**Cross validation**

We are measuring our models with the accuracy metric. In order to compare the results from different models, we use stratified cross validation with 10 folds or the average of ten stratified train-test splits. We noted that the stratified approaches reduce the variance of the estimate compared to unstratified methods.

**Gradient boosting**

Gradient boosting is a recent machine learning technique used for regression and classification tasks based on ensemble learning. Many weak prediction models are averaged and result in a strong prediction model. XGBoost is a gradient boosting method based on decision trees. It has gained popularity because of its very good performance in machine learning competitions **[[1]](#footnote-1)**. XGBoost also impresses with its high computing speed.

We use the XGBoost library for predicting the quality of battery contacting and to compare the performance to the other used models (neural network and logistic regression). In a first step, we preprocessed our data. Our features are represented by each 112 signal values from 2 sensors as input and our label consists of the respective classification goal. Data from sensor 1 was already denoised. We used wavelets to denoise the signals from sensor 2 in MATLAB. We then conducted a train-test split with the typical split ratio of 70/30 to train the model. For better comparability of our models, to avoid overfitting and to get the best indicator for our overall performance, we performed a stratified cross validation with 10 folds. In order to improve our hyperparameters and the result, a randomized (not exhaustive) grid search was conducted. Furthermore, we performed a principal component analysis to improve in calculation speed and complexity. Table 1 shows our results for the different classification tasks and our best performing feature combinations. “Sensor 1dn” are the denoised signals from sensor 1. “Sensor 2MAT” are the manually denoised signals with MATLAB from sensor 2.

Table 1: Comparison of results with different features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **y** | | | |
| **X** | NOK | WD40 | Gleitmo | WD 40 or Gleitmo |
| **Sensor 1dn** | 0.902 | 0.612 | 0.699 | 0.641 |
| **Sensor 1**  **Sensor 1dn**  **Sensor 2MAT** | 0,921 | 0.633 | 0.701 | 0.673 |

Since we only have limited calculation resources, we performed a not exhaustive grid search for hyperparameter optimization. In order to obtain the best results, it would be beneficial to perform an exhaustive grid search in a next step. The performance of our estimator would also benefit from further feature engineering approaches and more train data.

1. https://www.kaggle.com/code/dansbecker/xgboost/notebook [↑](#footnote-ref-1)