A Data-driven Approach for RUL Prediction of an Experimental Filtration System

H. Beirami[1], D. Calzà[1], A. Cimatti[1], M. Islam[1], M. Roveri[2], P. Svaizer[1]
[1]*Fondazione Bruno Kessler, Trento, Italy*
{*hbeirami,calza,cimatti,mislam,svaizer*}*@fbk.eu*
[2]*University of Trento, Trento, Italy*
marco.roveri@unitn.it

ABSTRACT

This paper illustrates a data-driven approach adopted to address the PHME2020 Data Challenge competition. The aim of the challenge was to estimate the Remaining Useful Lifetime (RUL) of an experimental filtration device analyzing its clogging status by means of static (e.g. data sheets, fluid type, sensors) and dynamic (e.g. sensing data) information. We address the problem employing different state-of-the-art feature extraction, feature selection, and machine learning techniques. In this paper, we describe the approach followed to assess the problem and to generate robust and adaptable prediction models together with a corresponding performance assessment and robustness evaluation. The performance of the proposed solution is calculated in term of penalty score. The final penalty score 57.24 ranked 2nd in the above-mentioned data challenge competition.

1. INTRODUCTION

In the last decades, there has been an increased interest in Prognostics and system Health Management (PHM) due to its capability of assuring productivity, performance, and reliability of a system while moderating the maintenance expenses. Predicting the Remaining Useful Lifetime (RUL) is the most challenging part of PHM procedure that makes creating reliable and robust models to predict the RUL of tremendous importance (Gouriveau, Medjaher, & Zerhouni, 2016).

In this paper we describe our data-driven approach to address the problem of creating a model to predict the RUL of a critical filter (subject to clogging) equipping a filtration system of an experimental rig built with the aim of simulating a filter clogging failure in a realistic plant. This scenario is of high interest in many engineering applications involving purification of liquids or gas, e.g. for intercepting contaminants in fuel for engines (Eker, Camci, & Jennions, 2015). The dataset for training and evaluation consists of several run-to-failure observations of the rig under different conditions, characterized by combinations of particle size and solid ratio. The final model shall be able to maintain its performance whenever a smaller portion of the dataset is used as training set, and if unforeseen data is given in input to the model.

In our approach, we did not try to model the degradation curves directly, as done for example with a parameterized model in (Sreenuch, Khan, & Li, 2015). We adopted a mainly data-driven approach, where the curves are learned from the experimental data during the training phase, but we also exploited to some extent a model-based component consisting in the computation of the amount of clogging particles progressively intercepted by the filter.

This paper is structured as follows. In Sect. 2 our understanding of the problem is provided. We then outline the approach, and provide details of the feature selection and model setup in Sect. 3. Moreover, in this section we also discuss our methodology for model validation and evaluation. Finally, we provide the analysis of the results and the comparison of proposed models in Sect. 4.

2. UNDERSTANDING OF THE CHALLENGE

The filtration system on which the datasets provided by the PHM Society were based, consists in an experimental rig built with the aim of simulating a filter clogging failure. The main components of this experimental system are:[1]

- A suspension composed by water and Polyetheretherketone (PEEK) particles as injected fluid. The particles are mixed with water in different concentrations in order to test the adaptability of the final models.
- A filter: the critical component subject to clogging.
- A peristaltic pump, which maintains stable the flow of the prepared fluid. It has been chosen in order to avoid issues related to particles in the fluid.

[1]We refer the reader to the challenge web page for additional details about the characteristics of the different components.
`http://phmeurope.org/2020/data-challenge-2020`.

- Tubing, the majority of which is made of rigid polypropylene to avoid expansions due to eventual pressure build up, which might interfer with the observed pressure increase as a result of the filter clogging.
- A dampener placed downstream with respect to the pump, which drops any pulsation in the fluid flow.
- A series of flow meter to record the flow rate.
- Two pressure transducers placed respectively upstream and downstream with respect to the filter, in order to record the pressure drop due to the filter clogging.

The challenge provides the participants datasets corresponding to different experiments where four different profiles, given by the solid ratio of the fluid (0.4, 0.425, 0.45, 0.475), and different particle sizes (small, medium, large) have been considered. The dataset is composed of several sensorial data such as the *flow rate* (`flow_rate`), together with *upstream* (`pup`) and *downstream pressure* (`pdown`). The *particle size* (`psize`) and the *solid ratio* (`sratio`) are also given. All the sensorial data has been sampled at 10Hz. For each particle size and solid ratio combination, the respective dataset contains recording of four sensorial sample streams. The available dataset does not include streams for the medium particle size case, while the 0.475 solid ratio is only provided in the validation set as depicted in Fig. 1. More in details, the dataset consists of two distinct parts:

Sample	Particle Size (micron)	Solid Ratio (%)	Condition	
1	45-53	0.4	1	Training
2	45-53	0.4		
3	45-53	0.4		
4	45-53	0.4		
5	45-53	0.425	2	
6	45-53	0.425		
7	45-53	0.425		
8	45-53	0.425		
9	45-53	0.45	3	
10	45-53	0.45		
11	45-53	0.45		
12	45-53	0.45		
33	63-75	0.4	4	
34	63-75	0.4		
35	63-75	0.4		
36	63-75	0.4		
37	63-75	0.425	5	
38	63-75	0.425		
39	63-75	0.425		
40	63-75	0.425		
41	63-75	0.45	6	
42	63-75	0.45		
43	63-75	0.45		
44	63-75	0.45		
13	45-53	0.475	7	Validation
14	45-53	0.475		
15	45-53	0.475		
16	45-53	0.475		
45	63-75	0.475	8	
46	63-75	0.475		
47	63-75	0.475		
48	63-75	0.475		

Figure 1. The structure of the PHME 2020 Challenge dataset.

- A training set containing 24 run-to-failure experiments. It includes data acquired using *small* and *large* particle sizes, and different solid ratio related to the profile numbers *1, 2* and *3*. For each combination of particle size and concentration, 4 different experiments were provided.
- A validation set containing 8 run-to-failure experiments. It includes data corresponding to experiments using *small* and *large* particle sizes, and the solid ratios that are not present in the training set (i.e. profile number *4*). For each combination of particle size and concentration, 4 different experiments were provided.

The sensed data allow to extract the pressure drop (`drop`), which consists in the difference between upstream and downstream pressures and gives an indication of the clogging status of the filter. According to the challenge, the filter is considered clogged as soon as the pressure drop reaches 20psi.

The main objectives consist in predicting the RUL of the filter given by its clogging status, and providing a model being able to maintain its performance whenever a smaller section of the dataset is used as training set.

In order to satisfy the objectives of the challenge, four different models were requested: one model trained using all the provided experiments, one by using only the 75% of them, one by using the 50% and one using the 25%.

Model performances will be evaluated based on the following penalty score:

$$\sum_{i \in \{25,50,75,100\}} (1.5 * MAE(M_i(Te)) + MAE(M_i(Tv))) \quad (1)$$

where *MAE* is the Mean Absolute Error function selected for the final evaluation, M_i is the model generated with *i%* experiments, *Tv* is the aggregation of the training+validation datasets, and *Te* is the test dataset.

3. APPROACH

The followed approach consists in using as first analysis the given validation set as test set, in order to test the adaptability of our model to unforeseen data, and further split the training set into training and validation. However, we used the full provided training+validation set to train the model for the final delivery. The main workflow consists of i) features extraction using rolling window techniques, ii) features importance and correlation extraction leveraging on the correlation matrix, Support Vector Machine (SVM) coefficients of a linear kernel, Recursive Feature Elimination (RFE) and monotonicity test with and without smoothing, iii) model setup and validation using a repeated cross validation technique, and iv) a repeated approach for testing the model on the training and validation sets in order to extract generalized final scores by averaging the errors of the repetitions. The entire approach has been implemented on top of the `scikit-learn` (Pedregosa et al., 2011) machine learning infrastructure. In the following, we provide the most relevant details of each of above steps.

3.1. Features selection

Since the definition of RUL of the filter is based on the time instant at which the pressure drop reaches the threshold of 20psi, our approach is based on the analysis of the time evolution of the curves representing that drop (Sreenuch et al., 2015). We characterized the drop evolution by means of several statistical features such as *kurtosis* (`kurt`), *mean-to-peak ratio* (`mpr`), *root mean square* (`rms`), *root mean square in frequency-domain* (`rmsf`), *wavelet spectral energy* (`wse`), *skewness* (`skew`), *variance* (`var`), *standard deviation* (`std`), *covariance* (`cov`) and the *slope of the lin-*

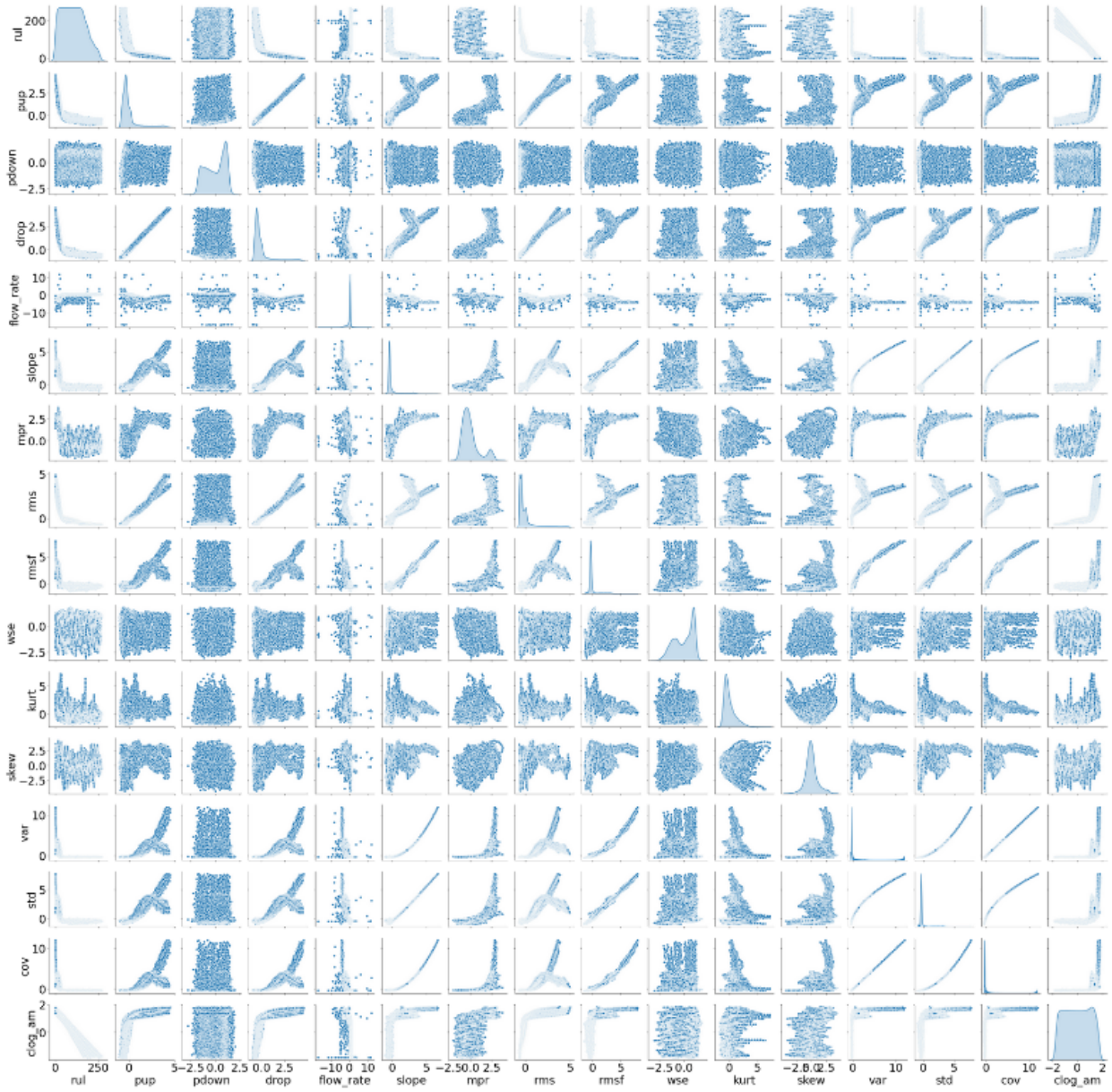


Figure 2. Pairwise bivariate distributions of the considered features.

ear regression (`slope`) (Hamadache, Jung, Park, & Youn, 2019). The computation of these features requires a proper length of rolling window. For such analysis, we applied a moving window with a length of 10 seconds which turns out to be a reasonable and practical window length given the dataset. This window corresponds to 100 sample points.

In addition to statistical features, we further extract a feature that aims to indicate the amount of cumulative dirt intercepted by the filter when the particles are suspended in the water flowing through it. To this extent, we introduced the new feature, `clog_am`, computed as the time integral of the product between flow rate, particle size and solid ratio.

To identify the most informative features, we applied several state-of-the-art feature selection techniques such as pairwise bivariate distribution analysis of the considered features (Fig. 2), correlation matrix with absolute values (Fig. 3), RFE and analysis of the coefficients of the linear kernel of a SVM [2] (Fig. 4), and monotonicity analysis (Fig. 5) with and without smoothing. All the mentioned techniques were applied both before and after a standardization of the features, although all the included figures refer to the standardized ones.

This analysis highlighted that the most important features ap-

[2]Information about RFE and analysis of the coefficients of the linear kernel of a SVM can be found in (Guyon, Weston, Barnhill, & Vapnik, 2002).

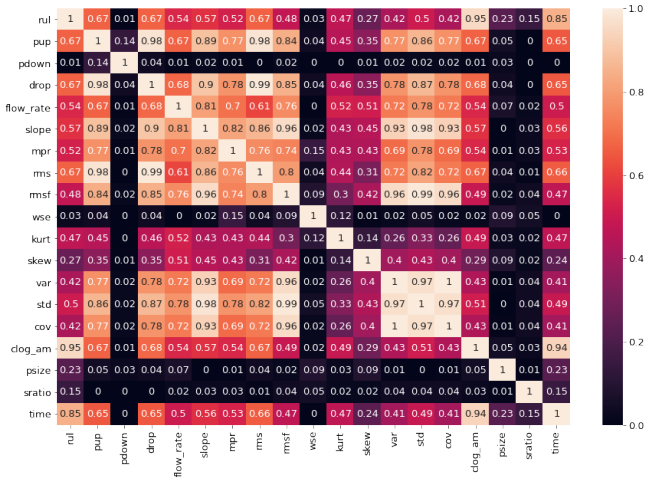


Figure 3. Correlation of the features: it represents the absolute value of the correlation.

pear to be `flow_rate`, `pup`, `drop`, `slope`, `rms`, `rmsf`, `var`, `std`, `cov`, and `clog_am`, together with `psize` and `sratio`, which define important information of the observed experiment. Thus, these are the features we used for the remaining phases of our approach.

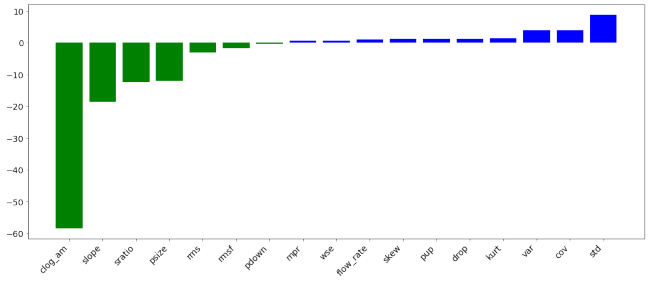


Figure 4. Features coefficients of a SVM linear kernel

We observerd, as it is expected, that the filter status (measured as *pressure drop*) degrades monotonically until the full clogging (when it reaches the 20psi as indicated in the challenge). Therefore, we discarded all the data after the first sample that results in passing this threshold.

3.2. Neural Network Structure Setup

To address the challenge, we considered several sequential machine learning models (e.g. support vector regressors) and neural networks (e.g. long short-term memory) with different structures, combining different preprocessing techniques. In the first model (named `A` hereafter), we adopted a preprocessing pipeline composed of a standard scaler, and a four-layers sequential neural network (see Fig. 6) implemented on top of the Keras (Chollet et al., 2015) infrastructure using the Tensorflow (Abadi et al., 2015) backend. The first layer (`dense_input`) takes the 12 standardized selected features,

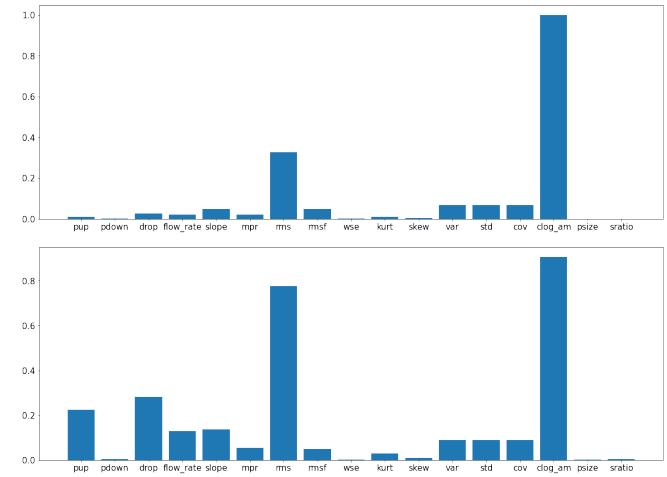


Figure 5. Features monotonicity coefficients without (picture above) and with (picture below) smoothing.

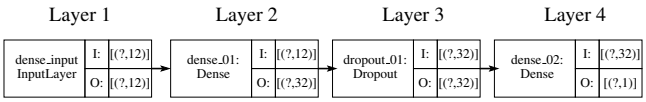


Figure 6. Neural network structure adopted for the challenge.

and outputs them as input of the following layer. The second layer (`dense_01`) takes the 12 outputs from the previous layer and generates 32 outputs by means of the `sigmoid` activation function. The third layer (`dropout_01`) takes the 32 outputs from the previous layer and drops the units with a rate of `0.06` in order to prevent an overfitting model. The final layer (`dense_02`) takes the 32 outputs from the previous layer and generates a single output corresponding to the estimated RUL by means of the `linear` activation function. The loss error function used is the mean-squared-error, while the adopted optimizer is `adam`.

The first additional model (named `B`) consists in the same structure as the above one, but adopting the `relu` activation instead of the `sigmoid` in the `dense_01` layer, and adjusting the level of the `dropout_01` layer. The second additional model (named `C`) has 16 nodes in the `dense_01` layer instead of 32, again with an adjusted level of the `dropout_01` layer.

We then extracted the MAE as metrics and set the number of epochs to 100. This last value in particular lets the training and validation losses converge without presenting signs of over- or underfitting. The complete structure of the final model has been selected according to the evaluation techniques described in section 3.3.

3.3. Model Validation and Evaluation

We applied repeated K-fold cross validation (Kohavi, 1995) in order to assess the generality of the considered different

machine learning models and neural networks for this data challenge. For simplicity, as regards this validation phase, we refer to the aggregation between training+validation sets as training set, while the mentioned validation sets are represented by the different K folds. Before proceeding with the repeated cross validation, a first training was done in order to have a first insight of the convergence of the losses (see Fig. 7 for the results of the network depicted in Fig. 6) and to be able to adjust the network structure in case of critical losses convergence issues, together with the number of training epochs. Typical K-fold cross validation (Stone, 1974) divides a dataset into K separate subsets. These subsets are mutually exclusive and approximately equal size. In K iterations, one subset is chosen and the training procedure is performed in the other $K-1$ subsets. Then, yielded model is tested on the chosen fold. In repeated K-fold cross validation, however, the whole aforementioned procedure is performed N times in a way that the indexes are shuffled to select K folds containing different portions of data among different repetitions (Vanwinckelen & Blockeel, 2014). Furthermore, in a repetition, each selected fold is used for validation; after the K iterations using each fold as validation set, the predicted RUL value is reassembled and the errors are extracted. For each repetition, a mean of the K losses is also extracted to assess convergence of the losses among the repetitions. The candidate model is selected only if training and validation losses are convergent and the errors are reasonably low. This approach contributes positively to stress the robustness and generality of the model prior to its evaluating routine. The choice of K plays an important role. Indeed, a poorly chosen value for K may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model). We addressed the challenge using $N = 6$ and $K = 5$. In particular, $K = 5$ was chosen since it guarantees that each train/test group of data samples is large enough to be statistically representative of the considered dataset.

We applied the above approach to evaluate the performance of each considered network.

The results of the network validation confirmed that the network structure depicted in Fig. 6 is worth to continue with the real training and evaluation (for this network the results are depicted in Fig. 7). Thus, we proceeded by training the network with the aggregation of different training+validation sets of the challenge to create corresponding models. In particular, we created four models each corresponding to the split of the training+validation data set to consider 25%, 50%, 75% and 100% of the dataset. To select the 25-75% subsets, we chose randomly 1 out of 4 for the 25%, 2 out of 4 for the 50%, 3 out of 4 for the 75% case from 4 different experiments for each combination of particle size and concentration.

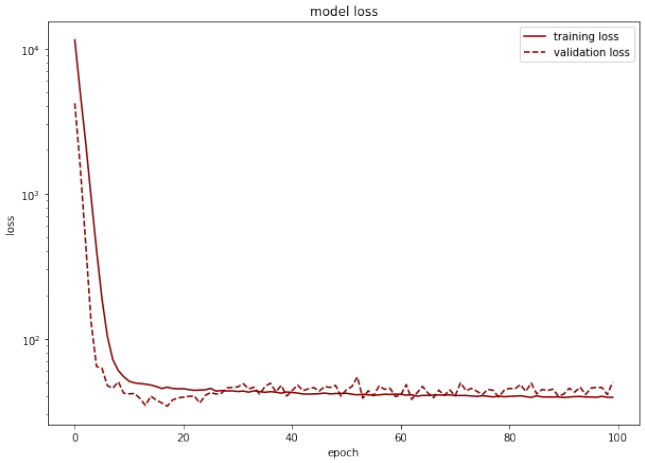
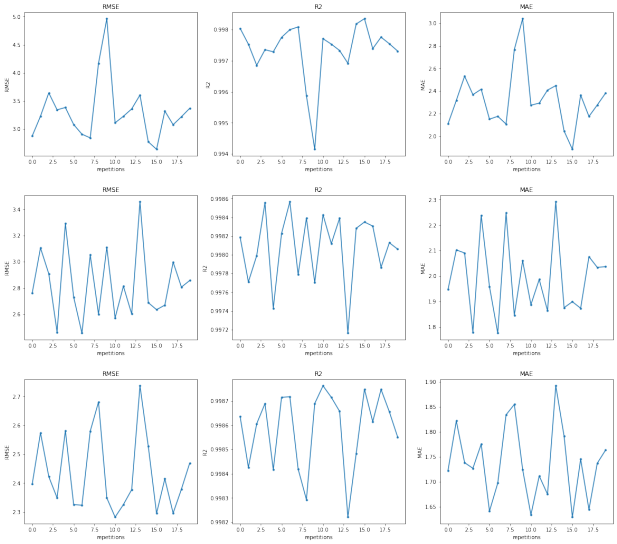


Figure 7. Losses over the epochs (in log scale).

The model based on 100% of the data is trained and further evaluated on the full aggregation of training+validation datasets, computing the corresponding MAE. The evaluation of the 25%, 50% and 75% models has been performed with the aim of extracting a robust and generalized error function.

Let s be the number of samples for each combination of *particle size* and *solid ratio*, p be the used percentage of training+validation data used to train the model (i.e. 75%, 50% and 25%), and k be the number of distinct combinations of *particle size* and *solid ratio*, then we derive from the training+validation dataset $\binom{s}{p*s}^k$ different subsets.



(RMSE) (R2) (MAE)

Figure 8. Evaluation results over the repetitions for the 25%, 50%, 75% cases for $N = 20$ (top 25%, mid 50%, bottom row 75% resp.).

Following repetitive steps have been performed in order to compute the models and to test the respective robustness for

each of the 25%, 50% and 75% cases. First, N folds were taken from the previously computed subsets of the training+validation datasets. Then, iteration over this set of N folds were performed, and after training the model on this training+validation dataset, we evaluated the network on the entire training+evaluation dataset to extract the respective MAE errors. We considered $N = 20$ for the experiments since it guarantees to have a statistically representative analysis of the considered datasets. The results for the network depicted in Fig. 6 on the three models are reported in Fig. 8. After verifying that the models are substantially equivalent, the model was retained for the considered case is the last model fitted.

The list of size N of the computed MAE errors values for each of the three model allows to verify the robustness of the proposed models. However, only the MAEs of the selected models (i.e. the last model fitted for each percentage among its repetitions) represent the data used to calculate the final score for coherence.

4. RESULTS

The penalty score function to be used for the challenge is the one of Equation 1. The approach discussed in Sec. 3.3, enabled us to calculate the portion of the penalty score of the training+validation datasets for the three considered models. The results of these models are reported in Table 1.

	MAE(M_i(Tv))		
M_i	A	B	C
100%	1.646	1.285	2.125
75%	1.763	1.358	1.814
50%	2.037	1.752	2.059
25%	2.381	1.977	2.826
Penalty Score	7.827	6.373	8.824

Table 1. Comparison between the considered models.

We can observe that the choice of a smaller amount of training data results in an increasing error value. Judging by MAE values, the results show that model B exhibits the best performance among the proposed models. However, validation and evaluation analysis showed that the models based on the sigmoid activation function (model A) are more robust to unforeseen data with respect to the relu function (model B). Therefore, we opted to use model A as the final candidate to run the challenge.

Equation 2 represents the final penalty score reported by the tests applied by the PHM Society on the provided models. The final score for the model A (which at the end ranked 2nd in the competition) is:

$$\sum_{i\in\{25,50,75,100\}} (1.5 * \mathit{MAE}(M_i(\mathit{Te})) + \mathit{MAE}(M_i(\mathit{Tv}))) = 57.24. \quad (2)$$

The notebook and all the material to reproduce the results reported in this paper can be downloaded from https://bit.ly/fbkphme20challenge.

CONCLUSION

The approach we adopted in this challenge was mainly inspired by a generic data-driven learning framework, in which the training phase plays the predominant role in mapping observations into the desired RUL estimate. We decided also to inject some model-based knowledge with the calculation of a specific feature that expresses the amount of dirt progressively intercepted by the filter, given the size and concentration of the suspended particles. We are aware that the model-based aspect could be improved by considering a more precise physics-based clogging progression model (Eker et al., 2015) and a more accurate fitting of the parameters characterizing the pressure-drop curves, e.g. with particle filtering, as done in (Sreenuch et al., 2015). Such improvements would be relevant especially when particle size and concentration cannot be known exactly and must be inferred from the observed data.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/ (Software available from tensorflow.org)

Chollet, F., et al. (2015). *Keras.* https://keras.io.

Eker, Ö. F., Camci, F., & Jennions, I. K. (2015, 12). Physics-based prognostic modelling of filter clogging phenomena. *Mechanical Systems and Signal Processing*, *75*, 395-412. doi: 10.1016/j.ymssp.2015.12.011

Gouriveau, R., Medjaher, K., & Zerhouni, N. (2016). PHM and Predictive Maintenance. In *From prognostics and health systems management to predictive maintenance 1* (p. 1-13). John Wiley & Sons, Ltd.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1-3), 389–422.

Hamadache, M., Jung, J. H., Park, J., & Youn, B. D. (2019, Jun 01). A comprehensive review of artificial intelligence-based approaches for rolling element bearing phm: shallow and deep learning. *JMST Advances*, *1*(1), 125-151.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In

Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes (pp. 1137–1145). Morgan Kaufmann.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Sreenuch, T., Khan, F., & Li, J. (2015). Particle filter with operational-scalable Takagi-Sugeno fuzzy degradation model for filter-clogging prognosis. *J. Aerosp. Inf. Syst.*, *12*(5), 398–412.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(2), 111–133.

Vanwinckelen, G., & Blockeel, H. (2014). Look before you leap: Some insights into learner evaluation with cross-validation. In *Proceedings of the 1st Workshop on Statistically Sound Data Mining, SSDM 2014, held at ECML/PKDD 2014, Nancy, France, September 15, 2014* (Vol. 47, pp. 3–20). JMLR.org.

Hani Beirami received his masters degree from the University of Tehran, Iran. He is currently enrolled as a Ph.D. student in the Information Engineering and Computer Science Department of the University of Trento funded by a grant by Fondazione Bruno Kessler, Italy. His research interests include formal methods and machine learning. The focus of his Ph.D. at the University of Trento is on improving the functionality of predictive maintenance.

Davide Calzà is Bachelor's student at the Information Engineering and Computer Science Department of the University of Trento, Italy. He received his High School diploma from the I.T.T. G.Marconi of Rovereto, Italy. He is part of the Embedded System Unit of the Fondazione Bruno Kessler in Trento, Italy, as junior research assistant since 2019. His interests focus on predictive maintenance in the field of rotating machinery analysis, diagnosis and prognosis.

Alessandro Cimatti is a senior researcher at Fondazione Bruno Kessler, Trento, Italy, where he coordinates the activities on Digital Industry and leads the research unit in Embedded Systems. His research interests concern formal verification of industrial critical systems, methodologies for design and verification of hardware/software systems, decision procedures and their application, safety analysis, diagnosis and diagnosability. He has published more than two hundred conference and journal papers in the fields of Formal Methods and Artificial Intelligence, and has co-chaired the FMCAD, SAT, DX and SEFM conferences.

Manjurul Islam received a Ph.D. degree in Computer Engineering from the University of Ulsan, South Korea in 2019. He is now working as a postdoctoral researcher in the Embedded Systems Unit of Fondazione Bruno Kessler in Trento, Italy. His research interests include multimedia signal processing, statistical modeling, and predictive maintenance in the field of industrial machine analysis, diagnosis and prognosis.

Marco Roveri received a Ph.D. degree in Computer Science from the University of Milano, Italy in 2002. He is an Associate Professor in the Information Engineering and Computer Science Department of the University of Trento, Italy. He was Senior Researcher in the Embedded Systems Unit of Fondazione Bruno Kessler in Trento, and before a researcher in the Automated Reasoning Division of the Istituto Trentino di Cultura, Trento, Italy, in 1998, first as a consultant, and in 2002 as a full time researcher. In 2008 he was appointed as a senior researcher in the Embedded Systems Unit of the Fondazione Bruno Kessler, Trento, Italy, and in 2020 he was appointed as Associate Professor in the University of Trento. His research interests include automated formal verification of hardware and software systems, formal requirements validation of embedded systems, model based predictive maintenance, and automated model based planning, and application of such techniques in industrial settings.

Piergiorgio Svaizer received his degree in Electronic Engineering from the University of Padua, Italy, in 1989. From 1989 to 1990, he was a consultant at the National Council for Research, Padua, where he worked on multi-band image processing and pattern recognition. In 1991 he joined the Interactive Sensory System Division at ITC-IRST, Trento. He is a now a senior researcher at the Center for Information and Communication Technology, Fondazione Bruno Kessler, Trento, Italy. Since 2001 he collaborates in teaching activities for digital signal processing with the University of Trento. His main research interests include digital signal processing, statistical modeling and time-frequency analysis of multichannel signals, microphone arrays for acoustic scene analysis and spatial selectivity, speech analysis, enhancement and separation, DSP for condition monitoring of industrial equipment.