Predicting remaining useful life of aircrafts' turofan engines

Fernando Martinelli Ramacciotti
fernandoramacciotti@gmail.com

Abstract

This electronic document is a template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document.

I. Introduction

Companies are always interested in cost reduction opportunities, since it may enables greater profits. Traditionally, companies schedule maintenance of their assets on a regular basis and then assess the need for replacement or fixing. In short, companies rely on corrective maintenance and actions are only taken when the asset is already degraded or failed. Predictive Maintenance, thus, aims to accurately provide remaining useful life (RUL) of assets so that businesses can prepare in advance when such asset comes to fail. Such prepared action can be replacement, maintenance, money borrowing and any other strategy that will minimize the loss of the to be faulty asset in question.

In this study, we used data-driven methods to evaluate RUL predictions, such as statistical and machine learning models. Such approaches rely less on domain knowledge and more on statistical patterns of the data in hand. Surely, statistical models combined with experts' inputs usually generates great results.

In order to have a good data-driven model, one must have quality data - and this is the bottleneck of such approach. Good data for predictive maintenance means that we have enough run-to-failure data from multiple assets so that our model can extract patterns. However, such data is costly to collect, once companies would lose money to run their machines to failure, even though they would be able to build a model to prevent it in the future. Luckily, NASA made available a simulated run-to-failure dataset for a fleet of turbofan engines [1], [11], originally used in Prognostics and Health Management competition of 2008 (PHM'08). Since then, the dataset have been extensively used for benchmarking data-driven models that predict RUL. A thorough review can be found in [8], [7]. Other approaches can be found in [3], [5], [6], [10], including winning models from the very same competition [4], [12].

There are four datasets, that differ on operational conditions and fault modes. In this study we used the most complex one: 248 train and 249 units (or engines), with six different operational conditions and two fault modes, alongside with noisy measurements. Each sample is considered a cycle and we have the sensor measurements. All training units were simulated-to-failure, while the test series are cut off sometime before failure. Our target function, therefore, is to predict the implicit RUL. We assume that after each cycle the RUL is reduced by one unit - therefore if a train unit has 200 samples, then the last point has RUL zero and the first sample a RUL of 200.

The key to a good model here is to preprocess the data. A raw plot of sensor data shows no clear trends over time and intermittent value shifts. A careful deep-dive revealed that such jumps were due to changes in operational condition, i.e. for each operational condition the sensors operates at different levels. We have, therefore, clustered the series into 6 cluster using k-means algorithm and standardized the series to zero mean and unit variance per cluster - and removed features that were constant within at least one cluster. It was interesting to notice that some sensors drift upwards when the engine fails, others downwards, while others do not show clear patterns. In addition, we have created features to cumulative count how many cycles each operational condition was set at any given time.

We have used two types of learning algorithms: regression and classification. Regression models continuously estimate RUL while for classification the problem statement would be *is the RUL less than an arbritrary value, e.g. 20?* or *will the engine fail at some time for the next upcoming 20 cycles?*. For regression we used Linear Regression as a baseline model and compared with two more complex machine learning approaches: Random Forest and Gradient Boosting regressors. For classification, we were interesting in predict if an engine has less than 20 RUL. We used Logistic Regression as baseline and, again, compared against Random Forest and Gradient Boosting (now as classifiers). We divided our training set into 5 different folds for cross-validation. Gradient Boosting and Random Forest, for regression and classification, respectively, performed better, i.e., had greater mean cross-validated scores - Mean Squared Error for regression and F_1 Score for classification. Such models were fine tuned, again cross-validation scoring, varying hyperparameters. All models were developed in Python.

The best mean cross-validation training score for the regression model scored an MSE of $2,745$ and test MSE of $4,781$. After postprocessing the prediction with Kalman filter [9], the test scored improved by 2%, down to $4,699$. Still, both training and test scores of the benchmark model, Linear Regression, were worse than the selected model.

The best mean cross-validation training score for the classification model scored an F_1 score of 0.83 and test MSE of 0.52. rSurp,singlyi the benchmark model, Logistic Regression, performed better on test set, with a F_1 test score of 0.55, even if a worse training score than the selected model.

In short, our models performed well on training set, but, even though we have used cross-validation folds to avoid overfitting, it seems that they could not generalize well, since the test results were much worse. Perhaps a more thorough

preprocessing improves model performance, with a more careful signal processing to remove noise. A custom cost function for model optimization could also be tested, penalizing RUL overestimation more, since it is more critical (i.e. late predicting is more critical than early). In fact, the loss function used in the original competition was asymmetrical to account such preference [11]. Moreover, different target function could be tested, treating RUL more as a health index - the motivation is because the sensors operates at a fairly constant range within each operational mode while the engine is healthy and only deviates when unhealthy.

This paper is organized as follows: section II introduces and describes the dataset; section III outlines the data preprocessing steps; section IV depicts the models used; section V describes the evaluation criteria to model selection; section VI presents the postprocessing step experimented on regression outputs; section VII discusses the results; and final remarks are in section VIII.

II. Data

The dataset used is from NASA's repository: Turbofan Engine Degradation Simulation Data Set. The data is described as follows:

"Data sets consists of multiple multivariate time series. Each data set is further divided into training and test subsets. Each time series is from a different engine, i.e., the data can be considered to be from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation is considered normal, i.e., it is not considered a fault condition. There are three operational settings that have a substantial effect on engine performance. These settings are also included in the data. The data is contaminated with sensor noise.

"The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective of the competition is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true Remaining Useful Life values for the test data.

"The data are provided as a zip-compressed text file with 26 columns of numbers, separated by spaces. Each row is a snapshot of data taken during a single operational cycle, each column is a different variable."[1]

The columns correspond to:

- unit number;
- time, in cycles;
- operational settings (3 columns);
- sensor measurements (21 columns).

More details about the simulation and damage propagation modeling can be found in [11]. It is worth noticing that such data was used in the PHM'08, the Prognostics and Health Management competition of 2008, where participants were challenged to create algorithms to predict RUL.

We have four different files:

- FD001: 100 train samples and 100 test samples, one operational condition and one fault mode;
- FD002: 260 train samples and 259 test samples, six operational conditions and one fault mode;
- FD003: 100 train samples and 100 test samples, one operational condition and two fault modes;
- FD004: 249 train samples and 248 test samples, six operational conditions and two fault modes;

The dataset used in this study is the *FD004*. A quick look at how our target variable, RUL, is distributed, depicted in Figure 1, reveals a right skewed distribution. In Table I the reader can find the variable statistical descriptions.

Statistic	Value
N	249
Mean	244.98
Std. Dev.	73.11
Min	127
Max	542

TABLE I
TRAIN RUL SUMMARY

[1] https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository

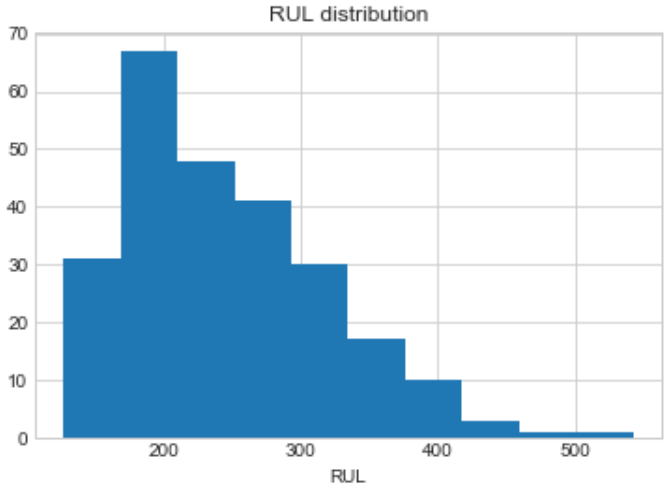

Fig. 1. Training RUL distribution

III. Preprocessing

Raw sensor data plotting reveals no clear patterns and insights about how each sensor trends over time until failure, as depicted in Figure 2. A visual inspection of the three operational settings variables indicates in [FIGURA OP SETTINGS] six different regions were samples agglomerate, which is expected given the six different operational regimes the engines can run. Back again to the raw sensor plotting over time, the intermittent behavior of the sensors measurement could be different operational regimes at each cycle - i.e., each operational mode shifts the operating range of each sensor and the operational conditions can be distinct at each cycle. Such hypothesis is supported by plots in Figure 3, where there are clear clusters for each sensor ranging from all possible RUL values, at least visually. Such scatter plots suggests that the operational condition can occur at any time of the engine, healthy or not, and changes intermittently. Clustering the series with kmeans algorithm, the clusters clearly emerge.

Given that sensors operates at different ranges and, on top of that, operational conditions set new ranges for each sensor, the next logical step is to standardize each sensor to zero mean and unit variance according to the correspond operational regime. We have also removed features that are constant within at least one regime, since it would not add any new information to our model. The remaining standardized sensor charts are shown in Figure 4, where is possible to see clear trends for some sensors over time. Sensors 1, 5, 16, 18 and 19 were removed since they were constant within at least one regime. Now it is clear that some sensors drift upwards and other downwards when the engines run towards failure.

Additionally, since regimes seem to affect operating ranges, we added custom features to cumulative count the number of cycles each operational mode was run per unit.

IV. Modeling

The formal derivations and construction of each algorithm used is out of the scope of this paper. For a more in-depth discussion about the underlying statistical assumptions for each model and applications, please refer to [2].

A. Regression

The idea behind using regression models is to generate a continuous output at any given input. Surely, with our data, the output, i.e. the RUL, is assumed discrete and the smallest step is one unit. Nevertheless, we did not postprocess our output to integer numbers.

The models selected here were: Linear Regression, Random Forest and Gradient Boosting. The first, linear regression, serve as a baseline model due its simplicity. However, one cannot underestimate the power of such simple model as we can give reliable, robust and, above all, interpretative estimations. Random Forest and Gradient Boosting are more complex models, that can also be used as classifiers as we did, that rely on ensemble of weak predictors. In addition, such models can capture non-linear relationships. They have tunable hyperparameter that are key to

B. Classification

The classification task in RUL-related problems is usually set to classify answer whether the asset is at failure, unhealthy or operating at critical range. It is also possible to set the problem to predict if a failure will occur within a predefined

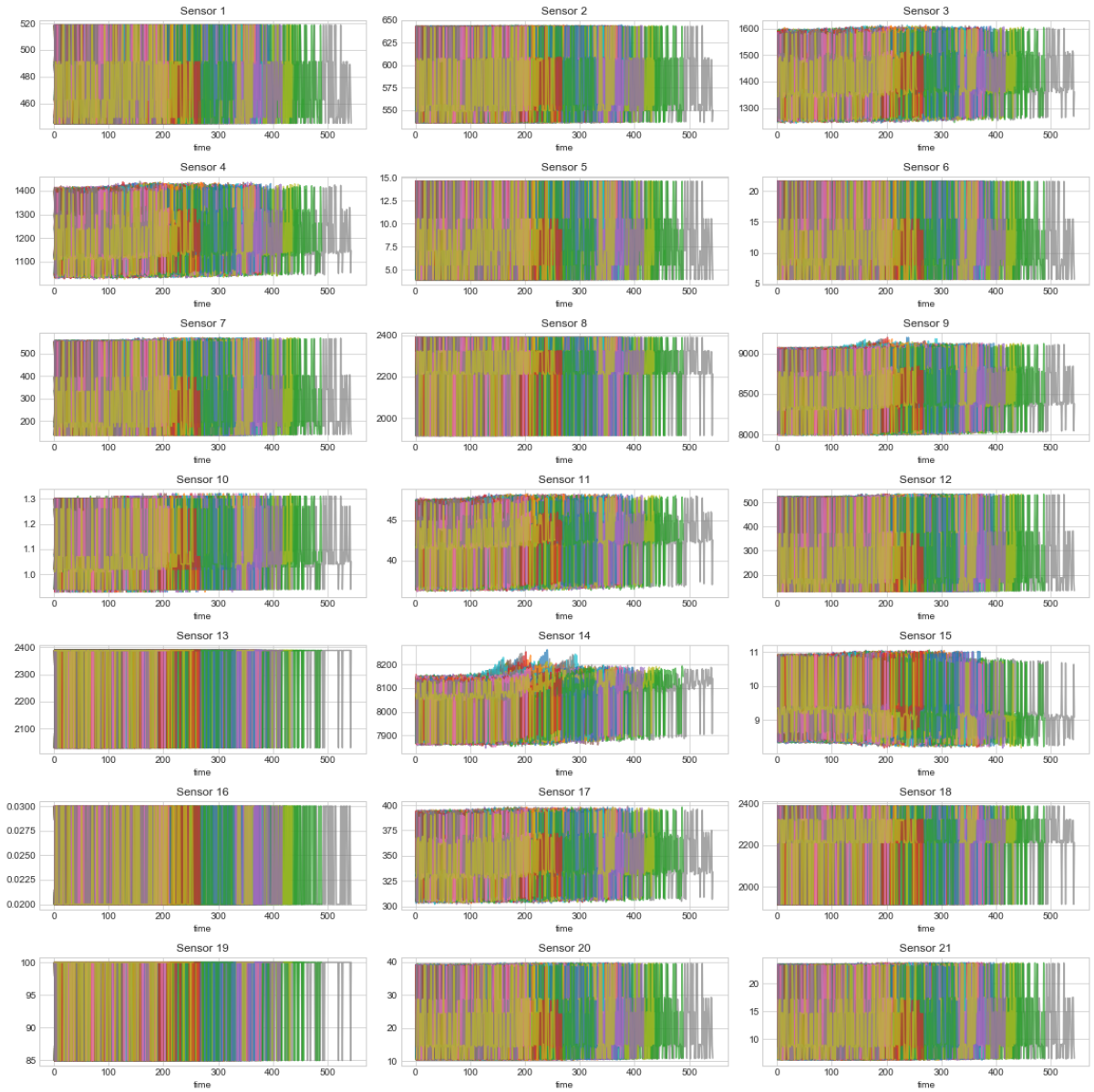

Fig. 2. Sensor measurements raw data. Intermittent series with no clear trends emerging over time

horizon - and that is how we modelled and labelled our data. We encoded our target variable, RUL as:

$$\text{RUL}_{\text{clf}} = \begin{cases} 0, & \text{if RUL ¿= 20} \\ 1, & \text{if RUL ¡ 20} \end{cases} \tag{1}$$

ance.

V. Model Selection

A. Evaluation Metrics

We evaluate our predictions using two types of metrics: Mean Squared Error (MSE) for regression models; and F_1 score for classification.

The MSE is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2, \tag{2}$$

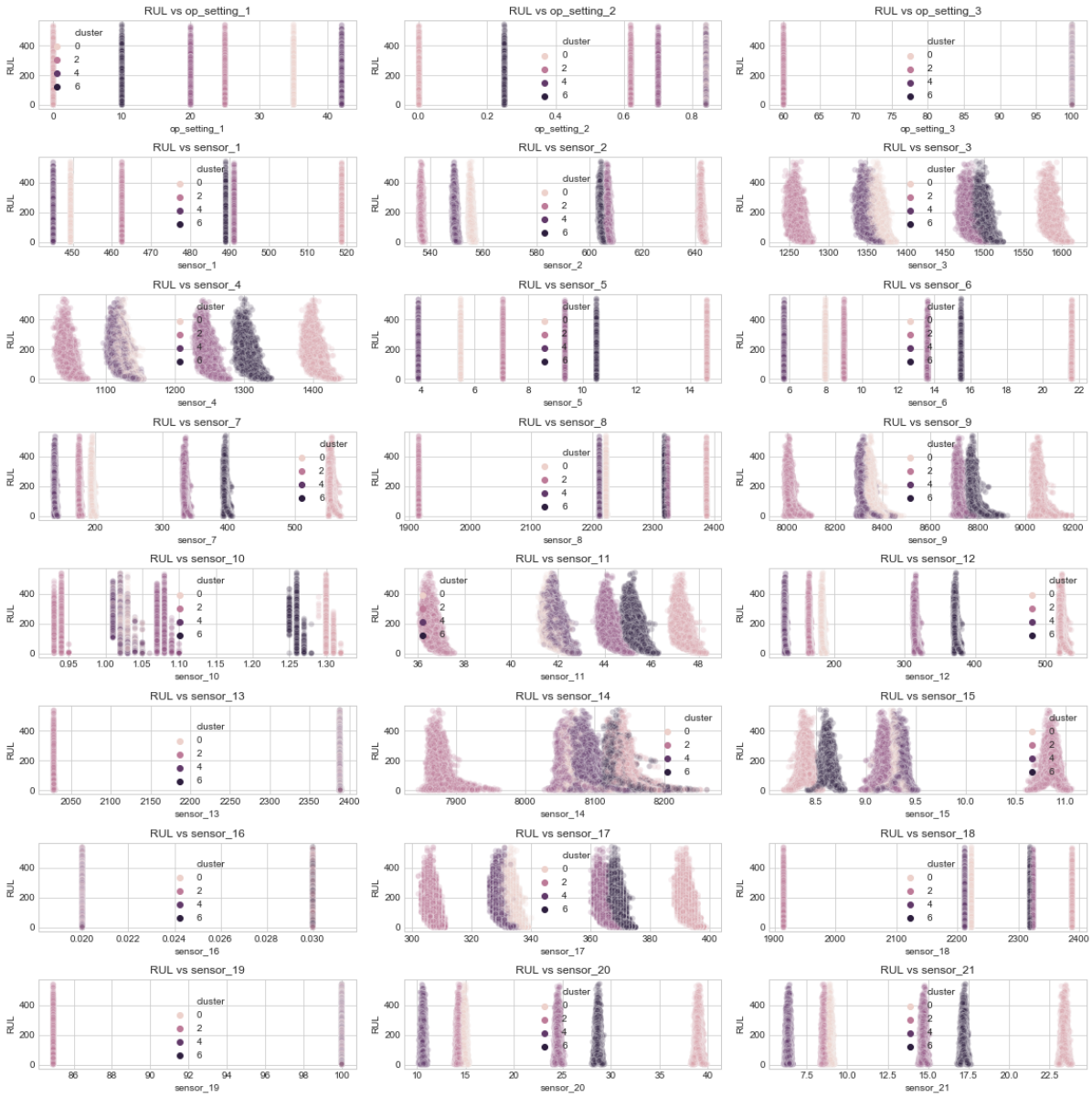

Fig. 3. RUL vs. each feature of the dataset. The six operational modes are clearly separable and identified by kmeans algorithm

where y is the true, or target, variable, i.e. RUL, $\hat{y}$ is the predicted RUL and N is the sample size, i.e. the number of predictions. Such metric is symmetric, i.e. equally weights over and underestimations and penalizes greater deviations. Therefore, the smaller, the better.

The F_1 Score here is defined as the harmonic mean of *precision* and *recall*, a special case of the more general definition of F_1:

$$\begin{aligned} F_1 &= \left(\frac{\text{precision}^{-1} + \text{recall}^{-1}}{2} \right)^{-1} \\ &= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} , \end{aligned} \tag{3}$$

where *precision* is the true positive over positive predictions and *recall* is the true positive over the true positive elements. Therefore, the greater, the better. In addition, it is clear that the possible values range from 0 to 1.

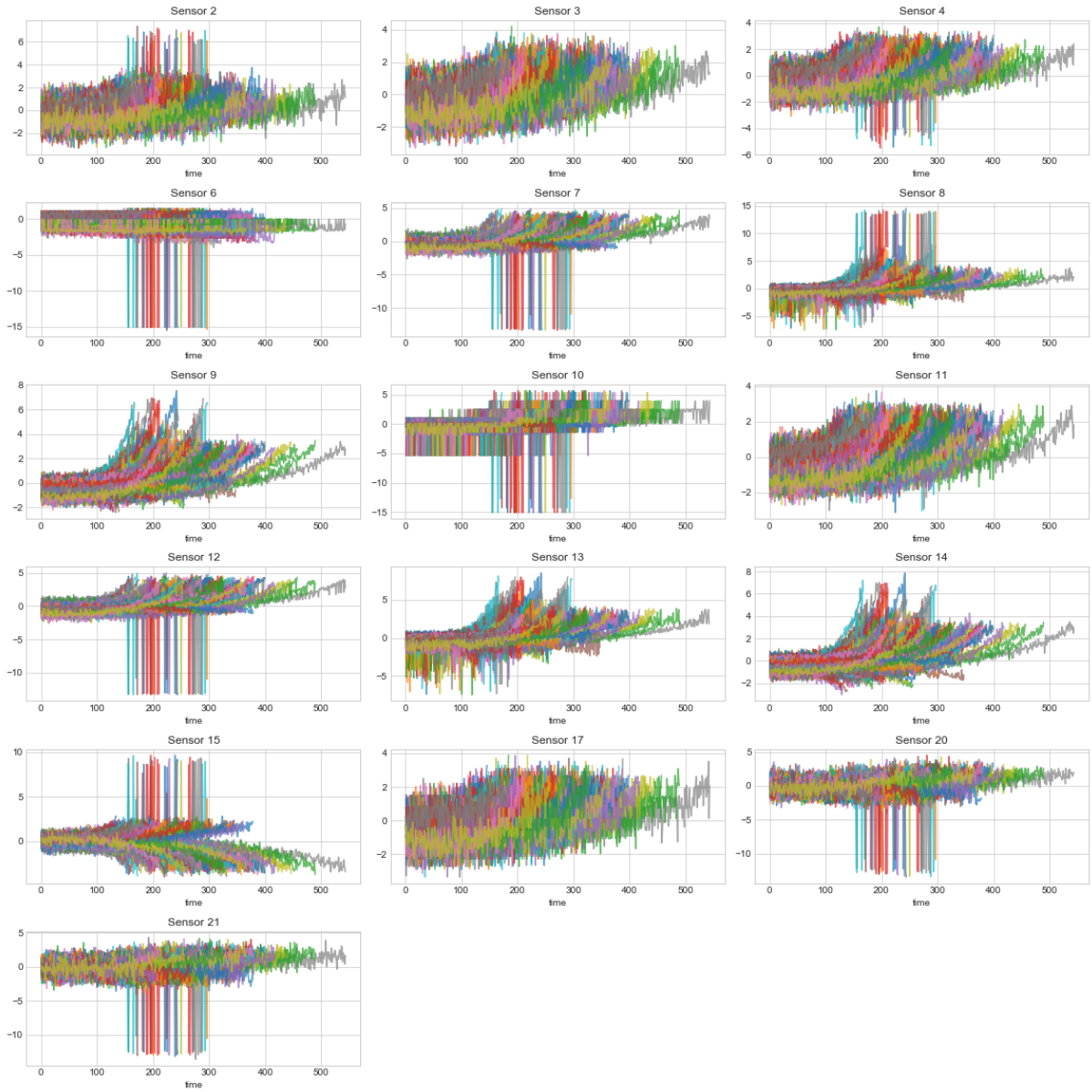

Fig. 4. Sensor measurements standardized for each regime. Now, clear trends over time emerges. Moreover, constant features within at least one regime were removed.

B. Cross-validation

C. Model selection

VI. Postprocessing

xxx

VII. Results

A. Regression

xxx

B. Classification

xxx

VIII. Conclusion

xxx

REFERENCES

[1] D. K. Frederick, J. A. DeCastro, and J. S. Litt, "User's guide for the commercial modular aero-propulsion system simulation (c-mapss)," 2007.

[2] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, NY, USA:, 2001, vol. 1, no. 10.

[3] N. Gugulothu, V. TV, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff, "Predicting remaining useful life using time series embeddings based on recurrent neural networks," *arXiv preprint arXiv:1709.01073*, 2017.

[4] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–6.

[5] C. Hu, B. D. Youn, P. Wang, and J. T. Yoon, "Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life," *Reliability Engineering & System Safety*, vol. 103, pp. 120–135, 2012.

[6] A. Mosallam, "A data-driven approach for remaining useful life prediction of critical components," 2014.

[7] E. Ramasso and A. Saxena, "Performance benchmarking and analysis of prognostic methods for cmapss datasets." *International Journal of Prognostics and Health Management*, vol. 5, no. 2, pp. 1–15, 2014.

[8] ——, "Review and analysis of algorithmic approaches developed for prognostics on cmapss dataset," in *Annual Conference of the Prognostics and Health Management Society 2014.*, 2014.

[9] H. Roark, "H2o machine learning and kalman filters for machine learning," H2O.ai, Tech. Rep., Jan. 2016.

[10] S. Sarkar, X. Jin, and A. Ray, "Data-driven fault detection in aircraft engines with noisy sensor measurements," *Journal of Engineering for Gas Turbines and Power*, vol. 133, no. 8, p. 081602, 2011.

[11] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–9.

[12] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–6.