POLITECNICO
MILANO 1863

AN2DL - First Challenge Report
harryplotter

Leonardo Bertolani, Giulio Mantovi, Riccardo Masetti, Samuele Tondelli

leobertolani, giuliomanto, riccardomasetti, samueletondelli

273866, 274615, 273574, 273360

November 19, 2025

1 Introduction

This project focuses on *multivariate time series classification* using **deep learning** techniques. We focused on:

- **Analyzing and preprocessing the dataset**, dropping useless features and records, and investigating noise patterns through extensive testing.
- **Developing recurrent (baseline), convolutional and attention-based architectures** for pain classification.
- **Tuning hyperparameters** and applying regularization techniques to improve generalization.

2 Problem Analysis

We were provided a dataset composed of 661 unique sequences, each 160 time steps long, each corresponding to one subject identified by `sample_index`. Each sequence included 31 joint measurements (`joint_00`–`joint_30`), 4 pain survey features, and subject characteristics (`n_legs`, `n_hands`, `n_eyes`). A separate file contained the training labels, classifying each subject into three pain levels: `high_pain`, `no_pain`, and `low_pain`. Our objective was to analyze and preprocess this training dataset, in order to subsequently train the best possible model capable of predicting these pain classifications.

2.1 Dataset Analysis

At first glance, the dataset appeared highly noisy and showed a strong imbalance across the three pain classes. This prompted us to perform a series of analyses to better understand its characteristics, which in turn guided our preprocessing strategy.

1. **Feature correlation**: We computed the *Pearson correlation matrix* (Fig. 1) and noticed that several features exhibited very similar patterns (e.g., `n_hands`, `n_legs`, and `n_eyes`), and that `joint_11` was highly correlated (93%) with `joint_10`; this led us to identify and drop redundant columns.
2. **Pain survey distributions**: We analyzed the distribution of the features from `pain_survey_1` to `pain_survey_4` to understand their relationship with true pain labels and assess their predictive value. Our analyses revealed that pain survey features showed a weak correlation with actual pain levels. Based on the distribution shown in Fig. 2 and additional experiments, these four features were found to contribute little to no information to pain prediction and we decided

to exclude them from training.

3. **Outlier detection**: Analyzing the statistical properties of each feature's time series revealed unrealistic values in several joint measurements (especially from `joint_13` to `joint_25`). This led us to test two approaches: dropping the problematic features entirely or attempting to extract meaningful information by identifying and removing outliers. Both strategies are detailed in Section 3.

4. **Sequence autocorrelation**: To determine the window size and stride for our models, we analyzed the average autocorrelation of the dataset sequences and identified an optimal range of 10 to 12.

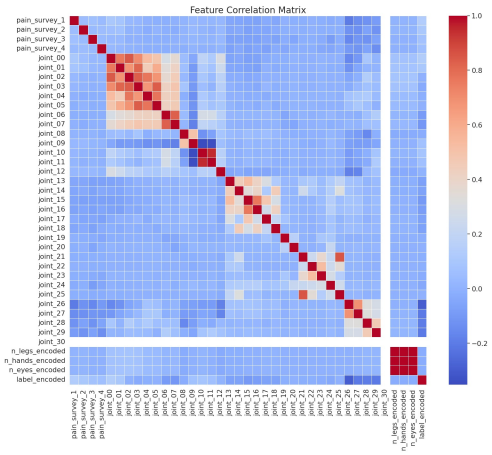

Figure 1: Heatmap of the **Pearson correlation matrix** for all features. Red indicates a strong positive correlation, blue indicates a negative correlation, and white/light colors represent near-zero correlation.

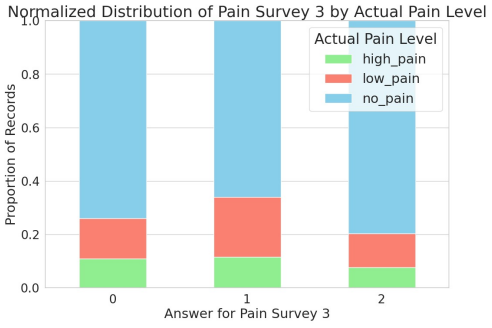

Figure 2: The figure represents the normalized distribution of the feature `Pain_Survey_3` by Actual Pain Level

3 Method

Our approach involved training two identical encoder-decoder RNN networks on two differently "sanitised" versions of the dataset.

First, we applied a **common preprocessing pipeline**: we kept only the first 30 joint channels (joints 0–29), dropped the timestamp, and **MinMax-scaled** all data to a $[0, 1]$ range (which provided better results than using a RobustScaler). Then, the data was split into an 80% training and 20% validation sets. Common regularization techniques have been applied to both branches, combining L_1 and L_2 **regularization** both with a factor of 10^{-4}, a **dropout** rate of 0.5, and **early stopping** with a patience of 500.

The two branches differed in their sanitisation strategy:

- **Branch no_mid** focused on feature selection, removing the 13 least stable joints (joints 13–25) to create an 18-dimensional input.

- **Branch no_outl** focused on outlier removal, retaining all joints but 30 and 11 and discarding outliers, identified as the training samples whose maximum joint value exceeded the 0.9996 quantile.

3.1 Model Architecture

The model architecture for both branches consists of a single-layer **GRU encoder** (128 neurons) followed by a **GRU decoder with Bahdanau attention** (128 neurons). The final classification is obtained by mapping the last hidden state of the decoder linearly to the class logits. The models were trained using the **Adam optimizer** (learning rate 10^{-3}, batch size 128) for up to 2000 epochs. The model with the highest validation F1 score was considered the best.

For training and testing purposes, each sequence was segmented into windows with a **window size of 12** and a **stride of 6**. Training was performed using **holdout**. The final classification of a sequence was determined by a **majority vote** across all its constituent windows. This method produced the best results overall, as shown in Section 5.

4 Experiments

Our final choices emerged from a range of experiments, involving both dataset preprocessing and architectural design.

4.1 Failed experiments

- **Data Augmentation:** We attempted to mitigate class imbalance for the 'low' and 'high pain' classes using time series augmentation (e.g. *jittering*). This didn't lead to any improvement.
- **Feature engineering:** We attempted to extract additional information by adding features for standard deviation and maximum angle per time series, motivated by their potential medical relevance, but these additions did not improve model performance.
- **Noise Filtering:** We tried denoising the training set, using **LPFs** and **EMA** processes, none of which provided meaningful results. We also tested a **per-joint trend–noise** reconstruction method, which yielded an average autocorrelation similar to the test set, but it still did not improve performance.
- **Weighted loss:** Given the strong class imbalance in the dataset, we experimented with **weighted cross-entropy loss** to address this issue. However, this approach unexpectedly caused models to over-predict the `high_pain` class, ultimately degrading overall performance.

4.2 Experimental Architectures

As for the architectures, we sought for the best solutions for time series classification, and we came up with a variety of approaches.

- **Baseline**: At first, we applied a baseline architecture consisting of a **two-layer GRU[1]/LSTM[2]**.
- **Convolutional Neural Network**: Based on some studies [3] we found, we tried adding a single and two **Conv1D layers** before the baseline. From our experiments we found the 1 layer generally performing better than the 2 layers.

5 Results

The table below lists the most significant results obtained from our trials. The same preprocessing was applied to all the listed models, and only the best results are shown. The final two models chosen for submission are in bold.

Table 1: Validation and public test F1-scores obtained on the experiments.

Model	Validation F1	Public Test F1
Baseline LSTM	0.8352	0.9317
Baseline GRU	0.8601	0.9453
CNN 1 layer	0.8541	0.8955
CNN 2 layer	0.8147	0.9205
Attention no_mid	0.8640	**0.9538**
Attention no_outl	**0.8821**	0.9219

6 Discussion

Although the best models demonstrated strong generalization, they had difficulty distinguishing between the `low-pain` and `high-pain` classes. This was likely due to the absence of features that could characterize the `low-pain` class in the training set.

Autocorrelation analysis revealed that the training set contained noise that was not present in the test set, possibly by design. Additionally, the validation F1 scores were unexpectedly lower than the public test scores. We attribute this discrepancy to the weighted F1 metric, which favors majority class predictions and complicates generalization assessment.

Finally, the results show no evidence of a superior architecture for this dataset. In particular, the baseline achieved relatively high scores.

7 Conclusions

Our approach achieved strong generalization despite challenges in distinguishing similar pain classes. Future work should prioritize advanced denoising techniques to address training data noise and explore complex architectures like transformers [4] and more carefully designed CNN layers.

References

[1] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[3] R. Mutegeki and D. S. Han. A cnn-lstm approach to human activity recognition. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 362–366, 2020.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.