

AN2DL - First Challenge Report

Guys In The Matrix

Emanuele Severino, Vincenzo Del Grosso, Mattia Rusconi, Riccardo Scarabelli

EmaSeve, cienzman, Mattia Rusconi, rikis02

280373, 278565, 274390, 279875

December 23, 2025

1 Introduction

This project addresses multiclass time-series classification for pain-level prediction using joint movements, survey responses, and physical characteristics. The workflow consists of:

- **Data Exploration:** to analyze distributions and outliers;
- **Preprocessing:** including scaling, feature selection, and handling class imbalance;
- **Model Building:**
- **Training and Validation:** to choose the architecture and the model with also hyperparameter optimization via grid search;
- **Final Training:** Only one time, **as the very last thing** test the model on the provided test set. **This is the very last step: NEVER CHOOSE THE MODEL ON THE TEST SET.**

2 Problem Analysis

2.1 Dataset Characteristics

Our exploratory analysis of the *Pirate Pain Dataset* revealed several key patterns. Many joint features form highly correlated groups (e.g., `joint_00–joint_07`,

`joint_26–joint_29`), while some joints are almost constant or extremely sparse (`joint_13–joint_25`). Anatomical descriptors are strongly imbalanced and redundant, and `joint_30` is constant. In contrast, joints such as `joint_08–joint_12` display rich continuous variability, suggesting informative motion patterns.

2.2 Main Challenges

The dataset poses several modeling difficulties: (i) strong feature correlations may induce redundancy or, alternatively, encode meaningful motion dependencies; (ii) severe class and feature imbalance risks biasing the model; (iii) skewed and sparse signals complicate normalization; and (iv) the temporal nature of the data requires sequence-aware architectures.

2.3 Initial Assumptions

Based on these observations, we assume that correlated joints are not necessarily redundant, since their temporal dynamics may differ, and that effective modeling requires proper normalization and recurrent architectures (RNN/GRU/LSTM) capable of capturing temporal evolution.

2.4 Data Preparation and Feature Processing

Preprocessing included casting joint features to `float32`, simplifying the metadata by replacing highly correlated anatomical descriptors with a single binary indicator, and removing constant features. Mutual information confirmed that joints and survey values carry most of the predictive signal. To stabilize heavy-tailed distributions, we evaluated both Min-Max and Robust Scaler normalization. A stratified train/validation split preserved class proportions, and class-weighted losses were used to mitigate label imbalance during training.

3 Method

The classification task is handled by a deep Recurrent Neural Network (RNN) model, implemented as a `RecurrentClassifier`. Given an input sequence $x \in \mathbb{R}^{T \times F}$ of length T and F features, the model processes x through a RNN, GRU, or LSTM. After the final layer, we extract the last hidden state $h_T^{(L)}$ —or its forward/backward concatenation in the bidirectional setting—and pass it to an MLP classification head:

$$\hat{y} = \text{softmax}\left(g\left(h_T^{(L)}\right)\right), \quad (1)$$

where $g(\cdot)$ is a sequence of linear, ReLU, and dropout layers.

To address class imbalance, we train the model with a weighted cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = - \sum_{c=1}^C w_c y_c \log \hat{y}_c, \quad (2)$$

optionally augmented with L1/L2 regularization:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta\|_2^2. \quad (3)$$

Optimization is performed with AdamW and mixed-precision training. Model selection relies on weighted F1 and early stopping on the validation set.

An important choice to mention is that **we replace the standard `arg max` with a class-specific thresholding scheme designed to compensate for the dominant *no_pain* class**. Let p_c denote the predicted probability for class c and t_c the corresponding confidence threshold. Instead of selecting the largest p_c , we first check whether any class satisfies $p_c \geq t_c$:

$$\hat{c} = \begin{cases} \text{any class } c \text{ such that } p_c \geq t_c, \\ \arg \max_c p_c & \text{otherwise.} \end{cases} \quad (4)$$

This mechanism encourages predictions of the minority classes (*low_pain*, *high_pain*) by requiring only moderate confidence for them, while enforcing a stricter condition for *no_pain*. If multiple classes satisfy their threshold or none does, the prediction reverts to the usual `arg max`. In this way, the model becomes less biased toward the majority class and more sensitive to ambiguous cases where minority-class evidence is present.

3.1 Cross-Validation and Hyperparameter Selection.

We use a ***K*-shuffle-split cross-validation** to obtain robust performance estimates: pirates (`sample_index`) are randomly partitioned into training and validation set, with all samples from a user in a single split. For each fold, the model is re-initialized, features normalized, and fixed-length sliding windows constructed. Validation F1 is recorded per split, and mean \pm std of best F1 across splits is reported.

Hyperparameters (e.g., RNN type, hidden size, dropout, bidirectionality, window size) are selected via **grid search** over this CV scheme. The configuration achieving the highest mean F1 is used for final training.

3.2 Final Training.

The final model is trained on the full dataset with the chosen hyperparameters, same preprocessing, windowing, and optimization strategy. Epochs are set at the best

epochs (also test with the mean epochs have been performed) based on CV results.

4 Experiments

We conducted a series of experiments to evaluate how different feature subsets affect model performance. Each experiment follows the same training pipeline, including early stopping and validation-based model restoration. Table 1 reports the best validation F1 and per-class F1 scores.

Table 1: Comparison of feature-selection experiments. Best values per column in **bold**.

Exp.	Features	Best Val F1	F1 ₀	F1 ₁
F1 ₂				
1	Full joints (selected) + survey	0.9405	0.9720	0.9071
0.7014				
2	Unique joints only + survey	0.9372	0.9700	0.8779
0.7470				
3	Non-unique joints subset + survey	0.8702	0.9700	0.8479
0.6670				
4	Outlier removal + undersampling	—	—	—
—				

Experiment 1 uses a broad set of informative joint features and achieves the highest overall validation F1. **Experiment 2** restricts the model to unique joint features and reaches a comparable score, with the best performance

on the *high-pain* class. **Experiment 3** evaluates only non-unique joints, resulting in a noticeable drop in performance, especially for *high-pain*. **Experiment 4** attempts outlier-based undersampling; however, due to the high-dimensional and noisy feature space, reliable neighbourhood-based filtering could not be performed.

Overall, results show that informative joint diversity is crucial for robustness, while aggressive feature reduction or undersampling can significantly harm performance.

5 Results

Our baseline is the majority-class predictor, achieving an F1-score of 0.67. All proposed models substantially outperform this reference, confirming that the network effectively captures temporal structure and discriminative information beyond class frequency. The best feature configuration reaches a validation F1 of 0.94, with strong performance on the *no-pain* (0.97) and *low-pain* (0.91) classes, and a moderate yet meaningful score on *high-pain* (0.70). Throughout the experiments, every design choice (feature selection, joint subsets, undersampling attempts) was compared against the random/majority baseline to ensure that improvements were genuine and not the result of noise or overfitting.

6 Discussion

The model demonstrates robust performance on the *no-pain* class, largely due to its abundance in the dataset, which facilitates stable learning. Predictions for the *low-pain* class are also reliable, though slightly less consistent. The main limitation concerns the *high-pain* class: its scarce representation leads to reduced generalization and higher uncertainty, reflected in a drop of the per-class F1 to around 0.7.

Attempts to mitigate this—such as feature subset selection, class-weighting, and undersampling of outliers—provided incremental

improvements but did not fully compensate for the imbalance. Moreover, identifying outliers in high-dimensional joint spaces proved difficult; many samples were too sparsely distributed to allow reliable cluster-based removal.

7 Conclusions

This work presents a recurrent neural architecture for multiclass pain-level prediction from multivariate temporal data, achieving a significant improvement over the majority-

class baseline. The best-performing configuration attains a validation F1 of 0.94, with strong detection of *no-pain* and *low-pain* levels.

Future work may focus on: (i) more advanced imbalance-handling strategies (e.g., a sort of synthetic minority sampling for time series); (ii) multivariate outlier detection to filter noisy or unreliable subjects; (iii) an incorporation of additional survey signals or domain-informed features. These directions may further enhance performance, particularly for the challenging *high-pain* class.