AN2DL - First Challenge Report
TeamSevilla

María Rivas Jiménez, Matteo Orsenigo, Aurea Maria Jimenez Garcia, Alessandro Tobia Cavalieri

Codabench mararivasjimnez, Codabench matteoorsenigo, Codabench aureamariajimenez, Codabench AleCava01v2

295768, 251682, 295734, 306833

November 19, 2025

1 Introduction

This project focuses on time series classification using artificial neural networks.
The goal is to **predict the real pain level of each subject** on the basis of their time-series motion data, which include sensor estimations of pain levels, subject characteristics and measurements of body joint angles.

2 Problem Analysis

The dataset contains 40 features. Each record is uniquely identified by a sample_index, which refers to one of 661 patients. For every patient, 160 time measurements were collected, resulting in a total of 105,760 records.

The columns n_legs, n_hands, and n_eyes are categorical and therefore must be converted into numerical form. All other columns are of type float64, which can be safely cast to float32 for more efficient memory usage. Additionally, the column joint_30 is constant across all samples. The dataset is also imbalanced: the "no_pain" class occurs far more frequently than the other two classes. As a result, the model is likely to be biased toward predicting this majority class.

3 Method

For the preprocessing stage, we first observed that joint_30 feature was constantly equal to 0.5 across all samples. Since this feature carried no useful variability, we removed it from the dataset. Next, we examined the distribution of the remaining joint-related features using boxplots.

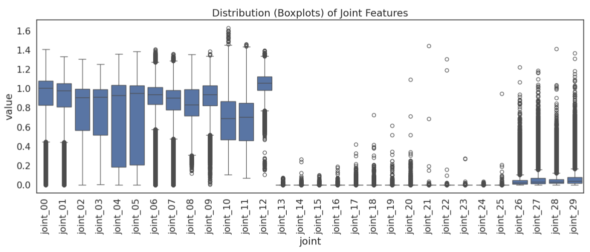

Figure 1: Distribution (Boxplots) of joint features

We found that the features from joint_13 to joint_29 were heavily skewed toward small values. To reduce this skewness, we applied logarithmic amplification.

$$x_{\text{transformed}} = \log(1 + \alpha x)$$

with α being the amplification factor.
Specifically, we separated these features into two groups, applying different strengths of amplification:

- **joint_13 to joint_joint 25** were assigned to the *low amplification group* (processed with $\alpha = 50$).
- **joint_26 to joint_29** were assigned to the *high amplification group* (processed with $\alpha = 5000$).

Then we encoded the categorical features n_legs, n_hands, and n_eyes: each of them was converted into a boolean variable, where the positive value corresponds to the categorical value "two".

When it came to building our models, we tried different methods: Recurrent Neural Network (RNN), Bidirectional Recurrent Neural Network (BiRNN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Gated Recurrent Unit (GRU) and Bidirectional Recurrent Unit (BiGRU).
We have also tried different loss functions:

- Cross-Entropy loss function

$$-\sum_{n=1}^{N} \mathbf{t}_n^T \log g(x_n|w) \tag{1}$$

- Weighted Cross Entropy loss function

$$-\sum_{n=1}^{N}\sum_{c=1}^{K} \alpha_c t_{n,c} \log g_c(x_n|w) \tag{2}$$

$$\alpha_c = \frac{N}{K \cdot N_c}$$

  With:

  - N: total number of samples
  - K: number of classes
  - N_c: number of samples in class c

  Rare classes have larger α_c, so their errors count more.

For regularization, we consistently used early stopping with a patience of 50. Subsequently, we experimented with tuning the hyperparameters. For the WINDOW and STRIDE hyperparameters, the optimal values were found to be 40 and 5, respectively, and these were kept constant across all models reported in 1. Similarly, BATCH_SIZE was fixed at 128, the learning rate at 1×10^{-3}, the number of layers at 2, and the number of neurons per layer at 256. L1 and L2 regularization were also tested but led to underperforming results.

4 Experiments

See 1 comparing the different models we tried, with some relevant characteristics that were changed from model to model, as well as the local validation F1 score for each of them and their test F1 score when submitted on Kaggle.
Most of our models were trained on stratified 60% train / 20% test / 20% validation, except for the ones marked with an asterisk (*) on the table, which were trained on stratified 90% train / 10% validation split to maximize the training data before the submission.

5 Results

The best model achieves an F1-score of 0.9666 on the test set and a F1-score of 1.000 on the validation set which indicates excellent performance and a strong ability to generalize.
Also, the F1 scores with the 90% train / 10% validation split are better than the scores with the 60% train / 20% test / 20% validation split.
Moreover, we can see that, in general, RNN gives us lower results than the other methods.

6 Discussion

The fact that the validation F1 reaches 1.000 means that the model can perform perfectly on unseen data making correct predictions. The test F1 remains very high with 0.9666 which confirms that the model is not overfitting to the validation set and instead it mantains excellent performance on a separate, unseen dataset.
Looking at the difference in performance between the models with the different data splits, we can deduce that in this case a bigger training set is conductive to better results. Obviously, having the validation set be too small would also lead to problems, but, with a dataset of this size, 90% train / 10% validation seems to be a good balance.

7 Conclusions

As part of the future development of this work, several directions appear particularly promising.

One possibility is to design a **two-stage decision system**, in which a first model handles the high-confidence cases, while a second model is queried only when the initial prediction is uncertain.

Another potential improvement involves introducing a **grid search hyperparameter optimisation** procedure, to systematically explore model configurations and enhance performance.

Table 1: Comparison of our models, named after Spanish cities. Best results are highlighted in **bold**.

Model	Type	Loss Function	Dropout Rate	Validation F1	Test F1
Madrid	BiLTSM	Cross Entropy	0.2	0.9221	0.9434
Barcelona	GRU	Cross Entropy	0.2	0.9342	0.9339
Valencia	RNN	Cross Entropy	0.3	0.8457	0.8807
Zaragoza	RNN	Cross Entropy	0.15	0.7861	0.7610
Salamanca	BiGRU	Cross Entropy	0.2	0.9382	0.9318
Toledo	GRU	Cross Entropy	0.25	0.9306	0.9240
Málaga	LSTM	Cross Entropy	0.2	0.9195	0.9239
Bilbao	RNN	Weighted Cross Entropy	0.2	0.8615	0.7609
Córdoba	LSTM	Weighted Cross Entropy	0.2	0.9312	0.8977
Granada *	LSTM	Cross Entropy	0.2	0.9326	0.9611
Valladolid *	LSTM	Weighted Cross Entropy	0.2	0.9409	0.9476
Murcia *	GRU	Cross Entropy	0.2	0.9475	0.9515
Segovia	BiLSTM	Cross Entropy	0.2	0.8943	0.8563
Sevilla *	GRU	Cross Entropy	0.2	**1.0000**	**0.9666**