Skeleton-Based AI-Driven Feedback Generation for Physical Rehabilitation via Social Assistive Robotics

Sergio Caño-Amor[1], Carmen Díaz-de-Mera[1,2], Fernando Fernandez[1,2], and Rodrigo Alarcón[2]

[1] Dpto. de Informática, Universidad Carlos III de Madrid, Av. de la Universidad 30, 28911 Leganés, Spain
[2] Inrobics Social Robotics, S.L.L., Av. Gregorio Peces Barba 1, 28919 Leganés, Spain

Abstract. The Inrobics Rehab platform allows users to exercise with a social assistive robot, be it for physical rehabilitation or just stimulation. The users follow the robot in a mirroring activity where they have to imitate the robot's movements, while the system checks that the user is correctly doing them by means of a 3D sensor. In this paper we consider the integration of a feedback generation system that helps users achieve the poses expected by the robot, so that it can behave fully autonomously. To generate this feedback we have considered the use of different Artificial Intelligence techniques, such as expert systems or machine learning models, including decision trees and neural networks. The problem has been modelled as a classification task, where metrics such as the accuracy, precision and f1 score have been used to evaluate the methods. Following this, we discovered that the best option according to these metrics is to choose the neural network using the explicit pose id. Despite of this we will find that using the random forest using the expected user angles, is not only faster, but also more stable to all classes.

Keywords: Social Assistive Robotics · Artificial Intelligence · Physical Rehabilitation

1 Introduction

Social Assistive Robotics (SAR) is a field of robotics that explores the use of robots to assist through social interaction [14]. A common application of Social Assistive Robotics (SAR) is exercise coaching, where a social robot showcases different movements and guides users in their execution. This application has been found to be useful in promoting and encouraging exercise in the elderly [7] and in boosting engagement and motivation in rehabilitation therapies, especially in pediatrics, such as cardiac rehabilitation [6], cerebral palsy rehabilitation [12] or post-stroke rehabilitation [13].

The Inrobics Rehab platform employs a social assistive robot as a co-therapist. The system consists on a social robot (Aldeberan's NAO robot), a 3D sensor (Orbbec's Persee+) and an application that professionals can use to configure rehabilitation sessions and monitor the results. The system, previously known as NAOTherapist [19], has been evaluated in rehabilitation therapies for children with cerebral palsy [17] and spinal cord injury [21]; and in an intensive therapy camp for children [18].

Among other activities and applications, the Inrobics Rehab platform can be used for physical stimulation and rehabilitation based on the repetitions of movements in a mirroring game, where the user imitates the robot. The robot will perform the movements and check that the user is doing the expected sequence of poses by means of the 3D sensor, from which the 3D coordinates of the skeleton of the patient can be gathered.

As of now, the robot waits for a determinate amount of time for the user to set the correct pose; if the waiting time is exceeded, the robot will proceed with the session. Sometimes this occurs not because the user is not doing the pose, but because they are not doing it exactly as the robot is expecting it. In these cases, what users need is specific feedback to help them to correctly set the pose. If users do not receive this feedback, they can get frustrated at the robot for "not working" or not "checking the poses correctly".

Feedback can be described as "the element that conveys the internal system status of a robot to a human" [15], it is a way for humans to understand the intentions, knowledge, and capabilities of the robot [11]. The Inrobics Rehab platform can benefit from a system that generates feedback while the robot is checking the users' poses; helping them achieve the movements correctly and avoiding frustration, while also making the robot more understandable.

The aim of this paper is to create a system to generate verbal feedback so that the robot can autonomously help users achieve the poses expected. For this purpose different Artificial Intelligence techniques will be explored, such as Expert Systems and Machine Learning models. In the following sections we describe how the data to create these models was gathered, which methods were used and their evaluation.

2 Related Work

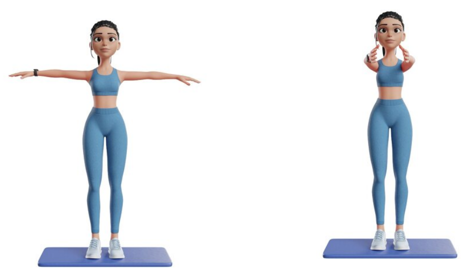
Research into the effect of different feedback models is extensive in Human-Robot Interaction. It has been shown that children's engagement is related to the robot's feedback during second language learning [1, 10], where positive feedback from the robot has a positive impact in their learning performance. During imitation games with autistic children, graded cued feedback was shown to improve their accuracy [9], while the type of feedback received by post-stroke rehabilitation patients impacted their task delay time [22].

Research focuses on different types of feedback, be it facial [5], auditory [20] or verbal [4]. Verbal feedback has been shown to be the most prominent type of feedback, although other modalities offer support in the interaction [16].

There is also research focused on understanding feedback in rehabilitation settings, with hierarchies and taxonomies, be it focusing on robotic rehabilitation systems [8] and digital feedback systems [2]; and in classical rehabilitation, understanding which feedback therapists favor during therapy [3]. In robotic rehabilitation systems, which is the case this work focuses on, research shows that feedback should focus on task success first, while the quality of the movement is a secondary objective [8].

3 Data gathering

In the Inrobics Rehab plaform a social robot asks users to set sequences of poses while checking that they are correctly performing them. The poses have been predefined by occupational therapists acting as clinical advisors, focusing on upper arm training. Each pose has an internal identifier, referenced in this paper as the pose id, used to differentiate them, as show in Fig. 1.



(a) Pose a21 (b) Pose a11

Fig. 1: Example of poses present in the system

Poses are constructions of different body angles in each of the articulations and movement planes of the body. Every articulation in our body has three different movement planes (Fig. 2). These are the sagittal, frontal and transverse planes. Following this, we will have different angle measurements for each articulation, depending on the movement plane we are taking into consideration.

Regarding this last part, and when talking about the angles and movements of our joints, it is important to be clear about what kind of motion we are referring to, since joints can move in different ways depending on the plane of movement. For example, the shoulder can move forward and backward in the sagittal plane – this is what we call flexion and extension. On the other hand, when the arm moves sideways away from or toward the body in the frontal plane, the movement is called abduction and adduction. Understanding these types of joint motions is essential, especially when giving feedback in rehabilitation or motor control tasks, as each direction involves different muscles and serves a different purpose.

Currently in our system we have a total of **18 different poses** that the user can perform and the system can recognize, and a total of **16 different types of**

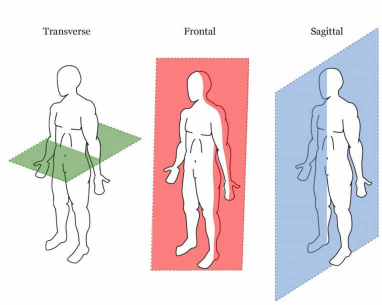


Fig. 2: The three human movement planes.

feedback, which have been defined with the help of clinical advisors. The poses that the user may perform, vary from a wide range, as they can contain: poses with both arms up and straightened or both arms touching the hips. Feedback may also come in different forms, as it could contain: feedback for crossing the right arm to the left or lowering our arms. This feedback provides information for the user so that he or she could adjust his body according to the expected pose.

The information that can be gathered to form a dataset includes mainly: the current angles of the pose that the user is not performing correctly, the expected pose and the different feedback that should be given in each case to train a supervised model. Each incorrect pose from the user can required several type of feedback. The expected pose can be represented in different ways, using only the pose id explicitly or the angles that represent the pose. As we focus in upper-limb training, the angles included are limited to the shoulder and the elbow, as the wrist is not relevant to the type of poses and feedback defined.

As we need labeled examples from incorrect poses with their corresponding feedback, we artificially created a datasest. Data regarding the angles of the body parts were collected using a 3D sensor, Orbbec's Persee+, that captured a depth image from the environment, from which the skeleton and corresponding 3D coordinates were extracted from the user using the external library NuiTrack. From these coordinates we calculated the different angles present in each articulation, both body parts of the upper and lower extremities (although, as mentioned, we will only use body parts of the upper extremities for this experiment).

We filmed different videos, placing our upper limbs in different positions for each specific pose and desired feedback. For example. With this technique, we collected different angles measurements for each different pose and different feedback for each pose, completing a total of almost 3.200 instances for our dataset. We managed to gather all these data, since each instance in the dataset corresponds to a frame in each of the videos for each pose and desired feedback.

Our problem is multilabel, in which each example can have several classes assigned, and has an unbalanced class distribution 3, in which there are more instances of some classes than others, as there are types of feedback which can be given in several different poses (such as: "raise your arms") and others which are mainly specific for a single pose (such as: "move your right arm to your left hip"). This aspect of the dataset is key for testing the different methods, as their performance should not decrease for the types of feedback with fewer examples.

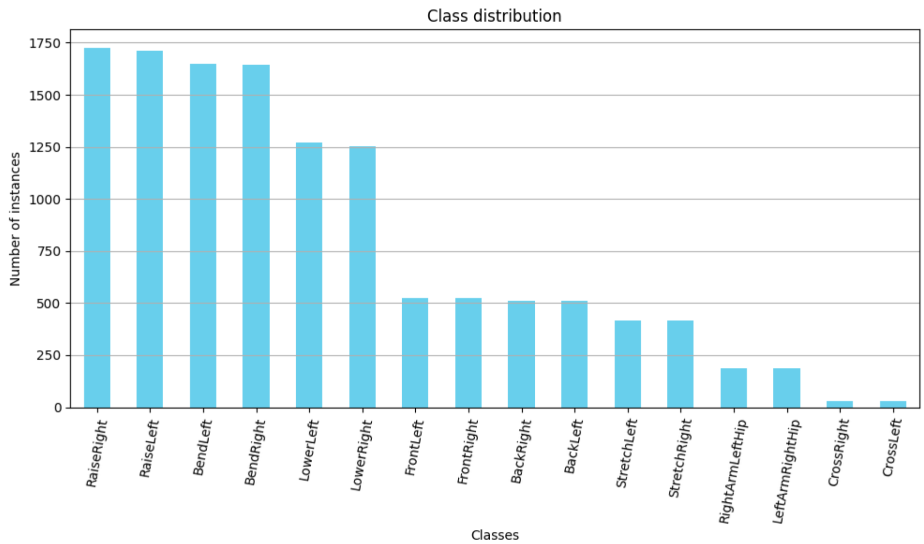


Fig. 3: Unbalanced class distribution for our problem.

4 Methods

The following section will explain and describe the different methods that have been followed to give feedback to the user according to the pose that the robot has asked for and the one that the user is currently doing.

In order to do this, several methods were developed. On the one hand, a rule-base method was proposed, focusing on an expert rule-system. On the other hand, several machine learning methods were also developed and compared, including neural networks (using two different approaches), a decision tree, and a random forest.

Appart from the previously mentioned taxonomy, we also divide our work according to the information we are including in the datasets used to train the model. Here, we will only find two divisions: a dataset containing the explicit pose id and a dataset containing, instead of the explicit pose id, the expected user angles. Both types of dataset contain the real-time user angles at the moment.

The main idea is to compare these different techniques and choose the best one taking into account the different results, metrics, difficulty, explainability, and time elapsed when inferring with them. The different methods implemented are the following:

4.1 Rule-Based Methods

Expert rule-system (using expected user angles): The first and only rule-based method that was implemented was a simple expert rule-system that managed to solve our problem and give feedback accordingly to the user. An expert rule-based system is an Artificial Intelligence (AI) approach that uses a set of predefined if-then rules to make decisions based on input data. These rules, created by domain experts, form a knowledge base that the system applies. An example of these types of rules would be:

if $goalfe\_shoulder\_right > userfe\_shoulder\_right$ **then**
```
    feedback.append("raise_arm");
```
end

Where, when taking into account shoulder flexo-extension, if the goal angle is greater that the actual user angle, the user will we given the feedback to raise the arm.

And, unlike machine learning, rule-based systems do not learn from data but follow fixed logic, offering transparency and control in decision-making.

Once the idea of an expert rule-based system is clarified, we will proceed to explain how ours was made. This method, as most in this paper, used the user angles approach to give feedback. In this method, firstly, all angles, movements and feedback types are encoded in different dictionaries that will be modified and accessed later. This approach mainly focuses on iterating all different angle movements, according to the different human movement planes, present in each of the upper-limb joints that the user is performing and compares them with the expected user angles for the corresponding pose asked by the robot. This comparison calculates the difference between this two angles and according to the angle and the plane the movement is being performed, it decides the feedback it is going to be given. It is important to note that this difference also includes the sign, as it helps determine the direction in which the movement should be performed. For example, in the case of shoulder flexion-extension, the sign of the value indicates whether the user should raise the arm or lower the arm.

This simple rule-based system allows for extremely fast execution, as it does not require any model training or complex computations. Its deterministic nature also ensures high precision in the outputs. This makes it particularly suitable for real-time applications where speed and consistency are critical.

4.2 Machine Learning Methods

Neural Network (using pose id): The first machine learning method that was implemented was a neural network that used the different angles that the 3D sensor was capturing from the user at a given time and the id of the desired pose (one of the 18 included in the application).

This first approach, (the one considering explicit poses, not the fact that we use Neural Networks) was considered the simplest and the most static one because of the id of the included pose. Including the pose id, makes the decision process easier as the system is learning to recognize poses according to their id instead of the angles the user currently has. It is also the most static because if a new pose is included in the future, the model will have to be re-trained to include that pose among the characteristics used to train the model.

Returning to our main concern, the neural network that was developed, was designed to solve a multi-label problem: giving multiple feedback to the user at a time. Following this idea, we built a neural network which was fine-tuned for our specific problem. The corresponding hyper-parameters where tuned and defined in an informal way (through experimentation) and the parametric configuration that was reached in the end, according to our results, performs really good when deployed. We also keep in mind that it may not be the most optimum configuration for the problem, according to how it was built.

The **hyper-parameters** that were **finetuned and experimented** with are the following:

1. **Model Architecture**
   (a) Input Layer: `shape = (X_train.shape[1],)`
   (b) Multi-Head Self-Attention:
      i. `head_size = X_train.shape[1]`
      ii. `num_heads = 32`
   (c) Normalization Layer:
      i. `epsilon = 1e-6`
   (d) Dense and Dropout Layers:
      i. `Dense(128, activation='relu')`
      ii. `Dropout(0.3)`
      iii. `Dense(64, activation='relu')`
      iv. `Dropout(0.3)`
   (e) Output Layer:
      i. `Dense(len(label_headers), activation='sigmoid')`
2. **Compilation**
   (a) Optimizer: `'adam'`
   (b) Loss Function: `'binary_crossentropy'`
   (c) Evaluation Metric: `'binary_accuracy'`
3. **Training**
   (a) Epochs: `100`

This neural network, had the following **layers**:

1. **An input layer**, with the same size as the number of features used in the problem
2. **A multi-head self attention layer**, as the attention (which calculates the importance of each part regarding the rest of the features) is calculated by several heads (32 in this case) and the input, which contains 3 entries (query, key, value) are all the same entry, the input layer.

3. **An adding layer**, which combines the input layer and the self-attention output, used to create a residual connection to facilitate the deep learning of the model.
4. **A normalization layer**, used to stabilize and accelerate the training phase.
5. **Dense and dropout layers**, which are used to first, model complex relationships (using the relu activation function) and then, deactivate some neurons stochastically to prevent the model from depending too much on some neurons.
6. **An output layer**, which gives an output vector (using a sigmoid activation function) as big as the number of labels to infer, fitting itself to our multilabel problem.

This neural network, was trained with data according to the upper-limb angles of the user at a given time and the pose id. It had several data related, for example, with the flexo-extension of each shoulder and the pose id.

This data was of course, filtered, cleaned and normalized to be later fed to the neural network. In the other hand, some of the different types of feedback that the neural network could return were: "lower your left arm" or "move your right arm backwards". Indicating the action that the user should perform in order to achieve the desired pose (making the desired and the user angles very similar). This data was encoded in one-hot encoding, where each instance of the data set had at least one of the different types of feedback active.

As mentioned before, this problem is multilabel, so it could have several feedback columns active. This happens because the robot (the one that this feedback system is made for), could give to the user several feedback at a time, i.e: "Raise your left hand and lower your right one".

Neural Network (using expected user angles): The next method that was implemented was the same neural network (architecturally speaking), but using the expected user upper-limb body angles instead of the pose id. This approach was followed because it was considered to be a much more dynamic and flexible approach, as the model, could be able to generalize more without fitting itself to the pose id. In this way, if in the future a new pose is added to the pose repertory, instead of re-training the whole model (the one based on the pose id), we could continue using the same one (the one with desired user angles), as the model will be fed with a different combination of angles and it will be capable of generalizing to a new pose not seen before.

In this way, the model was fed with the data previously shown except for the pose id column. The "Goal angles" were also added, that is, the same number of angle types that are recorded from the user, but with the desired user angles.

Decision Tree (using expected user angles): For the next method, we tried to develop another solution that could solve the problem in the most flexible way, that is, using the expected user angles, with a simpler and faster method. In order to do this, we chose to use a Decision Tree for the decision making, which, apart from being simpler and faster, could add explainability to the decision making area, allowing us to see in which joints the model was paying more attention to or which rules it decided to use.

Before anything else, we also have to take into account that we are solving a really unbalanced problem. This happens because, as multiple feedback can be given in each instance, it is much more common to give feedback as "raise both hands" more than "take your right hand to your left hip", which is only needed in very few poses.

So, in order to solve the fact that our data is not balanced, we used stratification when splitting between the training and test data set. Using this technique, we divide our data following the total data distribution, making sure no split is left without data from any feedback class. In this way, we are also replicating reality, as feedback such as "take your right hand to your left hip" is much more less likely to happen than "raise both hands" because of the number of poses each one is involved in.

After this, and when training the decision tree, we decided to make both a cross validation process and a grid search with the training data.

- Using **cross validation**, we divide the data into different number of folds, where one is going to be used for validation of the hyper-parameters of the model. This process is repeated taking as validation data each of the folds, one at a time, making sure the validation is performed with different parts of the data set. Using this, we make sure that there is no fold made in the split which is beneficial for the model.
- Using **grid search**, we create a grid of hyper-parameters (the options to tweak the model and change its behavior) that we are later going to be trying with all possible combinations between them and chose the ones that best results give to our problems.

And last but not least, as our problem is multi-label, a wrapper was used for this model, because a decision tree by itself does not support multiple labeling. This wrapper (MultiOutputClassifier from scikit learn) trained as much decision trees as prediction labels (all different types of feedback available) which learned by the one vs rest methodology. Using this, each tree learns to identify its corresponding feedback. Adding the output of each tree, we finally obtain our desired multi-label output.

Random Forest (using expected user angles): In order to try to improve the performance from our last method, a random forest model was developed. This makes sense, as the random forest can be considered as the evolution of the decision tree. The random forest method is, in fact, contained of several decision trees trained in a random way and aggregated to obtain a much more robust prediction. It introduces two main sources of randomness:

- **Bootstrapping:** Each tree is trained on a different bootstrap sample from the training set.

- **Random feature selection:** At each node, a random subset of features is selected, and the best split is determined only among this subset.

This, is combined again with the MultiOutputClassifier method to obtain a multi-label output. This method is also trained using both the cross validation and grid search techniques that we used with the decision tree before.

5 Evaluation

In order to analyze the performance of our different models, we need to find out how well they are doing infering the solutions for our multi-label problem.

All models were evaluated using the full dataset. For the tree-based models (Decision Tree and Random Forest), cross-validation was applied in order to find the best hyper-parameters for each model and built more general and robust performance metrics. The dataset was split into folds, and classification reports were generated for each fold, where we will find values for the precision, recall and F1-score for each of the target labels. From these, the mean and standard deviation of each metric were calculated to better reflect model performance across different data partitions. Table 2 summarizes the metrics obtained for each method.

It is important to mention that the low performance of the rule-based system can be attributed to the fact that it was not yet able to take into account the significance of each joint angle within a pose. Without this information, the system treats all angular deviations equally, which leads to a high number of false positives and therefore lowers its overall precision and accuracy. However, it is important to note that the recall values achieved by the rule-based system are considerably high. This suggests that the system is effective at detecting most of the necessary feedbacks, but lacks a proper filtering mechanism to discard unimportant variations.

Incorporating the angle significance will allow the system to focus only on meaningful deviations, reducing noise and improving overall performance. This enhancement is planned as part of future work.

6 Conclusions

After carefully evaluating all the proposed models using cross-validation and a detailed analysis of their performance metrics, we can conclude that the Random Forest model offers the best balance between accuracy, generalization, and computational efficiency. Although the neural network with attention and pose id input achieved slightly better results in some metrics (especially in terms of overall accuracy and average F1-score) the Random Forest model performed consistently well, with very high average accuracy and F1 and low standard deviation across folds, indicating strong reliability.

Table 1: Metrics for each of the methods evaluated

	Exact match accuracy	Precision (micro avg)	Recall (micro avg)	F1-score (micro avg)
Expert rule-based system	0.28	0.43	0.91	0.58
Neural Network (using pose id)	**0.89**	**0.95**	**0.98**	**0.97**
Neural Network (using user angles)	0.78	0.90	0.92	0.91
Decision Tree (using user angles)	0.73	0.90	0.94	0.92
Random Forest (using user angles)	0.87	**0.95**	0.97	0.96

Table 2: Prediction times for the evaluation set

	TIme (s)
Expert rule-based system	0.13
Neural Network (using pose id)	3.47
Neural Network (using user angles)	3.72
Decision Tree (using user angles)	0.88
Random Forest (using user angles)	0.7

Furthermore, the inference time of the Random Forest is significantly lower than that of the neural network, which is particularly relevant in real-time applications or environments with limited computational resources, such as interactive systems or assistive robots. Therefore, considering both performance and efficiency, the **Random Forest** stands out as the most balanced and recommendable option for our system, providing accurate and fast feedback to the user without compromising reliability.

References

1. Ahmad, M.I., Mubin, O., Shahid, S., Orlando, J.: Robot's adaptive emotional feedback sustains children's social engagement and promotes their vocabulary learning: a long-term child–robot interaction study. Adaptive Behavior **27**, 243–266 (5 2019). https://doi.org/10.1177/1059712319844182, https://journals.sagepub.com/doi/abs/10.1177/1059712319844182
2. Brennan, L., Zubiete, E.D., Caulfield, B.: Feedback design in targeted exercise digital biofeedback systems for home rehabilitation: A scoping review. Sensors 2020, Vol. 20, Page 181 **20**, 181 (12 2019). https://doi.org/10.3390/S20010181

3. Carr, K., Zachariah, N., Weir, P., McNevin, N.: An examination of feedback use in rehabilitation settings. Critical Reviews in Physical and Rehabilitation Medicine **23**, 147–160 (2011). https://doi.org/10.1615/CRITREVPHYSREHABILMED.V23.I1-4.110
4. Clair, A.S., Mataric, M.: How robot verbal feedback can improve team performance in human-robot task collaborations. ACM/IEEE International Conference on Human-Robot Interaction **2015-March**, 213–220 (3 2015). https://doi.org/10.1145/2696454.2696491, https://dl.acm.org/doi/10.1145/2696454.2696491
5. Cohen, L., Khoramshahi, M., Salesse, R.N., Bortolon, C., Słowiński, P., Zhai, C., Tsaneva-Atanasova, K., Bernardo, M.D., Capdevielle, D., Marin, L., Schmidt, R.C., Bardy, B.G., Billard, A., Raffard, S.: Influence of facial feedback during a cooperative human-robot task in schizophrenia. Scientific Reports 2017 7:1 **7**, 1–10 (11 2017). https://doi.org/10.1038/s41598-017-14773-3, https://www.nature.com/articles/s41598-017-14773-3
6. Céspedes, N., Irfan, B., Senft, E., Cifuentes, C.A., Gutierrez, L.F., Rincon-Roncancio, M., Belpaeme, T., Múnera, M.: A socially assistive robot for long-term cardiac rehabilitation in the real world. Frontiers in Neurorobotics **Volume 15 - 2021** (2021). https://doi.org/10.3389/fnbot.2021.633248
7. Fasola, J., Matarić, M.J.: A socially assistive robot exercise coach for the elderly. J. Hum.-Robot Interact. **2**(2), 3–32 (Jun 2013). https://doi.org/10.5898/JHRI.2.2.Fasola
8. Fruchter, D., Polak, R.F., Berman, S., Levy-Tzedek, S.: Hierarchy in algorithm-based feedback to patients working with a robotic rehabilitation system: Toward user-experience optimization. IEEE Transactions on Human-Machine Systems **52**, 907–917 (10 2022). https://doi.org/10.1109/THMS.2022.3170831
9. Greczek, J., Kaszubski, E., Atrash, A., Matarić, M.: Graded cueing feedback in robot-mediated imitation practice for children with autism spectrum disorders. In: The 23rd IEEE International Symposium on Robot and Human Interactive Communication. pp. 561–566 (2014). https://doi.org/10.1109/ROMAN.2014.6926312
10. de Haas, M., Vogt, P., Krahmer, E.: The effects of feedback on children's engagement and learning outcomes in robot-assisted second language learning. Frontiers in Robotics and AI **7**, 101 (8 2020). https://doi.org/10.3389/FROBT.2020.00101/BIBTEX
11. Hellström, T., Bensch, S.: Understandable robots. Paladyn **9**, 110–123 (2 2018). https://doi.org/10.1515/PJBR-2018-0009/MACHINEREADABLECITATION/RIS, https://www.degruyterbrill.com/document/doi/10.1515/pjbr-2018-0009/html
12. Malik, N.A., Hanapiah, F.A., Rahman, R.A.A., Yussof, H.: Emergence of socially assistive robotics in rehabilitation for children with cerebral palsy: a review. International Journal of Advanced Robotic Systems **13**(3), 135 (2016)
13. Matarić, M.J., Eriksson, J., Feil-Seifer, D.J., Winstein, C.J.: Socially assistive robotics for post-stroke rehabilitation. Journal of neuroengineering and rehabilitation **4**, 1–9 (2007)
14. Matarić, M.J., Scassellati, B.: Socially Assistive Robotics, pp. 1973–1994. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1_73
15. Mirnig, N., Gonsior, B., Sosnowski, S., Landsiedel, C., Wollherr, D., Weiss, A., Tscheligi, M.: Feedback guidelines for multimodal human-robot interaction: How should a robot give feedback when asking for directions? In: 2012 IEEE RO-MAN:

The 21st IEEE International Symposium on Robot and Human Interactive Communication. pp. 533–538 (2012). https://doi.org/10.1109/ROMAN.2012.6343806
16. Mirnig, N., Gonsior, B., Sosnowski, S., Landsiedel, C., Wollherr, D., Weiss, A., Tscheligi, M.: Feedback guidelines for multimodal human-robot interaction: How should a robot give feedback when asking for directions? Proceedings - IEEE International Workshop on Robot and Human Interactive Communication pp. 533–538 (2012). https://doi.org/10.1109/ROMAN.2012.6343806
17. Pulido, J.C., Suarez-Mejias, C., Gonzalez, J.C., Ruiz, A.D., Ferri, P.F., Sahuquillo, M.E.M., Vargas, C.E.R.D., Infante-Cossio, P., Calderon, C.L.P., Fernandez, F.: A socially assistive robotic platform for upper-limb rehabilitation: A longitudinal study with pediatric patients. IEEE Robotics and Automation Magazine **26**, 24–39 (6 2019). https://doi.org/10.1109/MRA.2019.2905231
18. Pulido, J.C., Fuentetaja, R., García, E., García, M., Abuín, V., González, J.C., Iglesias, A., Fernández, F.: A gamified social robotics platform for intensive therapies in neurorehabilitation. Intelligent Service Robotics **17**, 419–443 (5 2024). https://doi.org/10.1007/S11370-024-00521-W/FIGURES/9, https://link.springer.com/article/10.1007/s11370-024-00521-w
19. Pulido, J.C., González, J.C., Suárez-Mejías, C., Bandera, A., Bustos, P., Fernández, F.: Evaluating the child–robot interaction of the naotherapist platform in pediatric rehabilitation. International Journal of Social Robotics **9**, 343–358 (2017). https://doi.org/10.1007/s12369-017-0402-2
20. Rosati, G., Rodà, A., Avanzini, F., Masiero, S.: On the role of auditory feedback in robot-assisted movement training after stroke: Review of the literature. Computational Intelligence and Neuroscience **2013**, 586138 (1 2013). https://doi.org/10.1155/2013/586138, https://onlinelibrary.wiley.com/doi/full/10.1155/2013/586138
21. Salas-Monedero, M., Cereijo-Herranz, V., DelosReyes-Guzmán, A., Pérez-Borrego, Y., Gil-Agudo, A., García-Martín, F., Pulido-Pascual, J.C., López-Dolado, E.: Smoothness and efficiency metrics behavior after an upper extremity training with robic humanoid robot in paediatric spinal cord injured patients. Applied Sciences 2023, Vol. 13, Page 4979 **13**, 4979 (4 2023). https://doi.org/10.3390/APP13084979, https://www.mdpi.com/2076-3417/13/8/4979/htm https://www.mdpi.com/2076-3417/13/8/4979
22. Swift-Spong, K., Short, E., Wade, E., Matarić, M.J.: Effects of comparative feedback from a socially assistive robot on self-efficacy in post-stroke rehabilitation. In: 2015 IEEE International Conference on Rehabilitation Robotics (ICORR). pp. 764–769 (2015). https://doi.org/10.1109/ICORR.2015.7281294