# Automated Machine Learning for Remaining Useful Life Predictions
# Online Appendix

## A Search Space Description

The search space $\Lambda$ of AUTORUL allows the creation of 624 unique pipelines, which can be further configured by a total of 168 hyperparameters. In the following tables, more details regarding the available algorithms and hyperparameters are provided. For each algorithm, the total number of hyperparameters (#$\lambda$), the number of categorical (cat) and numerical (num) hyperparameters, is given. Numbers in parentheses denote conditional hyperparameters. The total number of hyperparameters does not add up to the reported 168 hyperparameters as hyperparameters of some algorithms, like, for example, TS Fresh or window generation, are included twice (once for tabular regression and once for sequence-to-sequence regression using neural networks). Components highlighted in italic are not directly included in the pipeline but influence the fitting procedure of the pipeline.

Table 1: Preprocessing Algorithms

| Name | #$\lambda$ | cat | num |
|---|---|---|---|
| Imputation | 1 | 1 (0) | 0 (0) |
| Exponential Smoothing | 2 | 0 (0) | 2 (0) |
| Robust Scaler | 2 | 0 (0) | 2 (0) |
| Normalizer | 0 | 0 (0) | 0 (0) |
| Min Max Scaling | 0 | 0 (0) | 0 (0) |
| Standardizer | 0 | 0 (0) | 0 (0) |

Table 2: Feature Engineering Algorithms

| Name | #$\lambda$ | cat | num |
|---|---|---|---|
| Window Generation | 2 | 0 (0) | 2 (0) |
| Flattening | 0 | 0 (0) | 0 (0) |
| TS Fresh | 43 | 43 (0) | 0 (0) |
| PCA | 2 | 1 (0) | 1 (0) |
| Select Percentile | 2 | 1 (0) | 1 (0) |
| Select Rates | 3 | 2 (0) | 1 (0) |

Table 3: Tabular Regression Algorithms

| Name | #$\lambda$ | cat | num |
|---|---|---|---|
| Extra Trees | 5 | 2 (0) | 3 (0) |
| Gradient Boosting | 6 | 0 (0) | 6 (0) |
| MLP | 6 | 2 (0) | 4 (1) |
| Passive Aggressive | 4 | 2 (0) | 2 (0) |
| Random Forest | 3 | 1 (0) | 2 (0) |
| SGD | 6 | 2 (0) | 4 (1) |

Table 4: Sequence Regression Algorithms

| Name | #$\lambda$ | cat | num |
|---|---|---|---|
| *Optimizer* | 4 | 0 (0) | 4 (0) |
| *Trainer* | 2 | 0 (0) | 2 (0) |
| CNN | 5 | 1 (0) | 4 (1) |
| GRU | 4 | 1 (0) | 3 (1) |
| LSTM | 4 | 1 (0) | 3 (1) |
| TCN | 5 | 1 (0) | 4 (1) |

# B Detailed Experimental Results

Besides the results reported in the manuscript, we want to provide further insights into the generated pipelines. Figure 1 contains visualizations of the final performances reported in *Table 1* of the manuscript per benchmark dataset.

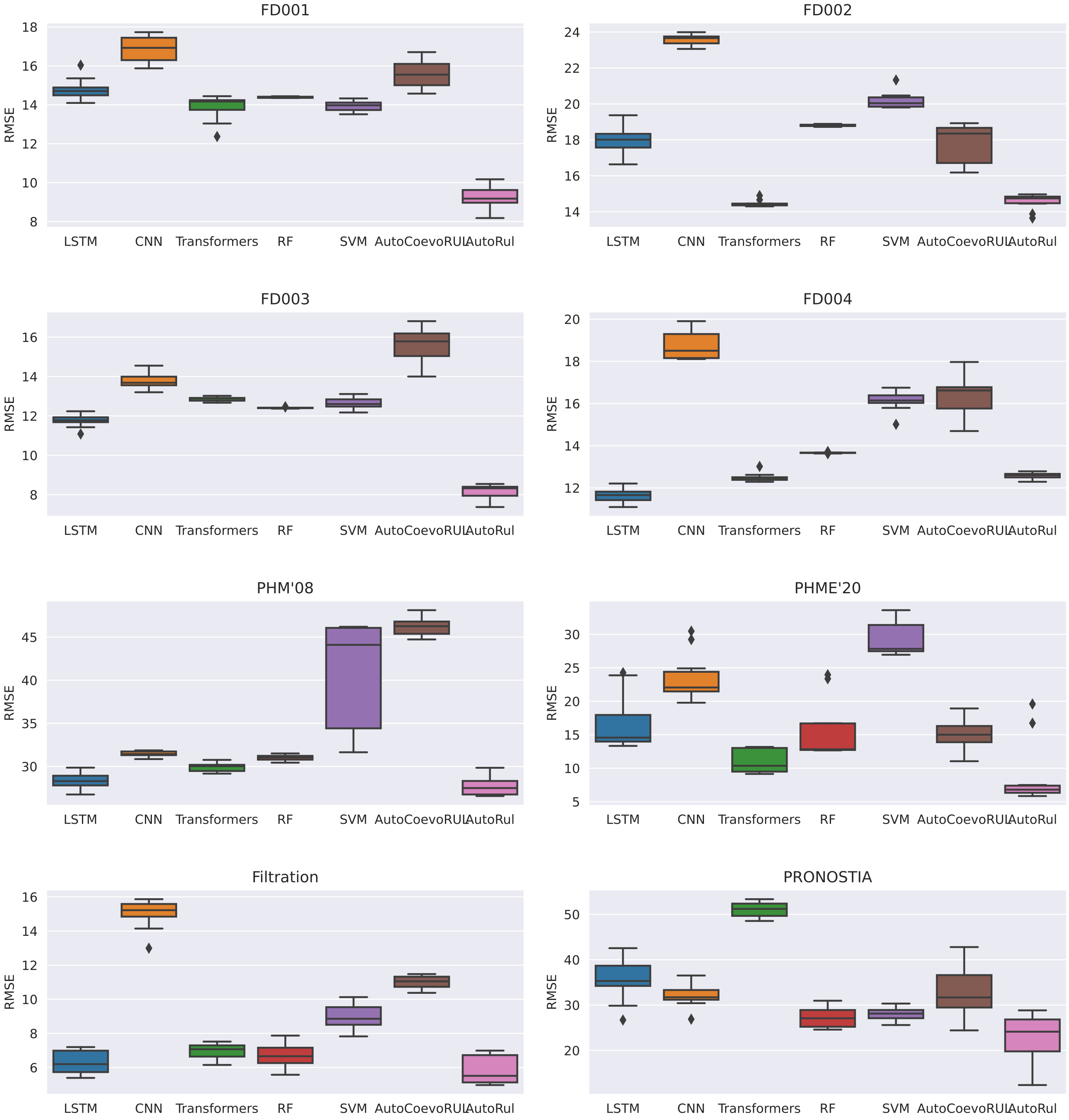

Figure 1: Performance visualizations of all tested RUL methods on all benchmark datasets.

Next, we take a closer look at the behaviour of AUTORUL during the optimization. Table 5 contains statistics about the number of evaluated configurations for each dataset. Reported are the total number of evaluated configurations, the number of successful evaluations, the number of failed evaluations (for example due to an exception during model fitting), the number of configurations where fitting was canceled after five minutes, and the number of configurations that were trained on the complete training data (no multi-fidelity approximation). In general, the vast majority of the evaluated configurations was fitted successfully. Less than 3% of the evaluated configurations were aborted after the configured timeout of five minutes, indicating that the limit was not selected to aggressively. Even though a large number of different configurations was evaluated for each dataset, only roughly 5% of the configurations were evaluated on the full budget, i.e., fully fitted until convergence.

Table 5: Statistics about the evaluated configurations. Results are averaged over ten repetitions.

| Dataset | # Configurations | # Success | # Failed | # Timeout | # Full Budget |
|---|---|---|---|---|---|
| FD001 | 761.8 ± 144.48 | 739.1 ± 135.70 | 8.5 ± 9.92 | 14.2 ± 6.71 | 41.3 ± 7.93 |
| FD002 | 932.5 ± 382.89 | 897.2 ± 391.24 | 9.1 ± 7.71 | 26.2 ± 14.28 | 50.2 ± 21.61 |
| FD003 | 648.8 ± 62.75 | 631.9 ± 65.17 | 3.3 ± 1.55 | 13.6 ± 5.64 | 35.4 ± 3.32 |
| FD004 | 991.9 ± 462.97 | 923.0 ± 356.89 | 46.7 ± 5.83 | 22.2 ± 5.83 | 53.2 ± 26.46 |
| PHM'08 | 355.0 ± 26.65 | 323.1 ± 25.04 | 28.5 ± 0.92 | 28.5 ± 4.34 | 18.7 ± 1.55 |
| PHME'20 | 886.9 ± 125.11 | 875.0 ± 122.06 | 8.0 ± 2.47 | 8.0 ± 4.90 | 48.4 ± 6.48 |
| Filtration | 818.8 ± 134.86 | 799.0 ± 137.68 | 5.7 ± 3.29 | 14.1 ± 10.09 | 44.7 ± 7.87 |
| PRONOSTIA | 471.5 ± 79.13 | 354.7 ± 76.85 | 86.8 ± 12.84 | 30.0 ± 9.15 | 25.3 ± 4.50 |

Finally, we take a closer look at the pipelines constructed by AUTORUL. Table 6 provides an overview of the constructed ensembles and the pipelines in them for each of the benchmark datasets. The *ensemble size* column shows the average number of pipelines in the constructed ensemble. It is apparent that the option to build ensembles is used for all datasets but the maximum ensemble size of ten pipelines is not consistently reached. Regarding the used template, both sequence-to-sequence regression and tabular regression are used. Depending on the dataset, either one of the two options can be basically used exclusively or both options can be mixed together. Similarly, for feature engineering all three available algorithms are extensively used with TS Fresh being used the most. Finally, for most datasets a specific type of regression algorithm is used significantly more often than others. Only for the PHM'08 dataset four different algorithm types are used roughly equally often. Random forests and TCNs are by far the most used regression algorithms. In contrast, SGD and extra trees are not included in any of the final ensembles and can probably pruned from the search space.

Table 6: Overview of constructed pipelines for each benchmark dataset. Numbers in parentheses represent the fraction of pipelines using the according component. The sum of fractions within one cell can be unequal to 1.00 due to rounding errors.

| Dataset | Ensemble Size | Template | Feat. Eng. | Regressor |
|---|---|---|---|---|
| FD001 | 8.6 ± 0.92 | seq2seq (1.0) | Flatten (0.35)<br>Limited (0.01)<br>TS Fresh (0.64) | CNN (0.20) GRU (0.03)<br>LSTM (0.02) TCN (0.74) |
| FD002 | 6.0 ± 1.84 | seq2seq (0.15)<br>Tabular (0.85) | Flatten (0.12)<br>Limited (0.67)<br>TS Fresh (0.22) | Gradient Boosting (0.13)<br>Random Forest (0.72)<br>GRU (0.03) LSTM (0.02)<br>TCN (0.10) |
| FD003 | 7.8 ± 1.17 | seq2seq (0.95)<br>Tabular (0.05) | Flatten (0.31)<br>Limited (0.08)<br>TS Fresh (0.61) | Gradient Boosting (0.01)<br>Random Forest (0.04)<br>CNN (0.10) GRU (0.01)<br>LSTM (0.03) TCN (0.81) |
| FD004 | 5.9 ± 1.22 | seq2seq (0.15)<br>Tabular (0.85) | Flatten (0.16)<br>Limited (0.36)<br>TS Fresh (0.47) | Gradient Boosting (0.03)<br>Random Forest (0.81)<br>CNN (0.07)<br>TCN (0.08) |
| PHM'08 | 5.8 ± 1.83 | seq2seq (0.74)<br>Tabular (0.26) | Flatten (0.18)<br>Limited (0.54)<br>TS Fresh (0.30) | Passive Aggressive (0.02)<br>Random Forest (0.24)<br>CNN (0.33) GRU (0.16)<br>LSTM (0.05) TCN (0.21) |
| PHM'20 | 7.2 ± 1.17 | seq2seq (0.99)<br>Tabular (0.01) | Flatten (0.05)<br>Limited (0.04)<br>TS Fresh (0.90) | Passive Aggressive (0.01)<br>CNN (0.01) GRU (0.36)<br>LSTM (0.33) TCN (0.28) |
| Filtration | 5.8 ± 1.83 | seq2seq (0.14)<br>Tabular (0.86) | Flatten (0.10)<br>TS Fresh (0.90) | Gradient Boosting (0.03)<br>Random Forest (0.83)<br>GRU (0.07) LSTM (0.03)<br>TCN (0.03) |
| PRONOSTIA | 4.5 ± 1.12 | Tabular (1.00) | Flatten (0.24)<br>Limited (0.27)<br>TS Fresh (0.49) | Gradient Boosting (0.16)<br>MLP (0.18)<br>Passive Aggressive (0.18)<br>Random Forest (0.49) |