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

WAF 2025

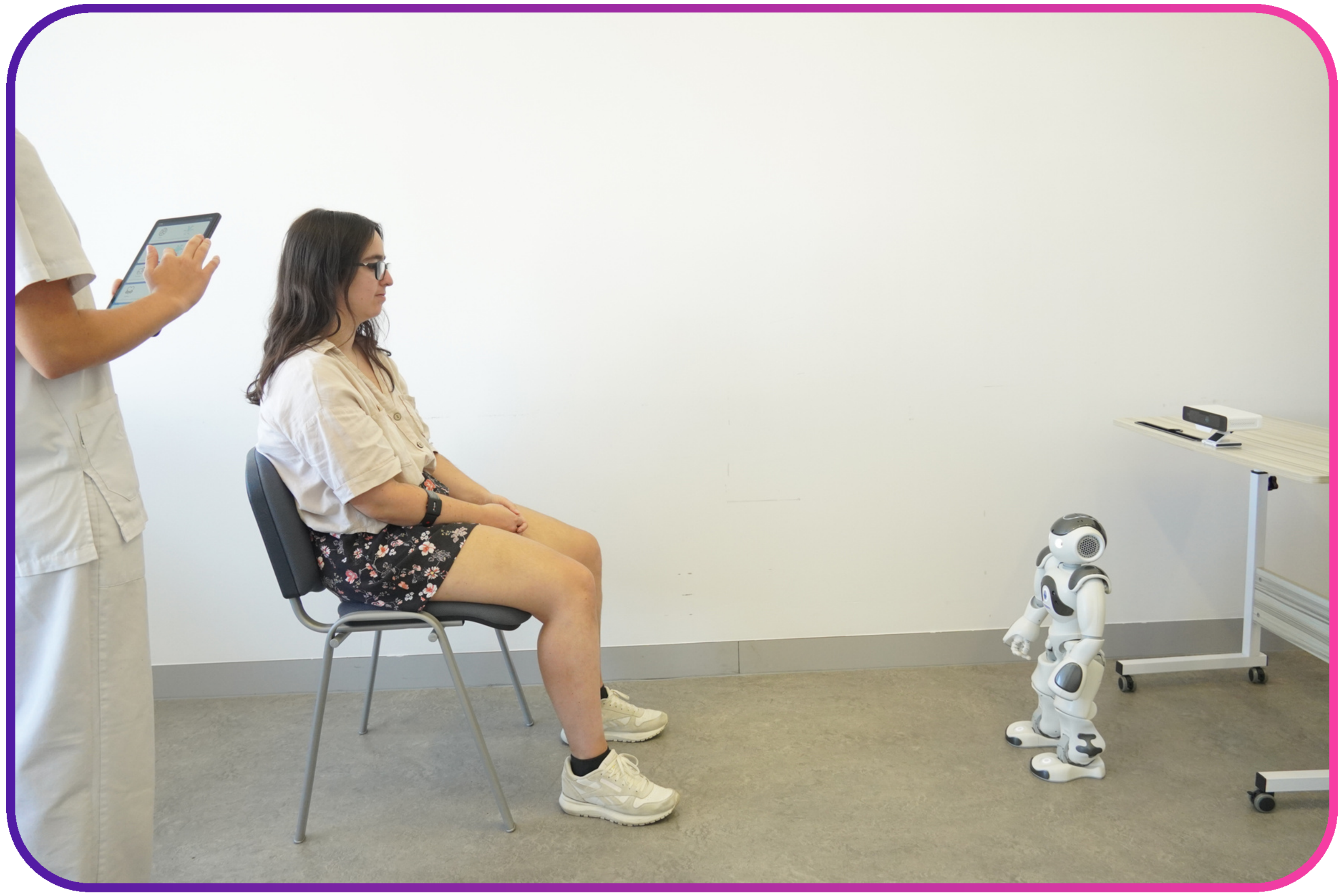
Sergio Caño Amor – Universidad Carlos III de Madrid
Carmen Díaz de Mera – Inrobics Social Robotics
Fernando Fernandez Rebollo – Universidad Carlos III de Madrid
Rodrigo Alarcón – Inrobics Social Robotics

Index

Introduction I

INROBICS REHAB PLATFORM

- **Social Assistive Robotics** for therapy and rehabilitation
- Components: **NAO robot** (co-therapist) + **3D sensor** (Orbbec Persee+) + **professional app.**
- **Tested** in therapies for children with cerebral palsy, spinal cord injury, and intensive therapy camps.

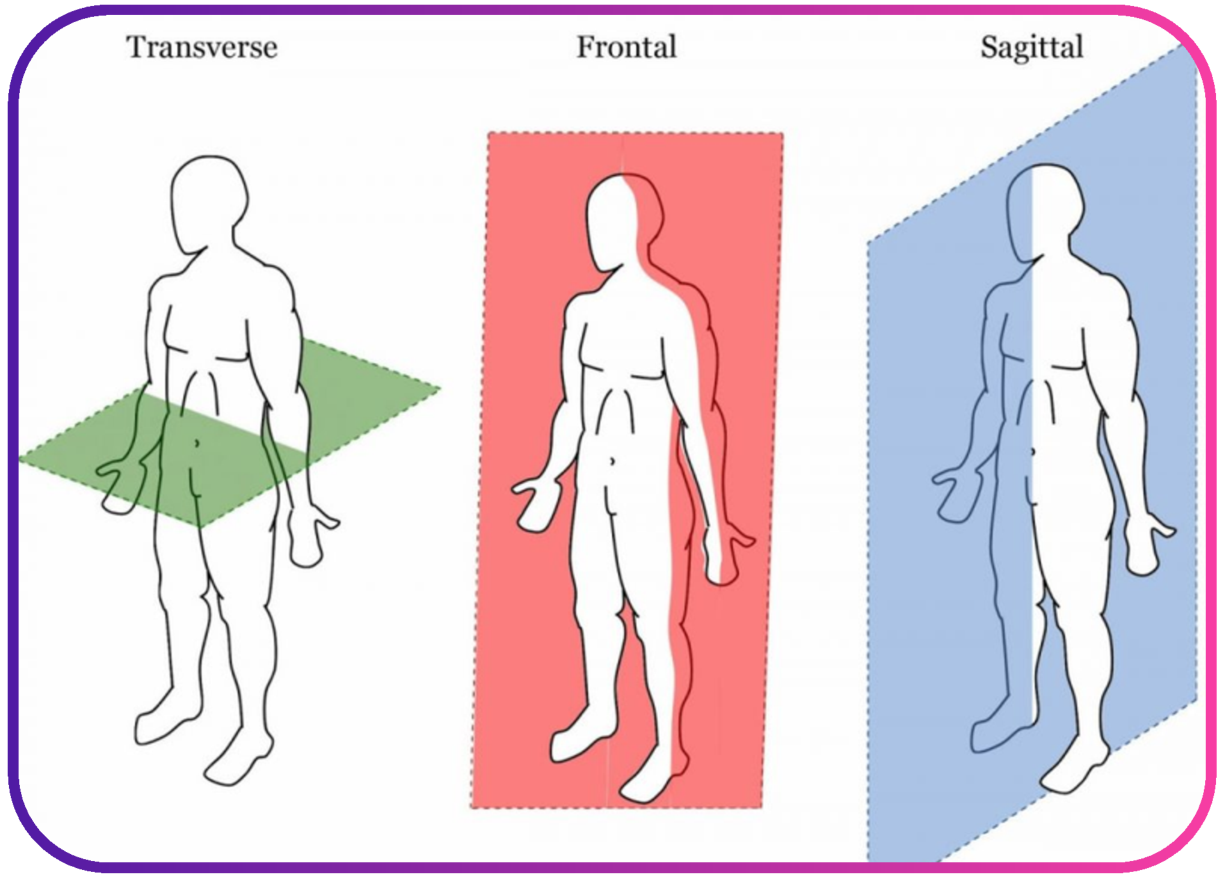


Introduction II

↘

REHABILITATION METHOD

- Focus on a **mirroring game** where patients imitate the robot's movements.
- 3D sensor tracks patient's skeleton and pose sequence.
- Robot waits for correct pose and gives encouraging feedback



AIM OF THE WORK

- Explore AI techniques (Expert Systems, Machine Learning) to Develop a system to generate **autonomous verbal feedback** to guide users to achieve the correct poses

Data Gathering I

↘

DATA

- Data focused on **upper limb movement.**
- Dataset is formed by: **18** different **poses** and **16** different types of **feedback** (“Lower your arms” or “Cross right arm to the left”).
- Poses can be represented by a **pose id** or a **set of angles**

DATA COLLECTION PROCESS

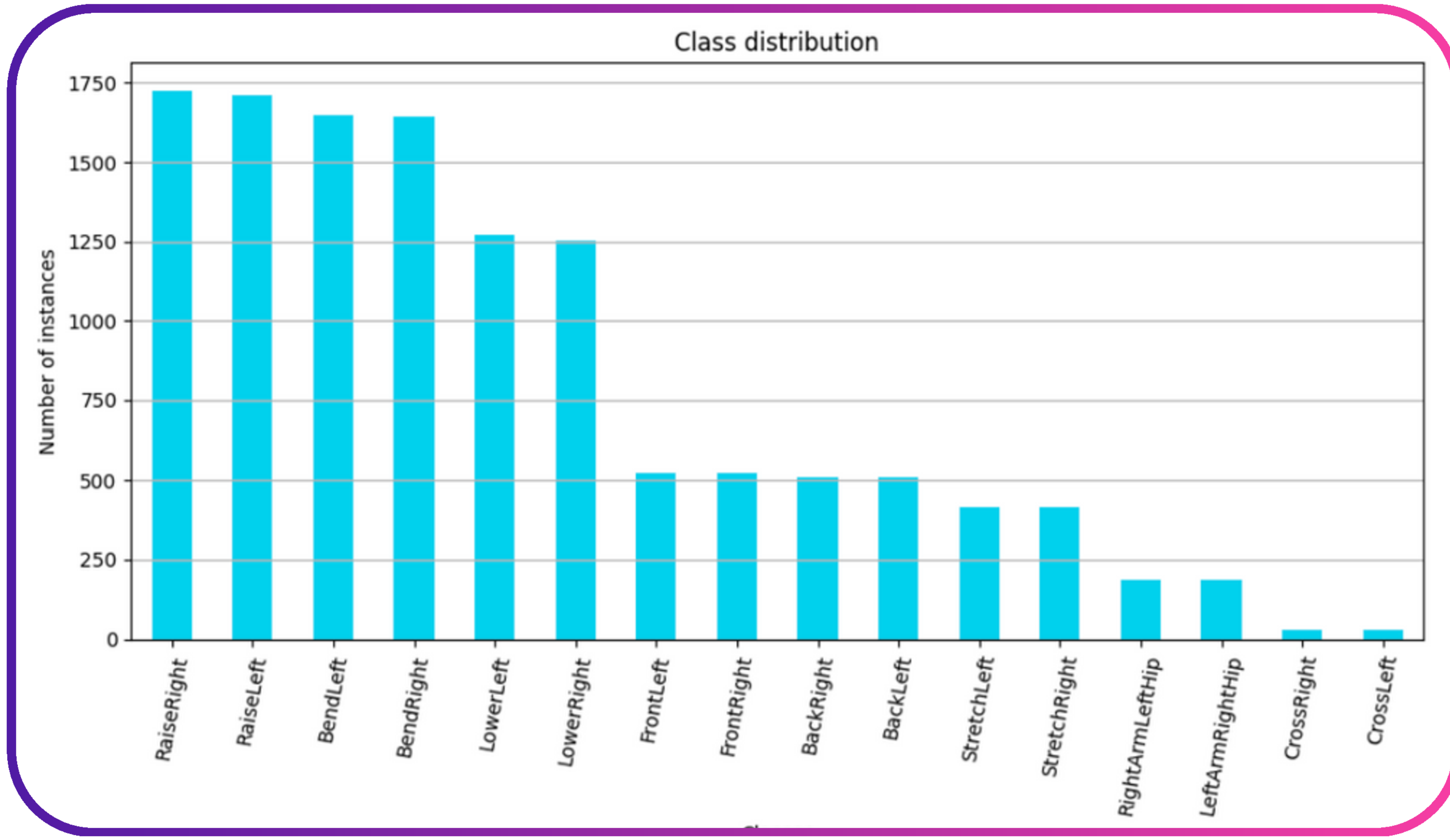
- 3D sensor (Orbbec Persee+) with **NuiTrack** to extract skeleton + 3D coordinates.
- **Videos recorded** of upper limb positions for each pose + feedback.
- Generated **dataset** of ~3,200 instances (frames from videos).

Data Gathering II

↘

DATASET CHARACTERISTICS

- **Multilabel problem:** one pose can require multiple feedback types.
- **Unbalanced distribution**
- Common feedback VS Specific feedback
- Key challenge: ensure models perform well even with rare feedback types.

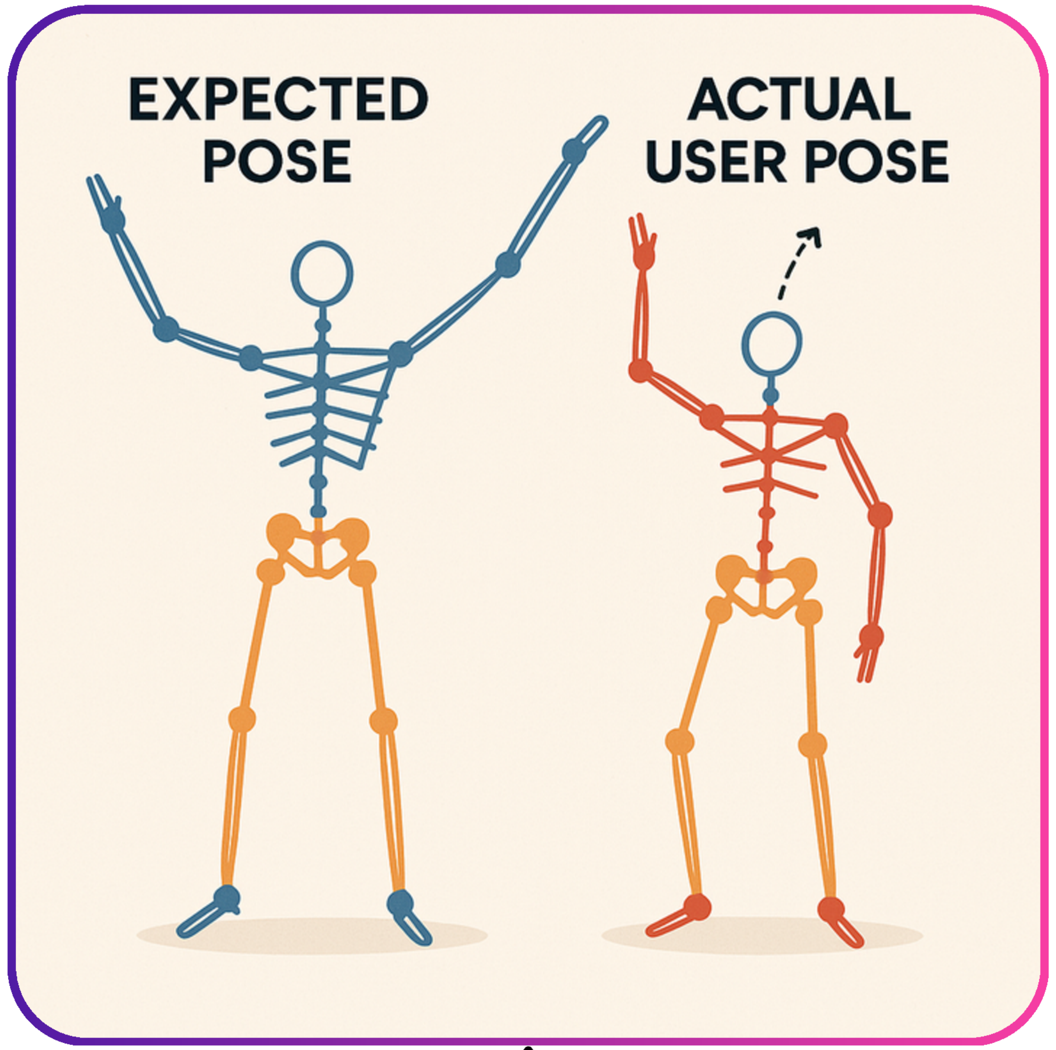


Methods

↘

APPROACH

- Multi-label Classification problem
- **Methods divided into:**
  - Rule-based system (expert-defined).
  - Machine Learning models (Neural Networks, Decision Tree, Random Forest).
- **Two dataset setups:**
  - With pose ID.
  - With expected pose angles.



Methods: Expert system

↘

APPROACH

- Expert rule-system comparing user vs. expected angles.
  - Example: if GoalLeftShoulderFEAngle > UserLeftShoulderFEAngle → feedback = "raise left arm".
- **Advantages:**
  - Fast, explainable, deterministic.
  - Ideal for real-time applications.
- **Limitations:**
  - Not adaptive, requires manual rules.

Methods: Neural Networks

↘

APPROACH (POSE ID)

- Input: user angles + pose ID.
- **Pros:** good performance.
- **Cons:** static, must retrain if new poses added.

APPROACH (USER ANGLES)

- Input: user angles + goal angles (no pose ID).
- **Pros:** dynamic & generalizable to new poses.
- Same architecture

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

Methods: Decision Tree

↘

APPROACH

- Input: user angles + expected angles
- **Advantages:**
  - Explainable
  - Fast and simple.
- **Challenges:**
  - Unbalanced data
- Uses MultiOutputClassifier for multilabel support, which trains a decision tree for each label.

Methods: Random Forest

↘

APPROACH

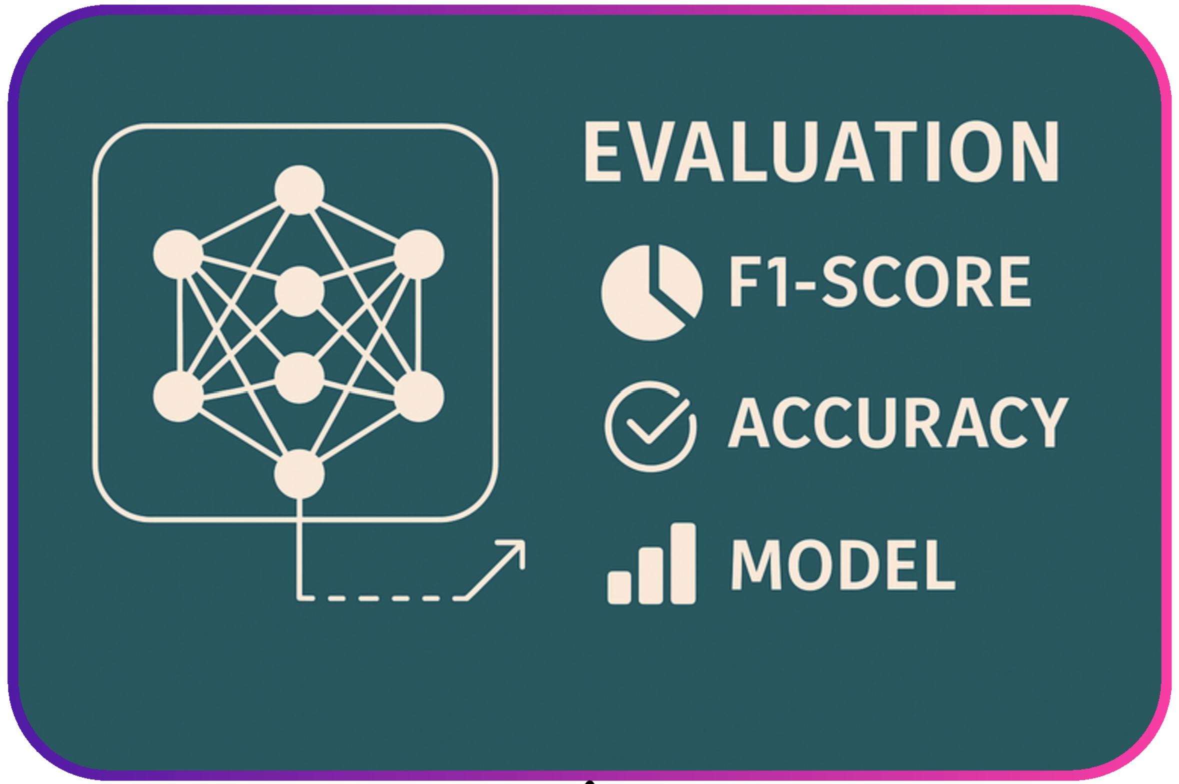
- Ensemble of decision trees with randomness:
  - Bootstrapping (different samples per tree).
  - Random feature selection at each split.
- **More robust predictions** than a single tree.

Evaluation I

↘

EVALUATION SETUP

- **Goal:** assess performance of models on our classification problem.
- Metrics: Accuracy, Precision, Recall, F1-score.
- Tree-based models (Decision Tree, Random Forest):
  - Used cross-validation to optimize hyper-parameters.
  - Mean & standard deviation calculated across folds.



Evaluation II

↘

MODEL COMPARISON

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

Evaluation III

↘

MODEL COMPARISON

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)	0.92

Conclusions

BEST CHOICE: RANDOM FOREST

- High average accuracy & F1-score.
- Low standard deviation → reliable performance.
- **Faster inference than Neural Networks** → suitable for real-time assistive robotics.
- **Most balanced and practical option** for generating feedback in rehabilitation tasks.

Future work

↘

SPECIAL ISSUE WORK

- Integrate **angle difference** data (expected user angles - actual user angles)
- **Leave-one-out** technique (to see if the model is capable of generalizing)
- **Incremental** training (see how many poses the model can already generalize)

International Journal of Advanced Robotic Systems

Thank you for your attention!

Any questions?