AN2DL - First Challenge Report
AN2DL Team

Daniele Spini, Luoxi Massimo Yu, Simone Delbuono, Giovanni Fiumalbi

Daniele Spini, Luoxi Massimo Yu, Simone Delbuono, Giovanni Fiumalbi

279837, 270826, 280048, 287388

November 17, 2025

1 Introduction

This project addresses a **supervised** time-series classification task using deep learning. The main **goal** is to predict the pain level of each sequence using the **F1 score** as the evaluation metric. This requires analyzing the dataset to remove redundant features, detect outliers, construct temporally consistent sequences, and handle the strong **class imbalance** while selecting effective **hyperparameters** and architectures.

2 Problem Analysis

1. **Dataset characteristics**

   The dataset includes:

   - `pirate_pain_train.csv` (105760, 40)
   - `pirate_pain_train_labels.csv` (661, 2)
   - `pirate_pain_test.csv` (211840, 40)

   Features:

   - **sample index**, **time**
   - four **pain-related** measurements
   - three categorical **body-part counters**
   - thirty **joint-angle** measurements

   Each sample index has 160 time steps. Labels: *no_pain*, *low_pain*, *high_pain*.

2. **Main challenges**

   - Detecting **correlated** or **non-informative** features (e.g., joint_30 constant)
   - Handling **outliers** (e.g., missing limbs)
   - Dealing with severe **class imbalance** (77.13% / 14.22% / 8.47%)
   - Applying a correct **sample-wise split** to avoid data leakage
   - Selecting appropriate **feature engineering** techniques
   - Choosing architectures capable of capturing **long-term dependencies**
   - Preventing **overfitting** with robust regularization
   - Tuning critical **hyperparameters** (window size, stride, learning rate, hidden size)

3. **Initial assumptions**

   - Joint-angle and pain signals vary **smoothly** over time
   - Sequences must be split by **sample index** to preserve independence
   - The class imbalance strongly affects learning and requires **weighted** or **focal loss**
   - **Ensembling** improves stability and generalization by combining complementary models

3 Method

The approach adopted by the team was incremental, starting from a simple architecture and subsequently adding data analysis and augmentation modules, as well as network functional layers.

As explained, the first relevant issue to address was the severe class imbalance. This problem was tackled in several ways that evolved throughout the challenge. An initial approach involved applying an **oversampling** technique to the minority classes with the addition of **Gaussian noise**, or applying **SMOTE** (Synthetic Minority Over-sampling Technique). Both strategies yielded slightly better results, but at the cost of overfitting the data and excessively increasing the sample count. Thus, the team decided to adopt a **focal loss** function with inverse frequency weights, so that classification errors on minority labels would result in greater loss values and guide the model to balance its learning toward less represented classes.

Another key aspect we focused on was the **extraction of significant features** and temporal patterns from the time series. Understanding how joint angles evolve over time is essential, as it provides richer context for the network to specialize. For this reason, we added features such as the **velocity** and **acceleration** of the joint angles, along with the rolling **mean and standard deviation** computed using a window of size 10.

The data was prepared by applying one-hot encoding to categorical variables, min–max normalization to numerical features, and by removing low-variance features and features with high correlation.

While exploring data preparation techniques that could better suit our data, we also experimented with different types of RNN, trying to understand the strengths and weaknesses of each one. In the majority of attempts, **bidirectional LSTM** and **GRU** turned out to be the best-performing models, though some highly performing non-bidirectional models were also found, especially after introducing feature engineering. We then introduced a **1D convolutional layer** before the RNN layer, with the purpose of extracting local features while preserving temporal relationships. Additionally, an **attention mechanism** was applied to the RNN outputs. The attention layer computes timestep-wise importance weights and produces a context vector which is fed as input to a final classification layer.

Among the most important techniques employed to optimize learning and improve the model's robustness are:

- **Label smoothing**, used to introduce additional flexibility into the loss function
- **AdamW**, adopted as the optimizer to apply weight decay directly to the model parameters
- **A learning rate scheduler**, used to dynamically adjust the learning rate throughout training
- **Gradient clipping**, implemented to prevent excessively large gradients from destabilizing learning

Grid search was used to find the optimal parameters for the various techniques and model configurations. Finally, the use of an **ensemble method** allowed us to further improve the model's overall performance: by aggregating the predictions of our three best-performing models, we were able to obtain a more stable and accurate final output.

4 Experiments

Table 1: Impact of main techniques on performance

Configuration	F1 Score
Baseline UniLSTM	0.9368
+ Class balancing (SMOTE + Focal Loss)	0.9401
+ Feature engineering	0.9470
+ Conv1D layer	0.9479
+ Attention	0.9484

Table 2: Per-class metrics (Validation set – 30% of training data)

Class	Precision	Recall	F1 Score	Support
No Pain	0.9935	1.0000	0.9968	154
Low Pain	0.9655	1.0000	0.9825	28
High Pain	1.0000	0.8824	0.9375	17
Macro Avg	0.9864	0.9608	0.9722	199

5 Results

5.1 Key Findings

- **Baseline:** A simple recurrent model without class balancing or feature engineering achieved low performance, with the F1 score dominated by the *no_pain* class.
- **Preprocessing & Features:** Removing constant/correlated features, applying min–max scaling, and adding derived statistics improved F1, especially for *low_pain*.
- **Balanced Loss:** Focal loss with inverse-frequency weighting increased minority-class re-

call more effectively than oversampling and SMOTE.

- **Architecture:** Adding a 1D-CNN layer and an attention mechanism improved both local and long-range temporal modeling.
- **Best Model:** The top single model was a CNN + UniLSTM + Attention architecture with AdamW, scheduler, gradient clipping, and label smoothing.
- **Ensemble:** Combining the three best models yielded the highest and most stable F1 score.

5.2 Unexpected Outcomes

- Gaussian-noise oversampling increased overfitting.
- SMOTE generated unrealistic temporal patterns in sequential data.
- Some non-bidirectional RNNs performed surprisingly well after feature engineering.

6 Discussion

Strengths. The model combines robust preprocessing, sample-wise validation, and an ensemble of cross-validated models, which improves stability and overall F1 performance.

Weaknesses. The architecture is relatively complex, and the "high-pain" class remains more difficult to detect due to strong class imbalance, making the model harder to interpret and more sensitive to hyperparameter choices.

Limitations. Results are constrained by the small and imbalanced dataset and may not fully generalize to populations or recording conditions different from those in the training data.

Assumptions. We assume that the pain label associated with each sample_index is constant throughout the time window. We also assume that train and test sets are drawn from the same distribution and that removing highly correlated or low-variance features does not discard clinically relevant information.

7 Conclusions

In this project, we developed an end-to-end pipeline for pain-level classification from time-series data, covering preprocessing, feature engineering, model training, and evaluation. We first ensembled the models obtained from cross-validation and then combined these with other strong models trained with different hyperparameters using weighted voting. This two-stage ensembling strategy improved stability and F1 performance compared to any single model.

References

Exercise Session Notebooks