a single item? *Journal of Clinical Nursing*, 32(5-6): 950–962, 2023.

[3] Y. Li. Deep reinforcement learning, 2018.

[4] M. L. Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[5] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[6] R. A. Berk. Classification and regression trees (cart). *Statistical learning from a regression perspective*, pages 1–65, 2008.

[7] X. Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[8] A. C. Lorena and A. C. P. L. F. de Carvalho. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, 14(2):43–67, Dec. 2007. doi: 10.22456/2175-2745.5690. URL `https://seer.ufrgs.br/index.php/rita/article/view/rita_v14_n2_p43-67`.

[9] C. M. Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[10] Y. Bao and Z. Liu. A fast grid search method in support vector regression forecasting time series. In *Intelligent Data Engineering and Automated Learning–IDEAL 2006: 7th International Conference, Burgos, Spain, September 20-23, 2006. Proceedings 7*, pages 504–511. Springer, 2006.

[11] K. Smagulova and A. P. James. A survey on lstm memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228(10):2313–2324, 2019.

[12] A. Tapus, M. J. Mataric, and B. Scassellati. Socially assistive robotics [grand challenges of robotics]. *IEEE Robotics Automation Magazine*, 14(1):35–42, 2007. doi: 10.1109/MRA.2007.339605.

[13] M. J. Matarić, J. Eriksson, D. J. Feil-Seifer, and C. J. Winstein. Socially assistive robotics for post-stroke rehabilitation. *Journal of neuroengineering and rehabilitation*, 4:1–9, 2007.

[14] J. Fasola and M. J. Matarić. A socially assistive robot exercise coach for the elderly. *J. Hum.-Robot Interact.*, 2(2):3–32, jun 2013. doi: 10.5898/JHRI.2.2.Fasola. URL `https://doi.org/10.5898/JHRI.2.2.Fasola`.

[15] D. Giansanti. The social robot in rehabilitation and assistance: What is the future? *Healthcare*, 9(3), 2021. ISSN 2227-9032. doi: 10.3390/healthcare9030244. URL `https://www.mdpi.com/2227-9032/9/3/244`.

[16] M. H. Lee, D. P. Siewiorek, A. Smailagic, A. Bernardino, et al. Opportunities of a machine learning-based decision support system for stroke rehabilitation assessment. *arXiv preprint arXiv:2002.12261*, 2020.

[17] J. Sanford, J. Moreland, L. R. Swanson, P. W. Stratford, and C. Gowland. Reliability of the fugl-meyer assessment for testing motor performance in patients following stroke. *Physical therapy*, 73 (7):447–454, 1993.

[18] J. C. Pulido, C. Suárez, J. C. González Dorado, A. Dueñas-Ruiz, P. Ferri, M. Encarnación, M. Sahuquillo, C. Echevarría, R. Vargas, P. Infante-Cossio, C. Luis, P. Calderón, and F. Fernández. A socially assistive robotic platform for upper-limb rehabilitation: A longitudinal study with pediatric patients. *IEEE Robotics Automation Magazine*, 26:24–39, 06 2019. doi: 10.1109/MRA.2019.2905231.

[19] M. H. Lee, D. P. Siewiorek, A. Smailagic, A. Bernardino, and S. B. i. Badia. Learning to assess the quality of stroke rehabilitation exercises. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 218–228, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362726. doi: 10.1145/3301275.3302273. URL `https://doi.org/10.1145/3301275.3302273`.

[20] M. H. Lee, D. P. Siewiorek, A. Smailagic, A. Bernardino, and S. Bermúdez i Badia. An exploratory study on techniques for quantitative assessment of stroke rehabilitation exercises. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20, page 303–307, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368612. doi: 10.1145/3340631.3394872. URL `https://doi.org/10.1145/3340631.3394872`.

[21] M. H. Lee, D. P. Siewiorek, A. Smailagic, A. Bernardino, and S. Bermúdez i Badia. Interactive hybrid approach to combine machine and human intelligence for personalized rehabilitation assessment. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 160–169, 2020.

[22] M. H. Lee. Intelligent agent for assessing and guiding rehabilitation exercises. In *IJCAI*, pages 6444–6445, 2019.

[23] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.