# Machine Learning for ALSFRS-R Score Prediction: Making Sense of the Sensor Data

Ritesh Mehta[1,*], Aleksandar Pramov[1] and Shashank Verma[1]

[1]Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332

**Abstract**

Amyotrophic Lateral Sclerosis (ALS) is characterized as a rapidly progressive neurodegenerative disease that presents individuals with limited treatment options in the realm of medical interventions and therapies. The disease showcases a diverse range of onset patterns and progression trajectories, emphasizing the critical importance of early detection of functional decline to enable tailored care strategies and timely therapeutic interventions. The present investigation, spearheaded by the iDPP@CLEF 2024 challenge, focuses on utilizing sensor-derived data obtained through an app. This data is used to construct various machine learning models specifically designed to forecast the advancement of the ALS Functional Rating Scale-Revised (ALSFRS-R) score, leveraging the dataset provided by the organizers. In our analysis, multiple predictive models were evaluated to determine their efficacy in handling ALS sensor data. The temporal aspect of the sensor data was compressed and amalgamated using statistical methods, thereby augmenting the interpretability and applicability of the gathered information for predictive modeling objectives. The models that demonstrated optimal performance were a naive baseline and ElasticNet regression. The naive model achieved a Mean Absolute Error (MAE) of 0.20 and a Root Mean Square Error (RMSE) of 0.49, slightly outperforming the ElasticNet model, which recorded an MAE of 0.22 and an RMSE of 0.50. Our comparative analysis suggests that while the naive approach yielded marginally better predictive accuracy, the ElasticNet model provides a robust framework for understanding feature contributions.

**Keywords**

ALS, ALSFRS-R score prediction, sensor data analysis, ElasticNet regression, Predictive modeling in neurodegenerative diseases



## 1. Introduction

Amyotrophic Lateral Sclerosis (ALS), also known as Lou Gehrig's disease, is a progressive neurodegenerative disorder that affects motor neurons in the brain and spinal cord. The ALS Functional Rating Scale-Revised (ALSFRS-R), widely utilized in clinical and research settings, stands as a critical metric employed by healthcare professionals to evaluate the functional state of ALS patients. The precise forecasting of ALSFRS-R scores is essential for assessing disease progression and the effectiveness of therapeutic measures. Recent developments in sensor technology have opened up new avenues for continuous, non-invasive monitoring of ALS symptoms. The fusion of sensor data with predictive modeling presents the potential for more accurate and timely forecasts of disease progression, significantly benefiting patient care and treatment management.

The iDPP@CLEF 2024 competition is an initiative aimed at leveraging sensor data [1] [2] and machine learning techniques to predict ALS progression. Task 1 involves predicting the ALSFRS-R scores assigned by medical professionals using sensor data collected via a dedicated app. Task 2 focuses on predicting self-assessment scores recorded frequently by patients. These tasks aim to enhance the accuracy and timeliness of ALS symptom monitoring and forecasting, providing valuable insights for patient care and treatment management. We hypothesized that the relatively small dataset, coupled with a high number of features, will pose significant challenges in our modeling efforts, necessitating the use of strong regularization techniques to avoid overfitting.

To address the prediction of ALSFRS-R scores, we implemented several techniques. We started with a naive model to form a baseline and establish a reference for comparison. The naive model simply carries the last observed value forward. We then explored various Machine Learning algorithms for regression,

*CLEF 2024: Conference and Labs of the Evaluation Forum, September 9-12, 2024, Grenoble, France*

*All authors contributed equally.

rmehta307@gatech.edu (R. Mehta); apramov3@gatech.edu (A. Pramov); sverma342@gatech.edu (S. Verma)

as well as a Long Short-Term Memory (LSTM) neural network, to model the temporal dependencies in the sequential sensor data. Performance was evaluated via the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics identified by the challenge organisers. The main outcome of our analysis confirmed the initial hypothesis, revealing that the small dataset size indeed complicated the modeling process, but effective regularization strategies, particularly through ElasticNet, were helpful.

The paper is structured as follows: Section 2 reviews related work, providing context and background on ALSFRS-R score prediction and the integration of sensor data with machine learning techniques. Section 3 details the methodology, including data processing steps and the modeling approaches we employed. Section 4 presents the results of our experiments, comparing the performance of different models and discusses our findings. Section 5 outlines future work directions, and Section 6 concludes the study.

## 2. Related Work

In recent years, the integration of sensor data and machine learning (ML) techniques [3] has shown promising results in improving the accuracy and reliability of ALSFRS-R score predictions. [4] developed machine learning models to objectively measure ALS disease severity using voice samples and accelerometer data, while [5] further focuses specifically on on deep learning methods to predict ALS disease progression. Outside of the ALS prediction field, [6] demonstrated the potential of temporal models in the healthcare domain by integrating EHR data; [7] applied various ML models to sensor data from accelerometers attached to dairy cattle for disease prediction and [8] demonstrates how temporal patterns from clinical and imaging data can be used to predict residual survival for cancer patients.

A more technical field of the literature deals specifically with the longitudinal aspect of the data and its effect on ML and DL methods. The recent wider application of ML methods in biomedical data has necessitated adapting traditional models to handle repeated measurements over time. [9]. [10] extends Random Forests to handle fixed and random effects. [11, 12] give a more general overview of existing methods, while [13] focuses specifically on a neural network adaptation that can handle fixed and random effects.

## 3. Methodology

### 3.1. Data Description and Preprocessing

The raw features dataset provided for the competition consists of two parts: static data and sensor data. The former contains patient-specific baseline (constant) data, such as sex, age etc. The latter contains 90 time-series of patient-specific sensor data, collected over an average of nine months through a dedicated app developed by the BRAINTEASER project, using a fitness smartwatch in the context of clinical trials. For some patients, the clinical data starts before the app data, and some patients whose app data goes beyond the last observed clinical data. This required some synchronization between the clinical and the app data, in order to avoid look-ahead bias. Let $t_1^s$ be the first sensor time point in the raw data, $t_1^c$ be the first clinical time point in the raw data, as defined by the days from diagnosis for a given patient. Overall, we consider the time-overlap between clinical and sensor data. Some special cases are handled as follows:

- If $t_1^s > t_1^c$, then we discard clinical data prior to one step back from $t_1^s$. For example, consider a patient that has clinical observations at days 690, 780, 873, and sensor data from 800 onwards. We discard the clinical observation 690, but keep the rest. As we will see later, the reason for this is that we will use the previous clinical score as a feature and hence the first value to be predicted would be the target at day 873. That first value will be predicted using the sensor data between 800 and 872, as well as previous clinical visit from the 780th day since diagnosis.
- If $t_1^s \leq t_1^c$, then we use $t_1^s$ as the starting point for the sensor data and we do not discard any clinical data.

- Sensor data after the last clinical observation is discarded.
- If the clinical data ends after the sensor data, we consider that point only if it is at maximum of 60 days from the last observed sensor data point. This is done so that we do not use too distant sensor data as a feature.
- Some patients have only one clinical observation. As we will use the previous clinical question responses in the set of features for the models, those patients will be discarded due to the missing previous response value.

Some patients had a few missing static data observations as well. Those were imputed by a simple median over all other patients for the respective feature.

The target variable (i.e., questionnaire responses) provides highly informative insights when visualized and analyzed over time, as shown in Figure 1. The visualization is done on the actual dataset we will use, i.e. after the pre-processing steps described above.

The curves reveal an initial period of slow degradation followed by a more rapid decline for many questions. We also notice that the scores change pretty slowly for most patients over the course of 100 days (the average number of days between consecutive visits to the clinician). Additionally, only a few patients manage to recover their scores to better levels. That observation has to be cautioned by the observation that the amount of patients with a longer follow-up decreases strongly over time. While at the beginning there are $n = 51$ observed patients (i.e. the full sample), this number drops to $n = 37$ for the second follow-up and decreases further to just $n = 5$ and $n = 2$ for the fifth and sixth follow-up respectively. This is also evident by the higher CI (Confidence Interval) bars on Figure 1.

A key insight here, as a result of above mentioned observations, is that the previous value of the score could be a useful engineered feature, which will also dictate the data pre-processing steps. Additionally, the heterogeneity in question types supports two modeling approaches: (1) treating the data as panel data with “question type" as a grouping variable for random effects, and (2) modeling each question separately.

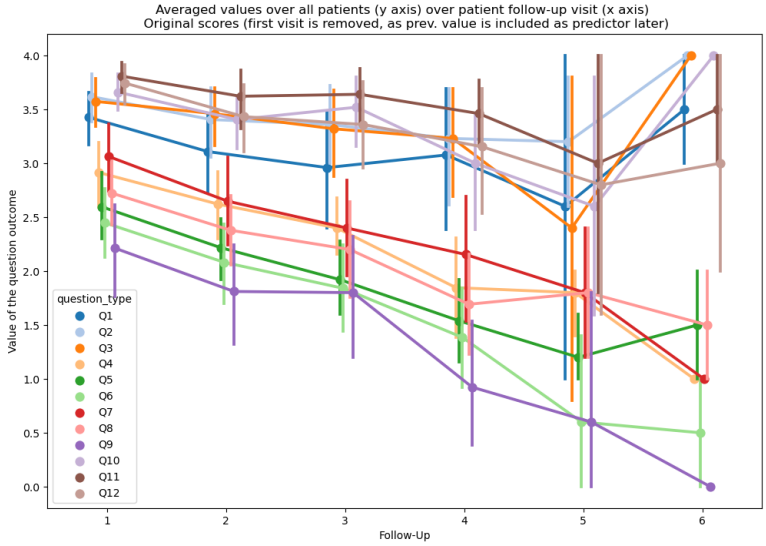



**Figure 1:** Averaged question response values for each follow-up time point. The x-axis shows the ordered visits by patient, whereas the y axis shows the average score. The vertical bars indicate a 95% CI around the estimated mean. As we have fewer and fewer patients for longer follow-ups, the CI widen. There is some degree of variability with respect to the intercept among the questions, as well as the slopes. The beginning periods exhibit slow deterioration which already suggests that using the previous value as predictor for the next score will be useful. The points are jittered for visualization purposes.

### 3.2. Features generation

For our modeling, we considered three groups of features: *target-based*, *static*, and *sensor-data*. While we used the static data *as-is*, we originally tried to build a baseline model based only on the target data and treat it as an autoregressive problem with some additional engineered simple transformations of the target dataset, e.g., the difference in days between two diagnosis, the first value of each patient, the previous value, one-hot encoding of the question type, the follow-up time-index (i.e. the x-axis in Figure 1). Based on this initial analysis, we quickly realized our initial hypothesis for the importance of the previous target value, stated in previous section, is valid, and we kept it as a feature for any subsequent modeling.

A central challenge in this study was the handling of the *sensor data* features. Those are observed daily (with some gaps), compared to the target (and related engineered variables such as the previous value), which is observed once every three months. To address this mismatch in the frequencies, we followed three different approaches:

**Approach 1:** By simply taking the median over a window of each sensor data column, where the window was defined to be between each two consecutive clinical visits - $[t-1, t)$.

**Approach 2:** By generating a vast array of features derived from a window in each sensor data column, where the window is between each two consecutive clinical visits - $[t-1, t)$, inspired by a feature-based time series analysis approach [14].

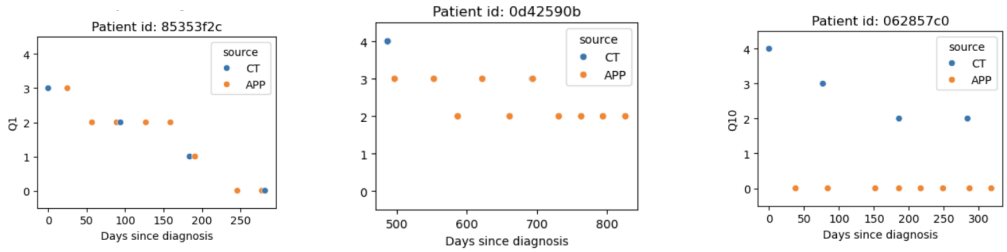
**Approach 3:** By using LSTM (Long Short Term Memory) cells for the sensor data and letting the network define the relevant transformations of the input.

Note that Approach 1 is a special and simple case of Approach 2. Here, we effectively express each sensor time window $[t-1, t)$, with just one number and a set of numbers respectively. Doing this across all the windows of all patients builds the sensor data feature set. Approach 3 on the other hand does not have pre-defined (set of) transformation(s) on the input sensor data and we let the neural network itself pick up the relevant transformation of the input. Its details are provided in section 3.5 where the LSTM model is described.

For Approach 2, we employed the `tsfresh` python library for systematic feature engineering from time-series and other sequential data [15]. Simple examples of those can be e.g. the maxima/minima/median/mean etc. over each sensor data window per each patient between two clinical visits. A full outline of the feature extraction procedure and the considered extracted features can be found on the documentation website of the package. Note that, when one considers different parameter setups of the various extracted features, one would end up with multiple extracted features per time series. We have around 100 sensor time series and many of the extracted features by `tsfresh` had problems like high amount of missing values, low variability etc. and so were removed from the set of feature candidates. On each of those extracted features we performed filtering, by calculating `tsfresh` Spearmann correlation coefficient between the feature and the (one-step-ahead) clinical values per each question, over all patients. The hypothesis tests for significance are adjusted following the procedure in [16]. For the final feature set based on the app data, we experimented with either a) keeping all the features that were deemed significant per question or b) by fixing the number of the top k (e.g. 10) in terms of lowest p-value to keep in the sensor feature set. As the final modeling was done for each question separately, we settled on the former choice.

### 3.3. Data augmentation

As already mentioned, one of the major challenges in modeling for Task1 was the lack of sufficient datapoints. Given the high amount of features, this made the models more prone to overfitting . Fitting complex deep models would exacerbate this problem further. To circumvent this issue, we decided to augment the Task1 data with that of Task2. (*Note that doing this reduces the average number of days between any two consecutive visits.*) This gave us ALSFRS-R scores progression as shows in Figure 2. There are a couple of advantages of doing this:




(a) In agreement (b) Makes patient usable (c) Not in agreement

**Figure 2:** Augmenting Task1's CT scores data with Task2's APP scores. a) An example of patient_id, question_id pair where APP scores are in agreement with CT scores b) An example of patient_id, question_id pair where adding APP scores allows the usability of this datapoint as it had only 1 CT score data c) An example of patient_id, question_id pair where the APP scores aren't in agreement with CT scores.

- adding more datapoints for training (more labeled data per patient, question tuple) as shown in Figure2(a)
- allowing certain patient's data to be not discarded entirely (due to the patient having only 1 visit to the clinician i.e. CT data) in an already short list of patients as shown in Figure2(b)

However, we cannot simply add this data unilaterally as there are a couple of concerns:

- adding the data from patient's self assigned scores results in consecutive scores being too close to each other.
- potential disagreement between the scores assigned by the clinician and that assessed by the patient resulting in cases like Figure2(c).

The solve the first problem, we simply restricted training and prediction to target scores roughly 100 days ahead, instead of using the very next available score. The second problem requires statistical analysis on a per patient, question pair to decide whether or not to augment this prediction task with patient's self assessment scores. We used the chi-squared test [17] to test for the Null the data from the two time series are sampled from the same distribution. Of the 610 patient, question pairs where this analysis was possible, we failed to reject the Null 559 times and thus their data was merged. For the subsequent analysis, we used either only Task1 data or the combined Task1+Task2 data, and we indicate which data set was used where applicable.

## 3.4. Modeling

Unless explicitly specified, all modeling efforts described are for Task1 i.e. predicting clinician's assigned scores. Our modeling setup used RMSE as the scoring function at all stages.

### 3.4.1. CV setup

The small dataset (129 datapoints per question only) posed several challenges with regards to choosing the appropriate cross-validation procedure for the hyperparameter tuning for our models. One key aspect is that the true (unseen) test set contains patients that are not present in the training set, rather than containing unseen data of the same patients. Hence, to assess the ability of our models to generalize well, we used a nested k-fold cross validation strategy [18, 19]. This consists of two loops - an inner, and an outer one.

In each inner loop, we set aside a test set of 10% containing complete data of the patients (i.e. all observations). For the remaining data, we perform a further k-fold cross-validation, whereby we also adapted it to contain a complete set of observations of each patient in both the training and the validation sets. That ensured that there is no information leakage between different times of the patient's data, as the true out of sample test (for the final submission) also contains data on unseen patients.

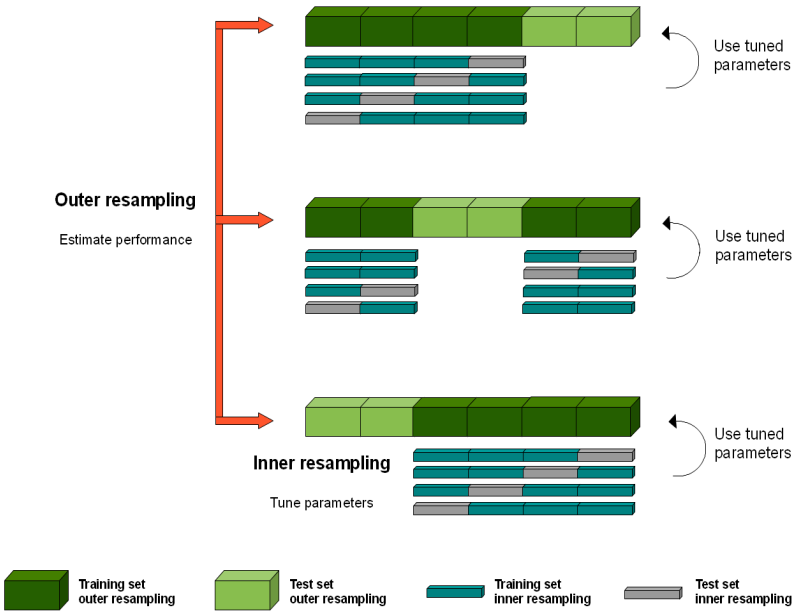



**Figure 3:** Cross-validation procedure for the hyperparameter tuning and final model fit in our work (Source: [20]). The original source illustrates the procedure without taking into account a grouped structure as in our dataset. Hence, in our work we adapted the procedure so that each patient's complete data is contained in either the inner-train/test subsets, or the outer test-set.

The outer loop then repeats the same procedure, but on another test set which covers another 10% of the patients. In total, we have 10 iterations in the outer loop, for each of which the RMSE is computed to allow for a model choice. The best hyperparameters overall are then chosen and all the data was fit using those prior the final submission.

Figure 3 illustrates the process for non-grouped data as described by [20]. The term "test inner-resampling" corresponds to what we call validation set. Overall, the additional "grouped" adaptation that we implemented to prevent information leakage, ensured that for any outer-loop/inner-loop combination, all the data for a given patient is contained in either the (outer-loop) test or the (inner-loop) train + validation set.

### 3.4.2. Traditional ML models

We experimented with a wide range of modeling techniques for predicting ALSFRS-R scores. This included traditional ML models as well as deep learning models, features described in Approach 1 as well as Approach 2 mentioned in section 3.2, with and without augmenting the training data from Task1 with that of Task2 as described in section 3.3. We achieved good results with traditional as well as deep learning models. We'll go over the details of LSTM modeling in the next section.

**Naïve Model**

As described in section 3.1, the scores for individual questions changes infrequently for an average patient over the course of two consecutive clinician visits (this is especially true since the scores are integers and don't allow partial progression from say 4 to 3). This suggests a Naive Model where the predicted score is the same as previous visit's score to serve as a baseline for all modeling approaches.

**Modeling Approaches**

In its given form, this is a multi-label multi-class classification problem. However, we can treat question_type as a covariate and transform this into a multi-class classification (with 12x datapoints). We can also train a separate model for each of the 12 questions which would also transform it into a

**Table 1**
Representative training data after pre-processing.

| Patient ID | Days since Diagnosis | Previous Value | Delta days in Future | Sensor Feature1 | ... | Sensor featureN | Future Value |
|---|---|---|---|---|---|---|---|
| **patient1** | 800 | 4 | 95 | 0.1 | ... | 22 | 3 |
| **patient2** | 700 | 3 | 90 | 0.4 | ... | 89 | 1 |
| **patient3** | 1400 | 2 | 120 | 0.2 | ... | 43 | 2 |
| **patient4** | 300 | 4 | 110 | 0.9 | ... | 09 | 4 |
| **patient5** | 500 | 4 | 100 | 0.0 | ... | 51 | 2 |

multi-class classification problem. In a similar vein, we can model the problem as a regression and round the outputs to the nearest integer in [0, 4]. Regression modeling is what worked the best in our case and this was our approach henceforth.

**The Overfitting problem**
One thing that became clear in very early stages of modeling was that overfitting was going to be the most prevalent issue which was evident across a number of models like linear regression, random forest, kNN, gradient boosted trees etc [21]. To this end we also included models geared specifically towards regularization like ElasticNet and Lasso [22]. The hyperparameter tuning was done via a grid search in the nested CV procedure described above.

**Training Data**
For the traditional ML models, the data was prepared to look as shown in Table 1, with all but last column as features and the last column as the target variable. Also note that this modeling was done on a per-question basis i.e. the $previous\_value$ and $future\_value$ in this table are for the question being modeled.

#### 3.4.3. LSTM

In our study, we employed a Long Short-Term Memory (LSTM) neural network to predict ALSFRS-R scores based on time-series sensor data. The LSTM model is well-suited for handling sequential data due to its ability to capture long-term dependencies. As depicted in Figure 4, our model processes sensor data collected over multiple time points using a series of LSTM cells. Each cell captures temporal patterns in the data at different time steps, which are then fed into subsequent cells. To handle the uneven number of days between baseline and target scores, padding was applied to standardize the input sequences. The output from the final LSTM cell is concatenated with baseline scores and static patient data to enhance the model's predictive capability. This combined feature set is then passed through multiple linear layers, each dedicated to predicting one of the twelve ALSFRS-R sub-scores. This architecture allows the model to leverage both dynamic sensor inputs and static information, providing robust predictions for each functional domain assessed by the ALSFRS-R score.

## 4. Results

### 4.1. Task1

The results for Task1 are shown in Table 2 below.

The table displays the RMSE of various models on each question on test data split from the train+val set. As can be seen here as well as Figure 5, previous_value has by far the most predictive power. For

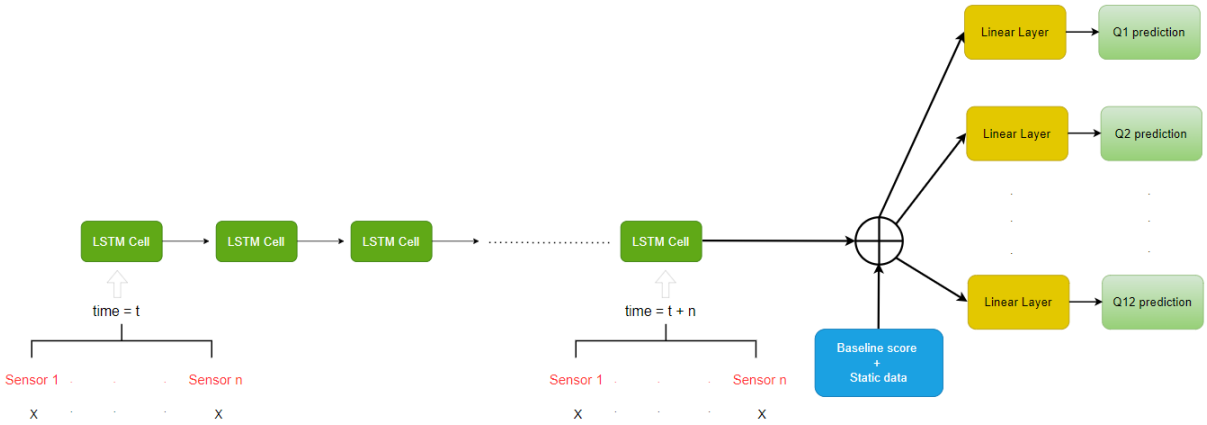



**Figure 4:** Architecture of the LSTM model used for predicting ALSFRS-R scores. The model processes time-series sensor data through sequential LSTM cells, capturing temporal dependencies. The output is combined with baseline scores and static patient data, then fed into multiple linear layers to predict each of the twelve ALSFRS-R sub-scores.

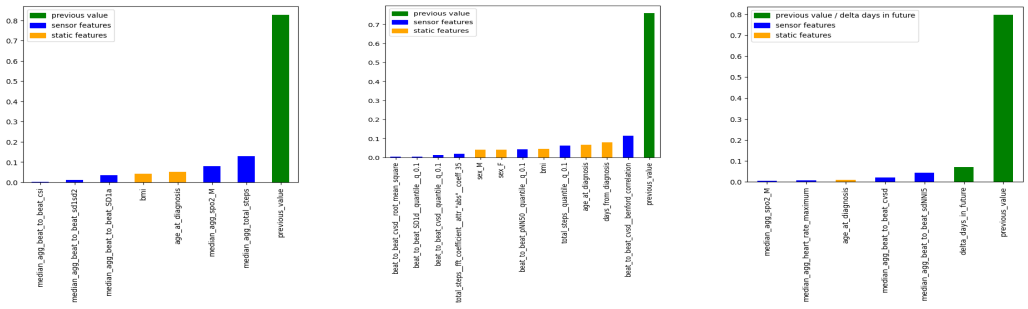
**Table 2**
Task1's RMSE for various models separately modeling each of the 12 questions in ALSFRS-R scores. Green boxes denote the best performing model for the question. FS = FeatureSet, denoting whether median sensor features are used or TsFresh features are used (more details in section3.2. T1 denotes that only Task1 data was used for training whereas T1+2 denotes that Task1 data was augmented with that of Task2 for training. EN denotes ElasticNet model.

| FS | Model | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | **Naive** | 0.47 | 0.38 | 0.43 | 0.65 | 0.69 | 0.77 | 0.74 | 0.62 | 0.78 | 0.5 | 0.46 | 0.54 |
| W/o TsFresh | **T1 EN** | 0.47 | 0.38 | 0.43 | 0.67 | 0.73 | 0.75 | 0.77 | 0.62 | 0.85 | 0.59 | 0.47 | 0.65 |
| | **T1 Lasso** | 0.47 | 0.38 | 0.43 | 0.66 | 0.73 | 0.79 | 0.79 | 0.64 | 0.79 | 0.55 | 0.49 | 0.54 |
| | **T1+2 EN** | 0.54 | 0.57 | 0.51 | 0.79 | 0.93 | 0.74 | 0.74 | 0.66 | 0.85 | 0.66 | 0.66 | 0.42 |
| | **T1+2 Lasso** | 0.54 | 0.56 | 0.49 | 0.66 | 0.85 | 0.7 | 0.69 | 0.64 | 0.86 | 0.71 | 0.66 | 0.42 |
| W/ TsFresh | **T1 EN** | 0.49 | 0.38 | 0.42 | 0.69 | 0.69 | 0.79 | 0.78 | 0.67 | 0.82 | 0.5 | 0.48 | 0.54 |
| | **T1 Lasso** | 0.47 | 0.38 | 0.42 | 0.71 | 0.69 | 0.8 | 0.77 | 0.67 | 0.81 | 0.52 | 0.46 | 0.46 |
| | **T1+2 EN** | 0.54 | 0.54 | 0.52 | 0.69 | 0.95 | 0.69 | 0.72 | 0.72 | 0.98 | 0.69 | 0.58 | 0.76 |
| | **T1+2 Lasso** | 0.57 | 0.55 | 0.52 | 0.67 | 0.8 | 0.69 | 0.73 | 0.69 | 0.88 | 0.7 | 0.64 | 0.72 |

submitting, we ran Grid search using cross validation on entire training data (without the test split) and the model that gave the best validation set RMSE was selected for that question.

Using the above methodology, our final model was **ElasticNet + Naive** model. This achieved an RMSE of **0.5048** and MAE of **0.2222** on the final unseen test set and the leaderboard. This model was just shy of the Naive model submission that we made which had an RMSE of **0.4912** and MAE of **0.2024**.

Our LSTM model produced an RMSE of **0.8249** and MAE of **0.4761**, which, unfortunately, was not as strong as the naive model or the ElasticNet/Lasso. This could be attributed to several factors, including the relatively small size of the dataset, which may have hindered the model's ability to capture complex temporal patterns. Additionally, the high variability in ALS progression among patients and potential noise in the sensor data could have impacted the model's performance.




(a) Task1 Q7 (b) Task1 TsFresh Q6 (c) Task2 Q5

**Figure 5:** Feature Importance charts showing "previous_value" is consistently the most important feature.

### 4.2. Task2

Due to time constraints, we didn't manage to submit our model for Task2. However, using the ground truth that was released after the competition deadline, we ran our ElasticNet + Naive model and it gave an RMSE of **0.5594** and MAE of **0.2803**. This is better than the top leaderboard submission with RMSE of 0.5774 and MAE of 0.2879.

### 4.3. Takeaways

Following is a summary of our takeaways:

- **Previous score** is an overwhelming contributor in prediction of the future score
- No single model could capture the essence of **all questions** better than the Naive model
- Small training set means that simpler models that focus on **regularization** perform well
- Having more granularity in scores either through having **floating point scores** or **larger span like [0, 10]** rather than [0, 4] would have allowed better modeling results, as 5 integer scale for scores seemed to mask inherent trends.
- Ablating **sensor data** in training didn't deteriorate the model performance much, which is evident from the Feature Importance scores graph in Figure 5
- **Notable features** that showed up in Feature Importances (other than previous_value) across multiple questions include:
  - static features: age_at_diagnosis and FVC
  - median sensor features: agg_respiration_alpha2_DimMean, agg_respiration_RMSSD, agg_respiration_SD1 and agg_total_steps
  - `tsfresh` sensor features: beat_to_beat_cvsd__benford_correlation, beat_to_beat_cvsd__quantile__q_0.6 and total_steps__quantile__q_0.1

## 5. Future Work

Future work will focus on clustering patient data to capture distinct patient phenotypes, allowing for more personalized predictions and treatment plans. We also plan to utilize freely available datasets such as PRO-ACT to enhance the robustness and generalizability of our models. Additionally, access to multimodal data sources such as audio data for speech, accelerometer data for muscle function, genetic data, imaging data, etc could help further improve the accuracy and reliability of ALSFRS-R score predictions. These efforts aim to refine our models and contribute to more effective ALS management and patient care.

Another point we plan to work on is integrating models for panel data. We've already experimented with this approach, where instead of fitting separate models for each question, we fitted a panel model

on the whole data and defined the grouping variable as the question, the patient, or a nested combination of both. The preliminary results from this approach were worse than the “per-question modeling" approach. However, we believe that following a method similar to that of [10] could still be beneficial. Such a methodology would make use of our existing models for the fixed effects part and augment them with random effects, e.g by treating the question as the grouping variable.

## 6. Conclusions

We explored various machine learning approaches to predict ALSFRS-R scores. The main goal of our study was to determine if the app data provides additional value that enhances the predictive accuracy of the models, based on evidence found in the dataset that we analyzed.

The methodology involved extensive data preprocessing to synchronize clinical and sensor data, ensuring the integrity and reliability of the dataset. We generated features from both static and dynamic data sources, with a particular focus on leveraging temporal dependencies in the sensor data through techniques like median aggregation, feature-based time series analysis, and LSTM neural networks.

Our comparative analysis revealed that while on Task1 the naive baseline model achieved slightly better predictive accuracy, the ElasticNet+Naive regression model offered a robust framework for understanding feature contributions. Moreover, on Task2 our model (though not submitted) was able to perform slightly better than the Naive model. Nonetheless, the previous value of the ALSFRS-R score emerged as a significant predictor across all models. Thus, our analysis did not find significant evidence that the app data provides an overall enhancement to the predictive accuracy of the model. Based on the leaderboard at the end of the competition, it would appear that the other teams reached a similar conclusion.

We caution against discarding the potential usefulness of the app data at this stage however, as the dataset brought several methodological challenges with it, the biggest of which was its very small size. We addressed it through rigorous cross-validation strategies and data augmentation techniques, ensuring that our models generalize well to unseen patients. The feature importance analysis highlighted and added value of certain static features and, for some questions, the importance of isolated sensor-derived features.

Overall, this work demonstrates the potential of integrating sensor data with machine learning models to enhance the monitoring and prediction of ALS progression, within the limits of a small dataset.

Given the strong persistence of the “previous value" in both our modeling, and the lack of significant improvement to the baseline model, we hypothesise that the dataset would benefit from a larger and more heterogeneous set of patients, not only to increase the amount of observations, but to also introduce more variability in the target variable, perhaps spread across more regions and with clinical evaluations done by a multitude of clinicians.

## Acknowledgements

The authors would like to extend their sincere gratitude to Dr. Jay Summet and the DS@GT CLEF team at Georgia Tech for their invaluable support and insights.

## References

[1] G. Birolo, P. Bosoni, G. Faggioli, H. Aidos, R. Bergamaschi, P. Cavalla, A. Chiò, A. Dagliati, M. de Carvalho, G. Di Nunzio, P. Fariselli, J. García Dominguez, A. G. Marta Gromicho, E. Longato, S. Madeira, U. Manera, S. Marchesin, L. Menotti, G. Silvello, E. Tavazzi, E. Tavazzi, I. Trescato, M. Vettoretti, B. D. Camillo, N. Ferro, Overview of iDPP@CLEF 2024: The Intelligent Disease Progression Prediction Challenge, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France, September 9th to 12th, 2024, 2024.

[2] G. Birolo, P. Bosoni, G. Faggioli, H. Aidos, R. Bergamaschi, P. Cavalla, A. Chiò, A. Dagliati, M. de Carvalho, G. Di Nunzio, P. Fariselli, J. García Dominguez, A. G. Marta Gromicho, E. Longato, S. Madeira, U. Manera, S. Marchesin, L. Menotti, G. Silvello, E. Tavazzi, E. Tavazzi, I. Trescato, M. Vettoretti, B. D. Camillo, N. Ferro, Intelligent Disease Progression Prediction: Overview of iDPP@CLEF 2024, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9-12, 2024, Proceedings, 2024.

[3] A. S. Gupta, S. Patel, A. S. Premasiri, F. G. Vieira, At-home wearables and machine learning sensitively capture disease progression in amyotrophic lateral sclerosis, Nature Communications 14 (2023). URL: https://api.semanticscholar.org/CorpusID:261062809.

[4] F. G. Vieira, S. Venugopalan, A. S. Premasiri, M. McNally, A. Jansen, K. McCloskey, M. P. Brenner, S. Perrin, A machine-learning based objective measure for als disease severity, NPJ Digital Medicine 5 (2022). URL: https://api.semanticscholar.org/CorpusID:246240960.

[5] C. Pancotti, G. Birolo, C. Rollo, T. Sanavia, B. Di Camillo, U. Manera, A. Chiò, P. Fariselli, Deep learning methods to predict amyotrophic lateral sclerosis disease progression, Scientific reports 12 (2022) 13738.

[6] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, J. Sun, Doctor ai: Predicting clinical events via recurrent neural networks, JMLR workshop and conference proceedings 56 (2015) 301–318. URL: https://api.semanticscholar.org/CorpusID:5842463.

[7] G. Vidal, J. Sharpnack, P. Pinedo, I. C. Tsai, A. R. Lee, B. Martínez-López, Comparative performance analysis of three machine learning algorithms applied to sensor data registered by a leg-attached accelerometer to predict metritis events in dairy cattle, volume 4, Frontiers, 2023, p. 1157090.

[8] N. F. Smedley, B. M. Ellingson, T. F. Cloughesy, W. Hsu, Longitudinal patterns in clinical and imaging measurements predict residual survival in glioblastoma patients, Scientific Reports 8 (2018). URL: https://api.semanticscholar.org/CorpusID:52846175.

[9] J. Pinheiro, D. Bates, Mixed-effects models in S and S-PLUS, Springer science & business media, 2006.

[10] A. Hajjem, F. Bellavance, D. Larocque, Mixed-effects random forest for clustered data, Journal of Statistical Computation and Simulation 84 (2014) 1313–1328.

[11] A. Cascarano, J. Mur-Petit, J. Hernandez-Gonzalez, M. Camacho, N. de Toro Eadie, P. Gkontra, M. Chadeau-Hyam, J. Vitria, K. Lekadir, Machine and deep learning for longitudinal biomedical data: a review of methods and applications, Artificial Intelligence Review 56 (2023) 1711–1771.

[12] S. Hu, Y.-G. Wang, C. Drovandi, T. Cao, Predictions of machine learning with mixed-effects in analyzing longitudinal data under model misspecification, Statistical Methods & Applications 32 (2023) 681–711.

[13] T. Wörtwein, N. B. Allen, L. B. Sheeber, R. P. Auerbach, J. F. Cohn, L.-P. Morency, Neural mixed effects for nonlinear personalized predictions, in: Proceedings of the 25th International Conference on Multimodal Interaction, 2023, pp. 445–454.

[14] B. D. Fulcher, Feature-based time-series analysis, in: Feature engineering for machine learning and data analytics, CRC press, 2018, pp. 87–116.

[15] M. Christ, N. Braun, J. Neuffer, A. W. Kempa-Liehr, Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package), Neurocomputing 307 (2018) 72–77.

[16] M. Christ, A. W. Kempa-Liehr, M. Feindt, Distributed and parallel time series feature extraction for industrial big data applications, CoRR abs/1610.07717 (2016). URL: http://arxiv.org/abs/1610.07717. arXiv:1610.07717.

[17] K. Pearson, X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 50 (1900) 157–175. URL: https://doi.org/10.1080/14786440009463897. doi:10.1080/14786440009463897.

[18] G. C. Cawley, N. L. Talbot, On over-fitting in model selection and subsequent selection bias in performance evaluation, The Journal of Machine Learning Research 11 (2010) 2079–2107.

[19] M. Kuhn, K. Johnson, Feature engineering and selection: A practical approach for predictive models, Chapman and Hall/CRC, 2019.

[20] V. Lyashenko, A. Jha, Cross-validation in machine learning: How to do it right, 2024. URL: https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right, accessed: 2024-05-24.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[22] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.