Difference Normalization makes in Training a Model for NASA's Turbofan Engine Degradation Data Set

Azur Causevic 03692754
[1]Department of Electrical and Computer Engineering, Technical University of Munich
azur.causevic@tum.de
August 16, 2021

1 Introduction

In this article, we are going to discuss findings regarding the data set "Turbofan Engine Degradation Data Set" [1] provided by NASA. Research was highly inspired by the work done by "Smart Data Solution Center" [2] in Baden-Würtenberg. In model introduced by "SDSC" [2], sigmoid transformation was used to process extracted features later used for building a machine-learning model. Main goal of this research is to find out whether, and to which extent does normalization of extracted features from provided data set influence the final evaluation of the model.

2 Methods

2.1 Tsfresh[3]
Main package used for feature extraction was "tsfresh" [3]. It automatically calculates a large number of time series characteristics, so called "features". Further, package contains methods to evaluate the explaining power and importance of such characteristics for regression or classification tasks.

2.2 Scikit learn[4]
This package was used for data processing and evaluations of the built model. "Scikit learn" [4] is a simple and efficient tool for predictive data analysis. It supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.

2.3 LightGBM[5]
"LightGBM" [5] is a gradient boosting framework that uses tree-based learning algorithms. It is designed with following advantages: faster training speed and higher efficiency, lower memory usage, better accuracy, capability of handling large-scale data.

2.4 Tqdm[6]
"tqdm" [6] uses smart algorithms to predict the remaining time and to skip unnecessary iteration displays. In our case, it was used to track the progress during the feature-extraction process.

2.5 Matplotlib[7]
"Matplotlib" [7] is a comprehensive library for creating static, animated and interactive visualizations in Python. In this research, several plots were built using this package.

2.6 Seaborn[8]
"Seaborn" [8] is a Python data visualization library based on "Matplotlib" [7]. It provides a high-level interface for drawing attractive and informative statistical graphics. It was especially useful to plot correlation matrix, showing correlation between different features.

2.7 Pandas[9]
"Pandas" [9] is a fast, powerful and easy to use open source data analysis and manipulation tool. We used it to load the data[1] provided by NASA.

3 Data Description

Engine degradation simulation was carried out using C-MAPSS. Four different were sets simulated under different combinations of operational conditions and fault modes.
In this research only one of four provided data sets[1] was used, due to others being rather unsuitable for machine-learning purposes.

Records several sensor channels were taken to characterize fault evolution. The data set[1] was provided by the Prognostics CoE at NASA Ames.

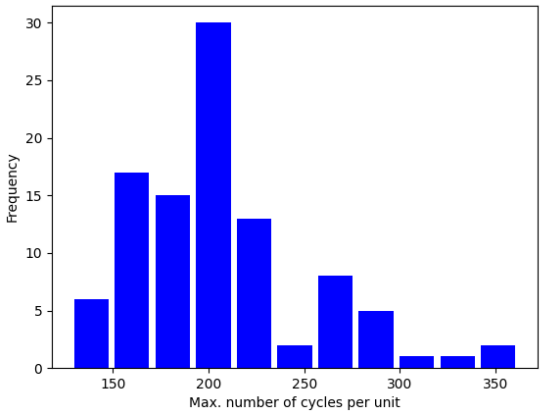

Figure 3.1 Units‘ lifetime frequency

Each unit in our data set had a certain number of cycles until it completely broke. Figure 3.1 shows the frequency of each units’ lifetime.

For each unit, 3 operational settings and 21 sensor measurements were given. These were taken after each finished cycle. Some sensor measurements had no variance in data and therefore a rather low predictive power (Figure 3.2). These sensor measurements were left out of the training set.

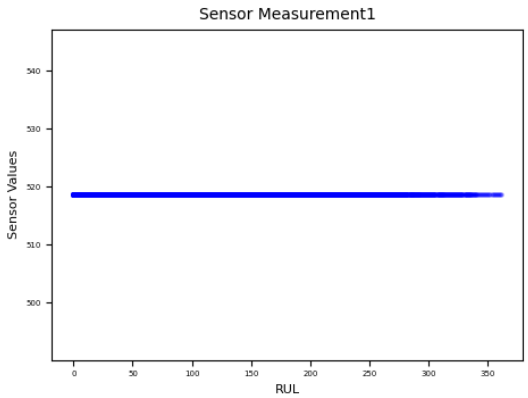

Figure 3.2 Sensor measurement with low predictive power

Remaining of sensor measurements (Figure 3.3) was forwarded to further processing i.e. feature extraction in time series, and data normalizing.

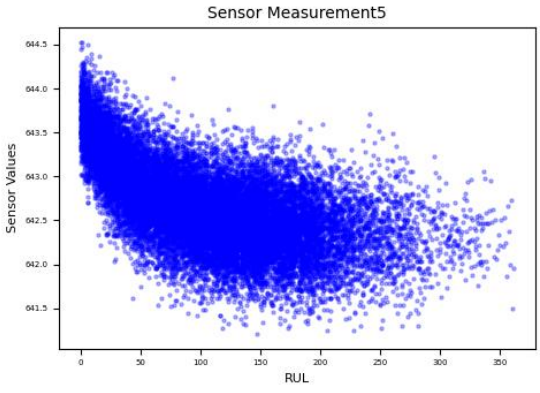

Figure 3.3 Sensor measurement data used for further processing

4 Building a model

One of the main challenges in building a model with this data set[1] was preprocessing of the data itself. This was due to a large noise in data set[1], and insignificant sensor measurements.

In the research conducted by "SDSC" [2], an ML model was built using data extracted via functions provided by "tsfresh"[3] package. This data was later processed with built-in functions and transformed using sigmoid transformation.

Similar approach was taken in our research except that for data transformation, we used "StandardScaler" provided by "scikit learn"[4] package. Also, in our research, sensor measurements with no predictive power (Figure 3.2) were not included in feature extraction process. Nonetheless, due to the size of original data set [1], feature extraction process took significant amount of time.

Since we were not provided with target data in the original data set [1], we had to calculate remaining useful lifetime based on the number of cycles each unit has. This means that the last performed cycle was the point of unit breakage, therefore making RUL = 0.

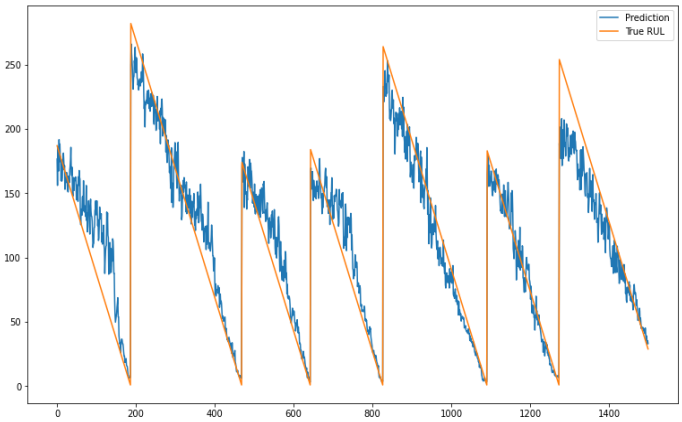

Figure 4.1 Plot of predicted RUL and true RUL on training data

After preprocessing data, we built a model using "LGBMRegressor" [5], and made predictions on training and test data. Figure 4.1 shows the plot of predicted and true RUL on training data.

Final step of our research included evaluation of model performance on training and test data.

5 Conclusion

In the research concluded, we could see that the performance of our model did not improve, nor worsen with using "StandardScaler" for normalization of the training data. The mean average error recorded on training set was 19.06 and R2-score 0.85. Root mean squared error yielded result of 25.77.

Our model performed as good as the model introduced by "SDSC"[2] – on training set, but had a rather poor performance on test set. Root mean squared error of training set was 40.41, mean average error yielded 30.04. Figure 5.1 depicts development of mean average error after each prediction.

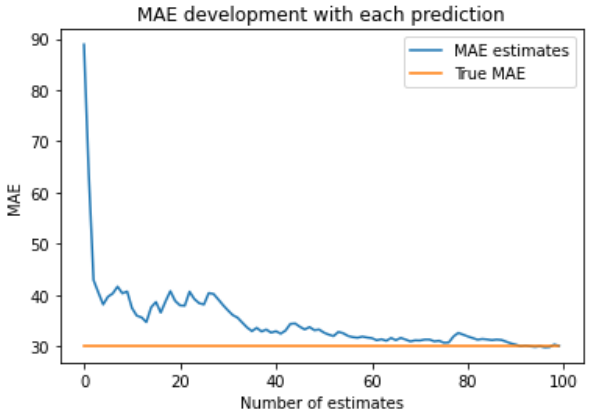

Figure 5.1 MAE after each prediction

References

[1] **A. Saxena and K. Goebel (2008). "Turbofan Engine Degradation Simulation Data Set", NASA AMES Prognostics Data Repository (https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/), NASA AMES Research Center, Moffett Field CA**

[2] **Dr. Andreas Wierse (2019). "Turbofan remaining useful life Prediction", (https://www.sdsc-bw.de/turbofan-remaining-useful-life-prediction/), SICOS BW GmbH**

[3] **Maximilian Christ, Nils Braun, Julius Neuffer, others. (2016-2021),(https://tsfresh.readthedocs.io/en/latest/index.html), Blue Yonder GmbH**

[4] **Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011. Scikit-learn: Machine Learning in Python.**

[5] **Microsoft Corporation (2021). https://lightgbm.readthedocs.io/en/latest/index.html**

[6] **Casper da Costa Luis (2015-2021). https://tqdm.github.io/**

[7] **J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, (2007). https://ieeexplore.ieee.org/document/4160265**

[8] **Waskom, M. L.,seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021 (2021). https://doi.org/10.21105/joss.03021**

[9] **McKinney W, others. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. p. 51–6. (2010). https://pandas.pydata.org/docs/**