Predicting Remaining Useful Life with Machine Learning Algorithms for Predictive Maintenance

Gonçalo António Gonçalves Gago Felício
goncalo.antonio@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

December 2021

Abstract

Asset failures are hard to foresee and incur significant losses in the manufacturing industry. The present work explores the application of ML techniques to predict the RUL of assets from monitoring data, in the context of PdM. A successful PdM strategy can not only reduce breakdowns and downtime of assets, but also maximize production, improve product quality, and safety of workers. We propose three benchmark models using the RF, XGBoost, and LSTM, and a BiLSTM model, with sliding time window, to further optimize through parameter and features selection. The proposed models are applied to the N-CMAPSS dataset provided by NASA and the results obtained show the effectiveness of the models. Both the parameters selection and the features selection increased the performance of the BiLSTM model. Limitations were identified with the chosen dataset and proposed framework, however the results obtained show promise for further research and in providing critical information to support decision making for predictive maintenance strategies.

Keywords: Remaining useful life prediction, Predictive maintenance, Machine Learning, Bi-directional long short-term memory , N-CMAPSS

1. Introduction

In the manufacturing industry, an asset failure can be described as the asset performing its intended function defectively, for example, a product is manufactured by the asset with quality standards below those predefined, or the asset completely halting its operation, known as breakdown. The latter of the situations has more severe consequences, and both require maintenance actions to be taken. The consequences of a failure can be low, however, in safety-critical systems, such as the aircraft industry, failures can have extremely high economic impacts and even loss of life. The only way to avoid failures is by properly designing, installing, operating, and maintaining an asset [17].

Maintenance is the management of assets and control of costs, however asset monitoring and management are arguably complex tasks. Maintenance costs represent a significant share of all the costs associated with manufacturing and production. Depending on the industry maintenance costs can account for 15% up to 70% of the total production costs[26]. Additionally, maintenance, often times is an ineffective process, with up to one third of all investment being wasted in improper or unnecessary maintenance actions. The main cause for this is the lack of knowledge regarding the actual health condition of assets and the need for maintenance actions[17]. Relying on statistical trend data from manufacturers to predict failures will lead to ineffective maintenance and waiting for an asset to breakdown is also not an option in safety-critical situations[19].

PdM is a maintenance method that, through the monitoring of the actual operating condition of an asset and other important indicators is able to predict an impending failure and allow enough time for maintenance actions to be taken, preventing the failure. The main advantages of this maintenance method is the increased reliability and availability of assets, improved environmental and worker safety and reduced costs in parts inventory and maintenance labor. Successful implementations of PdM strategies have shown the effects of these advantages, namely[19]: reduction in maintenance costs of 25% to 30%; elimination of breakdowns of 15% to 70%; and reduction of downtime of 35% to 45%, . The technological advancements from I4.0 and IoT, specifically sensor technology, as well as developments in AI, in specific, ML, with great success in prediction algorithms, has converged in new predictive maintenance techniques through the use of ML algorithms [4, 30, 15, 29, 25, 12, 28, 10, 20, 3].

2. Predictive Maintenance

Historically, in the period before World War II, maintenance was seen as an added cost to production, without the corresponding value to the company. The most common form of maintenance was to repair assets after they broke down, considering this the cheapest solution. This maintenance strategy is known as RTF and, today, is considered to be the most costly type of maintenance. Currently, high-level management understand the benefits of a proper maintenance strategy, which were overlooked in the past. Maintenance can, not only prevent asset failures but also[17]:

- maximize production;
- optimize useful life of equipment;
- reduce breakdowns and downtime;
- improve product quality and inventory control.

These advantages are crucial to succeed in the highly competitive environment that manufacturing companies face these days, as well as meet the increasing costumer demands in areas such as environmental and public safety, product quality and product reliability.

2.1. Predictive Maintenance

The focus of this thesis is RUL estimation for Predictive Maintenance. There are many definitions of PdM in the literature [7, 19] with small differences between them, therefore, for this work, it was considered that PdM is a type of planned maintenance where signs of impending failure are monitored and detected in order to predict when the failure of assets will occur, and carry out the appropriate maintenance work to prevent the failure altogether.

PdM uses the most effective technique to determine the current condition of assets and, based on this, schedules maintenance actions when they are needed. Successfully implementing a PdM maintenance strategy optimizes the availability of assets and greatly reduces costs of maintenance since asset failures are prevented altogether and the time between repairs is maximized. Product quality, reliability, productivity, and profitability are all increased as a direct consequence [17].

PdM is also specially helpful when reliability is most important, situations where breakdowns have severe consequences, such as nuclear power plants, emergency systems, and aircraft industry. Another reason for choosing PdM is the current situation of Industry 4.0. Developments on technologies of Internet of Things, sensor technology, Cloud Services, and ML have created space for new and promising solutions in PdM [4, 24]. It is the combination of all these technologies that has provided the data and tools required to achieve better results than traditional PdM techniques [34].

2.2. Machine Learning for predictive maintenance

Traditional PdM techniques have been used in PdM management strategies extensively with varying degrees of success [17]. One thing that is common in all traditional techniques is the requirement of expert knowledge of maintenance and machine dynamics in order to operate and interpret the results. However ML techniques do not require the dedicated expert knowledge of the asset being studied. This means ML algorithms are not restrained to a single domain and can be adapted to suit many different situations.The success of ML algorithms in developing models for forecasting has been applied in a wide variety of industries and situations, including image processing, robotics and speech recognition. Therefore, the rise of ML techniques in the PdM management circles can be attributed to these factors, additionally the results and success of such implementations has cemented their position.

2.2.1 Random Forest

Random Forest is an ensemble of Decision Trees, a ML algorithm with great versatility capable of both classification and regression tasks. The output of the RF model is calculated collectively by the individual Decision Trees. The premise is that an aggregated answer from many DT is better than a single answer from a single DT. Despite the simplicity of the RF algorithm it is one of the most powerful ML algorithms available [6].

Mathew, V. et al. [16] train and compare several ML algorithms to predict RUL of turbofan engine, using the famous CMAPSS dataset from the Prognostics Data Repository of NASA. In this study the authors conclude that RF shows the best results among 10 other ML algorithms, including, LR, DT, SVM, KNN, K-Means and GBoost. The performance index used is once again RMSE. The authors conclude that RF outperforms the other techniques because it captures the variance of the input variables at the same time and enables a high number of observations to take part in the final prediction, a consequence of its structure.

2.2.2 Extreme Gradient Boosted Trees

XGBoost is a tree boosting system that is similar to the RF algorithm but uses a majority voting technique to define the final class and the sequential tree boosting technique. XGBoost has appeared in the prognostics field often alongside RF and LR models and has shown similar results.

Binding, A. et al. [9] use these three ML techniques in forecasting machine downtime of printing

machines based on real-time predictions of future failures. The authors use not only machine-related data but also unstructured data in the form of operator notes, which improved the performance of prediction models. The metrics used to evaluate the models include the AUC, ROC, PRC and number of TP, FP, TN, FN at different decision thresholds. The results show that RF and XGBoost show similar results and outperform LR.

2.2.3 Long-Short-Term-Memory

LSTM algorithm is a neural network model composed of LSTM cells. The LSTM cell was proposed in 1997 by Hochreiter, S. and Schmidhuber, J. [8]. The greatest advantage of the LSTM cell over other ML techniques is the ability to identify long-term dependencies in the data which makes this ML technique specifically suitable to study time-series data, long text or audio recordings. The architecture of a LSTM cell is shown in Figure 1.

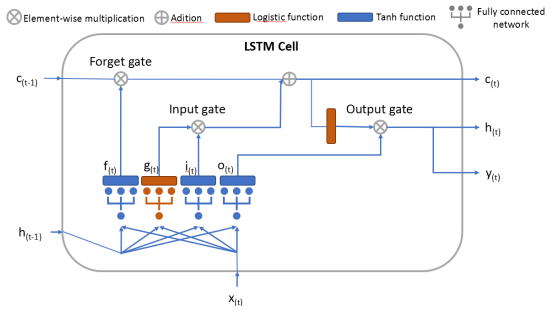


Figure 1: Architecture of a basic LSTM cell based on [6]

The ability to recognize long-term patterns comes from the network being able to save in the long-term state information that is relevant, and discard information that is not useful. The long-term state, identified in Figure 1 by $c_{(t-1)}$ goes first through a *forget gate*, where the useless information is discarded, then new information is added with the addition operation. The new information to be added is selected in the *input gate*. The result of the operation is the variable $c_{(t)}$ which is sent out without further operations to become the long-term state of the next step. The long-term state is also copied at this point and goes through the *output gate*, which filters the long-term state to obtain the short-term state $h_{(t)}$ and the output of this time step t, $y_{(t)}$.

The input vector $x_{(t)}$ and the short-term state of the previous step $h_{(t-1)}$ relate to the process previously described through 4 different fully connected layers and are the source of new information. These layers control the mentioned gates, *forget, input* and *output gates* through activation functions. The activation function used to control the gates is the sigmoid function that outputs values in the range [0, 1] and the tanh function that outputs values in the range [-1, 1]. With the sigmoid activation function, the gates are closed if the output value is 0, and open if the output value is 1.

The network analyses the input vector and the previous short-term state in the main layer using the tanh activation function, with output vector $g_{(t)}$. $f_{(t)}$ controls the *forget gate*, selecting which part of the long-term state is discarded. $i_{(t)}$ controls the input gate, selecting which part of $g_{(t)}$ is added to the long-term state of the previous step, generating the long-term state of the current step. Lastly, $o_{(t)}$ controls which part of $c_{(t)}$ is read and becomes the output of the current step $y_{(t)}$.

With this architecture the LSTM cell has the ability to learn which information or memory to retain in the long-term state (with the input gate), how long should it hold on to this memory (with the forget gate) and what memory should be extracted when needed (with the output gate). This gives it the desired ability of capturing long-term patterns in complex time-series data to use for RUL Prediction [6].

The LSTM network computations are shown in Equations (1) to (6).

$$i_{(t)} = \sigma\left(W_{ix} \cdot x_{(t)} + W_{ih} \cdot h_{(t-1)} + b_i\right) \quad (1)$$

$$f_{(t)} = \sigma\left(W_{fx} \cdot x_{(t)} + W_{fh} \cdot h_{(t-1)} + b_f\right) \quad (2)$$

$$o_{(t)} = \sigma\left(W_{ox} \cdot x_{(t)} + W_{oh} \cdot h_{(t-1)} + b_o\right) \quad (3)$$

$$g_{(t)} = tanh\left(W_{gx} \cdot x_{(t)} + W_{gh} \cdot h_{(t-1)} + b_g\right) \quad (4)$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \quad (5)$$

$$h_{(t)} = o_{(t)} \otimes tanh\left(c_{(t)}\right) \quad (6)$$

Where W_{ix}, W_{fx}, W_{ox} and W_{gx} are the weights of each of the four layers connected to the respective gates with respect to the input vector $x_{(t)}$, W_{ih}, W_{fh}, W_{oh} and W_{gh} are the weights of the same four layer with respect to the short-term state $h_{(t-1)}$ and b_i, b_f, b_o and b_g are the bias of each layer. The $\otimes$ symbol represents element-wise multiplication. The output of the layer at step t is the short-term state $h_{(t)}$.

LSTM models have been used in RUL prediction given its advantages with time series data. de Oliveira da Costa, P.R. et al. [5] propose a LSTM network combined with global Attention mechanisms in order to learn the relations between RUL and sensor data, and applied the model to the CMAPSS dataset, obtaining competitive results with other state-of-the-art methods. The combination with attention mechanisms allowed the authors to visualize temporal relationships between the target RUL and sensor data, the performance index

metrics used are the RMSE and a scoring function *Score* proposed by Saxena, A. et al. [18].

Zhang, Y. et al. [31] developed a LSTM model for predicting the RUL of lithium-ion batteries. The proposed model uses the resilient mean squared back-propagation method for optimization and the dropout technique to avoid over fitting. The developed model uses experimental data including temperatures and current rates to train the model and is then compared with SVM and a simple RNN models, outperforming both.

In conclusion LSTM networks have been used both with experimental and simulated data successfully in prediction models due to its ability on analysing large quantities of data, as well as its ability in capturing temporal patterns over the long term. As challenges in using LSTM networks we can identify that a large amount of hyper-parameters must be tuned either through techniques like grid search or through manual testing for optimal performance.

2.2.4 Bidirectional LSTM

BiLSTM is a ML technique composed of two independent and equal LSTM neural networks that are propagated in two opposite directions. One of the LSTM networks has forward propagation while the other has backwards propagation. Both the backward and forward propagation are simultaneous and combine to form to the output unit which is represented in Equation (7).

$$y_{(t)} = W_{y\overrightarrow{h_{(t)}}} \cdot \overrightarrow{h_{(t)}} + W_{y\overleftarrow{h_{(t)}}} \overleftarrow{h_{(t)}} + b_y \qquad (7)$$

Where $\overrightarrow{h_{(t)}}$ and $\overleftarrow{h_{(t)}}$ represent the output of the forward and backward propagation LSTM network, at time step t, respectively. $W_{y\overrightarrow{h_{(t)}}}$ and $W_{y\overleftarrow{h_{(t)}}}$ represent the weights with respect to the forward LSTM layer and the backward LSTM layer, respectively. b_y represent the output layer bias. Finally $y_{(t)}$ is the output of the BiLSTM network at time step t.

The main difference from a basic LSTM neural networks is that the BiLSTM can detect dependencies in the data not only from past information, but also from future information. This is a desirable feature that has been applied in text classification to better capture the preceding and succeeding context that exists in sentences [14]. In the context of time series data, this capability is also desirable to improve the accuracy of RUL predictions.

Zhao, C. et al. [32] proposed a double-channel hybrid deep NN based on CNN and BiLSTM. The model extracts the relevant features using the CNN and and captures the temporal dependencies of the data with the BiLSTM. The model also uses a sliding time-window for data preprocessing. Testing the model on the CMAPSS dataset the results show a better performance than other state-of-the-art models.

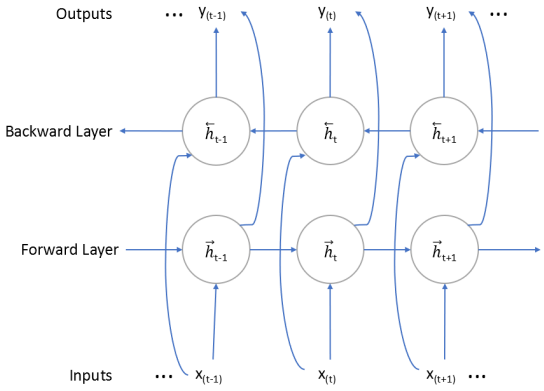


Figure 2: Architecture of a BiLSTM algorithm

The BiLSTM has been used by researchers to predict RUL successfully and is capable of all the advantages of a basic LSTM network while also capturing future information dependencies in the data, improving the accuracy of the models predictions.

3. Framework proposed

3.1. Proposed Machine Learning models

For this work, we propose a RF, a XGBoost, a LSTM, and a BiLSTM model. The RF, XGBoost and LSTM models will serve to obtain baseline results, from which we can then compare with the results obtained from the BiLSTM model. The RF and the XGBoost models were chosen due to the ease of use of these models as well as their good performance in both RUL prediction and computation time. Both models will be default models. The LSTM neural network is used as it often shows better performance than the RF and the XGBoost models. The LSTM model parameters are not optimized, but are chosen based on previous relevant literature [33]. The LSTM model proposed has one input layer and one hidden layer with 64 neurons each, the output layer is a regression layer with one neuron. The remaining parameters are a dropout of 0.5, a learning rate of 0.001, a batch size of 512, the optimizer functions RMSprop, an early stopping technique with a patience of 20 epochs, and a training/validation split of 90%/10%, respectively.

Finally, we propose a BiLSTM network, due to its ability to capture information from past and future information in the dataset. We expect results to be even better than the results of LSTM network after appropriate parameter selection. The base model parameters, similar to the case with the LSTM model, are chosen based on [22]. The layer topology is on input layer and three hidden layers with 64, 32, 16, and 8 neurons, respectively. The output layer and remaining model parameters are the same as with the LSTM model. As a limita-

tion to using the BiLSTM model we estimate that training and testing will take a considerably longer time. To counter this limitation the parameter optimization process will be a manual process, this is less time consuming than an automatic method, but is also less precise and does not guarantee that an optimal solution is found. Additionally, the use of cloud services is employed, through the use of the Kaggle platform[1] for the training and testing of all the models. This platform provides a graphics processing unit (GPU) for the intensive computations performed in training and testing complex models, such as the BiLSTM model.

One of the limitations of ML prediction algorithms is the generalization and transferability to new, unseen data. To measure this the BiLSTM model will be trained and tested with a subset of the full dataset available. Afterwards the remaining subsets of data will be evaluated with the final BiLSTM model proposed. Lastly, the full dataset is evaluated on the BiLSTM model proposed, and the results compared with previous ones.

3.2. Evaluation metrics

The evaluation metrics used are the ones suggested in [18]. This is the RMSE and NASA's scoring function, denominated *Score*. This metrics have been used extensively by researchers in the context of RUL prediction and make it easier for comparison of results between models. The formula for each performance metric can be seen in Equation (8) and Equation (9).

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\Delta^{(j)})^2} \tag{8}$$

$$s = \sum_{j=1}^{m} exp(\alpha \left|\Delta^{(j)}\right|) \tag{9}$$

For both equations m is the total number of data samples, $\Delta^{(j)}$ is the error in the RUL prediction of sample j, that is, the RUL predicted by the model minus the real RUL at sample j, α is $\frac{1}{10}$ if the RUL prediction is over-estimated and $\frac{1}{13}$ if the RUL prediction is under-estimated. The reasoning for this imbalance is that, in maintenance, an under-estimation of RUL would lead to sub-optimal maintenance, but an over-estimation would lead to asset failure, which is a situation much more costly for companies than the alternative. The RMSE is a performance metric that has been commonly used in RUL estimation research and accurately represents the overall error of estimation and is symmetric, unlike the Score function.

The behaviour of both performance metrics are represented in Figure 3, which clearly shows the symmetric and asymmetric behaviour of each metric.

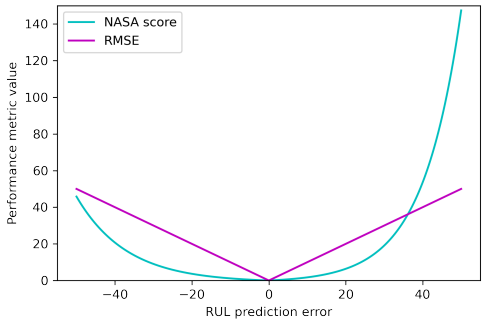


Figure 3: Behaviour of the performance metrics used in evaluation

3.3. Model parameters selection

During the parameter optimization, the parameter being tested changes while the remaining parameters are fixed. At each iteration the loss function and the RUL prediction graphs are analysed to help guide the choice of parameters. The loss function chosen is the MSE function, a common function for regression problems. While this does not guarantee an optimal solution, this process is not too time consuming and improves the performance of the model.

3.3.1 Layer topology

The layer topology refers to the number of layers of the network and the number of neurons of each network. In general, a larger and deeper network can model more complex data. As a drawback, these layers take longer to train and test, while not always presenting better performance. After a relevant number of combinations is tried and a good performance is obtained the optimization of the layer topology is stopped and the next parameter is tested.

3.3.2 Dropout

Dropout is a powerful technique that helps a model avoid over fitting when training a network and increases generalization and robustness. It causes the network to randomly drop a subset of neurons and their respective connections from the NN. A dropout of 0.5 means there is a 50% probability of dropping each neuron in a specific layer. By removing neurons from the network, we force the network to not become overly dependent on any specific neuron [21]. The dropout is applied uniformly to all the hidden layers and input layer, and the values tested are $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$.

[1] https://www.kaggle.com/

3.3.3 Early stopping

The early stopping technique is technique that automatically stops the training of the model when the performance starts degrading. This is a very useful technique used to choose the number of epochs of training. The early stopping technique used monitors the loss function values of the validation data at each epoch and stops the training if the loss function value does not improve over a period of 20 consecutive epochs. Using this technique not only helps in selecting an optimal number of epochs to optimize performance, it also reduces the training time.

3.3.4 Sliding time window size

The sliding time window size refers to the number of samples that the model uses for each prediction. This influences how much information is accessed for each prediction. This is a parameter used when preparing the data before training the model. In general, the larger the time window size, the more information is used, and the better are the model predictions. The base BiLSTM model uses a sliding time window with size 50 and step size 1. This means that, during the preprocessing of data, sequences of size 50 are generated, if m is the number of samples in the input data, then $m - 50 + 1$ sequences are generated. Increasing the time window size can improve performance but also increases the training time significantly. The time window sizes tested were 25, 50 and 100.

3.4. Feature selection and analysis

3.4.1 Feature selection

Feature selection is an important step to improve the quality of the data, therefore, before training the models the dataset is analysed to find features that have missing values and features that have only constant values. The features with these characteristics are then removed from the dataset.

3.4.2 Feature analysis

Feature analysis was performed after the parameters selection on the optimized BiLSTM model, and is based on the results of two different techniques. The Pearson correlation matrix and the Importance analysis from the RF model. The performance of the models is evaluated for three different feature selections. One model has all the features, another has features selected based on the Pearson correlation coefficients and the last model has features selected based on the Importance analysis from the RF base model. The results are then compared to determine the effects on performance based on the chosen techniques, regarding the proposed BiLSTM model and the chosen dataset.

4. Dataset description and analysis

The dataset chosen was obtained from the Prognostics Data Repository[2], designed for the development of prognostic algorithms and provided by NASA.

The main advantages in using this dataset is not only its quality, but its quantity as well. The N-CMAPSS dataset provides full run-to-failure trajectories of a fleet of turbofan engines. Having the time-to-failure is crucial for the development of ML models and it's not typically available from real life applications. This is both due to the rarity in failures occurring from excessive maintenance, as well as from the inherent sensitive nature of failures that inhibit companies from publicly sharing their assets' data[1].

4.1. Data description

The N-CMAPSS dataset is divided into 10 sub-datasets with differences between each other, mainly in regards to the number of engines and the failure modes in each sub-dataset. Each data file is divided into a development dataset and a test dataset. This is the splitting of data used when training and testing the ML algorithms. Both datasets contain 6 types of variables: the operative conditions, the measured sensors signals, the virtual sensors, the engine health parameters, the RUL label, and auxiliary data.

The measured signals are estimates of the measured physical properties at different points along the engine. These measurements were obtained from the simulation model. There are a total of 14 different variables, including physical properties such as flow, speed, temperature and pressure. These variables simulate the condition monitoring measurements that real life applications perform.

Table 1: Sensor measurements - x_s

Id	Description	Unit
Wf	Fuel flow	pps
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T48	Total temperature at HPT outlet	°R
T50	Total temperature at LPT outlet	°R
P15	Total pressure in bypass-product	psia
P2	Total pressure at fan inlet	psia
P21	Total pressure at fan outlet	psia
P24	Total pressure at LPC outlet	psia
Ps30	Static pressure at HPC outlet	psia
P40	Total pressure at burner outlet	psia
P50	Total pressure at LPT outlet	psia

The virtual sensors are estimates of the unobservable properties that are not part of the condition

[2]https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/turbofan-2

monitoring signals, calculated from the measured signals. There are also a total of 14 variables, including properties such as temperature, pressure, flow, and stall margin.

4.2. Data sampling

The raw data is sampled in seconds. This means that the model has very a fine degree of information for training. However, a common problem, specially with complex models, is that the bigger the size of the data file, the longer it takes to train and test the models. Observing our specific dataset and the behaviour of the features over time, we can conclude that increasing the sampling size of the data is beneficial to reduce the size of data while still capturing the features of the data. For this purpose a sampling size of ten minutes was chosen.

4.2.1 Feature scaling

Normalization is a transformation of data from its original range of values to a specific desired range of values. The method applied in this step is the min-max normalization. This method re-scales the range of the data to the range of values of [0, 1], the formula is shown in Equation (10) . Min-max normalization was chosen as it is a simple method of feature scaling that has been previously used to great success in the literature [23] [32]. The min-max normalization was used only for the LSTM model and the BiLSTM model, as with the RF and XGBoost model, normalizing the data has no effects, since the models are independent of the scale of data.

$$x' = \frac{x - min\,(x)}{max\,(x) - min\,(x)} \tag{10}$$

5. Results and Discussion

5.1. Benchmark models results

Table 2 shows the results of the base models for both evaluation metrics,RMSE and the NASA Score, the same results are seen in Figure 4(a) and Figure 4(b) through bar plots for better visualization and comparison, while Figure 5 show the RUL predictions of the base models. The LSTM model shows the best performance for both evaluation metrics, however, looking at the RUL predictions of the BiLSTM model we see a desirable behaviour, that is not seen in the RF and the XGBoost model, but can be seen also in the LSTM model. The initial RUL prediction of the BiLSTM model is poor, but the model is capable of quickly adjusting and approximating the real RUL curve very accurately. It is worth mentioning that, even though the BiLSTM model has worst performance on the RMSE metric, it has a better performance on the Score metric, compared with the RF and the XGBoost model. This is a consequence of the asymmetry of the Score metric.

Table 2: Base models results on test data DS01: RF, XGBoost, LSTM, BiLSTM

Base Models	Score	RMSE
RF	5170	7.657
XGBoost	5004	7.505
LSTM	**3368**	**6.294**
Bi-LSTM	4421	9.275

5.2. Model parameters optimization

The parameter optimization was performed with the dataset DS01, which makes the final BiLSTM optimized for this dataset, but not necessarily for the remaining datasets, given that the datasets are different, specifically in the failure modes. The optimization of a BiLSTM model for each dataset was not performed as it is too time consuming, and given the time constraints. This is something that can be improved upon to obtain an optimized model for each subset of the data, however, given completely new data, the model will most likely still show worse performance than seen in our results.

First, several network architectures were tried starting with the base model. The results of the evaluation metrics on the test dataset DS01 for each network is shown in Table 3. The best performance is given by the network B(256,256,64) with a RMSE of 5.015 and a Score of 2307. This network has a contracting form, this is, the initial layers have a larger number of neurons and the last layers have less neurons. This structure seems to work better for our situation than a constant form or an expanding one, as the results indicate.

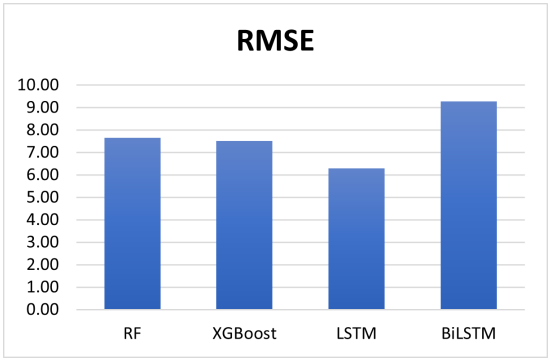
Table 3: Layer topology tuning of BiLSTM network

BiLSTM Network	Score	RMSE
B(64,32,16,8)	4421	9.275
B(64,64,64,64)	2701	5.446
B(256,256,256,256)	8556	10.75
B(64,256,32,32)	9755	11.61
B(128,128,64,64)	2886	5.744
B(256,256,64,64)	2923	5.676
B(256,256,64)	**2307**	**5.015**
B(128,128,64)	3750	6.632

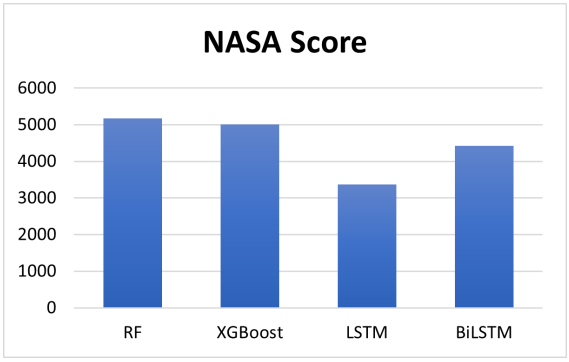
The remaining parameter selection showed that the base BiLSTM model parameters were already the best performing parameters. The final BiLSTM model has a Dropout of 0.5, learning rate of 0.001, batch size of 512, training/split validation of 10% and sliding time window size of 50.

5.3. Feature analysis results

Three possible models were tested on the DS01 dataset, one model was trained with all the fea-



(a) RMSE of the base models: RF, XGBoost, LSTM and BiLSTM



(b) NASA score of the base models: RF, XGBoost, LSTM and BiLSTM

Figure 4: Comparison between the performance of the base models on test data DS01

tures, another was trained with the reduced features based on the Pearson correlation matrix, the last model was trained with the reduced features based on the importance analysis from the RF base model. The results are presented in table 4. The best performance was given by the reduced features based on the importance analysis. Also the network with reduced features based on the correlation matrix performed worse than using all the features. This means that the dependencies of the dataset cannot be well represented through the Person correlation coefficient.

Table 4: Results of the feature selection based on the feature analysis proposed

Feature Selection	Score	RMSE
All features	2307	5.015
Correlation matrix	3851	6.91
RF Importance	**1852**	**4.398**

5.4. Performance of final BiLSTM model on all sub datasets

The optimized BiLSTM network was applied to the remaining datasets and the results are shown in Table 5. For these results all features are used to better compare the results between models, as the feature analysis was performed only for DS01. Since the model was optimized using the DS01 dataset, it has a poorer performance on the remaining sub datasets, as was expected.

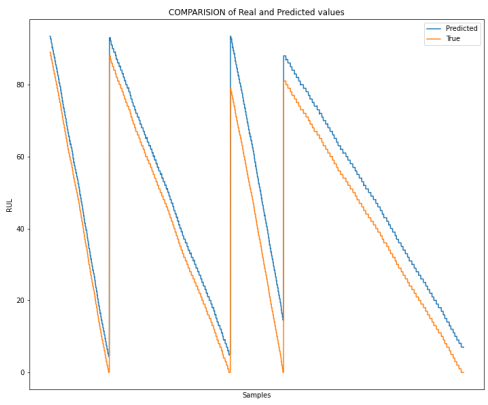
It is clear that the BiLSTM network is capable of providing very good RUL predicitions, as seen from the RUL prediction of DS01, but the model is not directly transferable to the remaining datasets. A new hyper parameter optimization would need to be performed to achieve better results on the remaining datasets.

Lastly the results of the BiLSTM model applied to the complete dataset simultaneously. In regards to the RMSE the result of the full dataset and the mean of the results of each dataset is not too different, with a RMSE of 14.08 and 13.67 respectively. The Score metric can be compared by summing all the Score values of each sub dataset obtained in Table 5, which gives 207676. This is a better Score value than the one obtained in Table 6 of 284079. We can conclude that the BiLSTM model applied to the full dataset predicts more over estimations than when applying the model on each dataset individually.

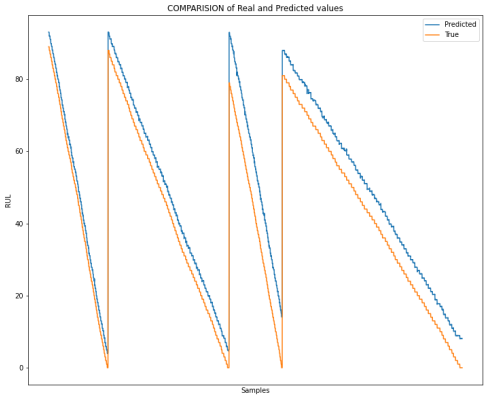
5.5. Comparison with results of other models applied to the CMAPSS dataset

As far we know there are no other ML models in the literature applied to the N-CMAPSS dataset to compare with the current results. There are, however, results from other models applied to the CMAPSS dataset, the precursor to the N-CMAPSS dataset used in this work. Even though the datasets have significant differences they have the same origin.

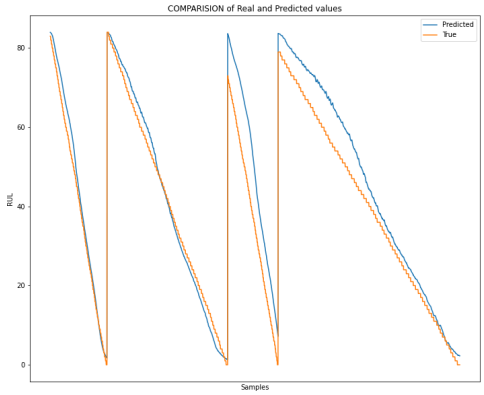
Focusing on the results of the proposed model for the dataset DS01, as this was the model used for the parameter optimization, we can see a significant improvement on the RMSE obtained with the proposed model when compared to other models. Specifically, an improvement of 63.3% on the best model applied to the CMAPSS dataset. The benchmark models also show better results in terms of RMSE, even with comparatively less complex ML algorithms than those used on the CMAPSS dataset. These results show the improvement to the quality of the N-CMAPSS dataset, when compared to the CMAPSS dataset. It also shows the benefits of using larger and finer dataset on the asset being studied, for ML algorithms. The Score performance index obtained is worse, not because the RUL predictions are less accurate, but because the number of samples predicted is much higher. Since the Score function is a sum of the error, without us-



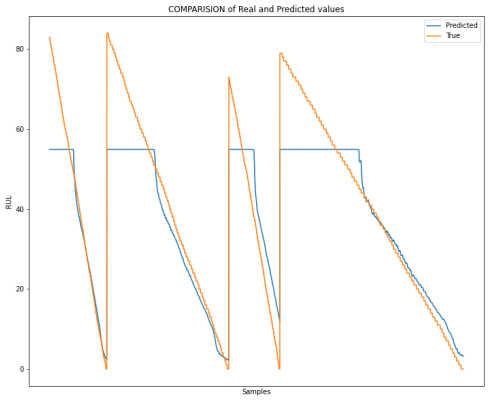
(a) RUL prediction of the base model RF



(b) RUL prediction of the base model XGBoost



(c) RUL prediction of the base model LSTM



(d) RUL prediction of the base model BiLSTM

Figure 5: RUL prediction of the base models on test data DS01

Table 5: Results of the BiLSTM network on all sub datasets

Dataset	DS01	DS02	DS03	DS04	DS05	DS06	DS07	DS08a	DS08c	Mean
Score	2307	5657	7369	89725	24988	19317	50292	4356	3665	23075
RMSE	5.015	13.63	8.202	23.58	17.77	17.64	21.19	5.999	10	13.67

Table 6: Results of the BiLSTM network on the complete N-CMAPSS dataset

Dataset	Full N-CMAPSS dataset
Score	284079
RMSE	14.08

Table 7: RMSE of other models with the CMAPSS dataset

ML method	FD001	FD002	FD003	FD004
MLP [2]	37.56	80.03	37.39	77.37
SVR [2]	20.93	42.00	21.05	45.32
RVR [2]	23.80	31.30	22.34	34.34
CNN [2]	18.45	30.29	19.82	29.16
MODBNE [27]	15.04	25.05	12.51	58.66
LSTM [33]	16.14	24.49	16.18	28.17
BiLSTM [22]	13.65	23.18	13.74	24.86
DCNN [13]	12.61	22.36	12.64	23.31
CNN-BiLSTM [32]	12.58	**19.34**	**12.18**	**20.03**
DAG [11]	**11.96**	20.34	12.46	22.43

ing the same amount of samples in all the models, it is not possible to accurately compare the results.

6. Conclusions

The present work explores RUL estimation using ML algorithms in the context of Predictive Maintenance. We identify the main reason for industries to suffer of inefficient maintenance to be the lack of knowledge regarding the real condition of assets and their need for maintenance.

The RF, XGBoost and LSTM model are suitable benchmark models given their ease of use and, concerning the LSTM model, its similarity with the BiLSTM model. The BiLSTM model is also suitable to explore further given its capabilities and potential for great performance results. It is, how-

Table 8: Proposed models results on the N-CMAPSS dataset DS01: base RF, XGBoost and LSTM model and final BiLSTM model

ML Method	Score	RMSE
RF	5.170×10^3	7.657
XGBoost	5.004×10^3	7.505
LSTM	3.368×10^3	6.294
Bi-LSTM	$\mathbf{1.852 \times 10^3}$	**4.389**

ever, a complex model that requires a lot of time to train and test, given the extensive computations required. Without the access to a GPU to perform these computations, it would not have been possible to optimize the BiLSTM model, either through an automatic or manual method, and achieve good results.

Regarding limitations on the evaluation metrics chosen, the RMSE was found to be suitable in comparing between the models proposed in this work and also with other models found in the literature. However, the Score metric is only suitable to compare between the models proposed, as it depends on the number of samples of the input data. It is not possible to compare this performance metric with models outside the present work, unless they use the same number of input samples.

The feature analysis proposed showed that, for the dataset DS01, using the Pearson correlation coefficients for feature selection degrades performance but using the importance analysis from the RF base model improved the results achieving the best performance of a **Score of 1852** and a **RMSE of 4.398**, an improvement of 80.3% and 87.7%, respectively, when comparing with the BiLSTM model with no feature selection. Since the importance analysis was applied only to the DS01 dataset, it is not possible to see the effects of feature selection on the remaining datasets.

The results of the BiLSTM model applied to the remaining subsets of data show that the performance on the new, unseen data is significantly worse than the performance for the sub dataset DS01. From these results we can conclude that, even though the BiLSTM has the capability of extremely accurate RUL predictions, it requires proper parameter optimization. Additionally, manually optimizing the parameters requires expert experience, and does not guarantee optimal solutions. We can identify the layer topology as the main parameter to explore, given its influence on the results obtained. The proposed BiLSTM model was also applied to the full dataset, but the results of this approach were not satisfactory.

Lastly the results of the proposed models are compared with state-of-the-art model applied to the CMAPSS dataset the predecessor of the N-CMAPSS dataset. It is a limitation of this work that it is not possible to compare the proposed models with other models applied to our dataset, however none were found given the recent nature of the dataset. Compared with models applied to the CMAPSS dataset, all the proposed models outperform, when applied to the DS01 sub dataset, even being less complex models. This is a consequence of the quality of the dataset being higher. The main contributing factor identified is that the N-CMAPSS has significantly more data samples, that increase the models accuracy in RUL prediction.

In conclusion, several limitations and challenges were identified with the chosen dataset and proposed framework, however the main objectives of accurately predicting RUL, in the context of PdM, with ML algorithms, and the comparison of the proposed models with other ML models from the literature, were achieved. The results obtained show promise for further research and in providing critical information to support decision making for predictive maintenance strategies.

6.1. Future work

Most of the proposals stem from tasks that were intended to be performed but could not be due to time constraints. Regarding the feature analysis, we propose an in-depth study of the correlations between the features and the RUL for all subsets of data available. By better understanding this correlation, we can improve on the feature selection process and improve the results on RUL prediction for all subsets of data. Regarding the framework, the future work proposed focuses on two aspects. First, applying an automatic parameter selection method in order to confidently achieve optimal solutions when selecting the parameters of the models. Second, perform the training of models several times in order to calculate average, median, maximum and minimum, as well as the frequency distribution of the performance results in order to reduce the effect of variability on the results presented. With this two improvements we could achieve better predicting models and a good degree of confidence on the results obtained.

Acknowledgements

This work is supported by my supervising team, at IDMEC, Instituto Superior Técnico (IST), Universidade de Lisboa (ULisboa), and my supervising team at Axians, Portugal. Their contribution was indispensable for this work. I would like to personally thank my advisors, Professor João Sousa from IST, Daniela Moniz and Margarida Solas from Axians for all the time, patience and support they gave to this work. It wouldn't have been possible without your guidance.

References

[1] M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1), 2021.

[2] G. S. Babu, P. Zhao, and X.-L. Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications*, pages 214–228. Springer, 2016.

[3] M. Calabrese, M. Cimmino, F. Fiume, M. Manfrin, L. Romeo, S. Ceccacci, M. Paolanti, G. Toscano, G. Ciandrini, A. Carrotta, M. Mengoni, E. Frontoni, and D. Kapetis. Sophia: An event-based iot and machine learning architecture for predictive maintenance in industry 4.0. *Information*, 11(4), 2020.

[4] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19), 2020.

[5] P. R. d. O. da Costa, A. Akcay, Y. Zhang, and U. Kaymak. Attention and long short-term memory network for remaining useful lifetime predictions of turbofan engine degradation. *International Journal of Prognostics and Health Management*, 10(4), 2019.

[6] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[7] H. M. Hashemian. State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement*, 60(1):226–236, 2010.

[8] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[9] O. Janssens, M. Loccufier, and S. Van Hoecke. Thermal imaging and vibration-based multisensor fault detection for rotating machinery. *IEEE Transactions on Industrial Informatics*, 15(1):434–444, 2019.

[10] S. Ji, X. Han, Y. Hou, Y. Song, and Q. Du. Remaining useful life prediction of airplane engine based on pca–blstm. *Sensors*, 20(16), 2020.

[11] J. Li, X. Li, and D. He. A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction. *IEEE Access*, 7:75464–75475, 2019.

[12] X. Li, Q. Ding, and J.-Q. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018.

[13] X. Li, Q. Ding, and J.-Q. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering System Safety*, 172:1–11, 2018.

[14] G. Liu and J. Guo. Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338, 2019.

[15] Z. Liu, W. Mei, X. Zeng, C. Yang, and X. Zhou. Remaining useful life estimation of insulated gate biploar transistors (igbts) based on a novel volterra k-nearest neighbor optimally pruned extreme learning machine (vkopp) model using degradation data. *Sensors*, 17(11), 2017.

[16] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar. Prediction of remaining useful lifetime (rul) of turbofan engine using machine learning. In *2017 IEEE International Conference on Circuits and Systems (ICCS)*, pages 306–311, 2017.

[17] R. K. Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[18] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1–9, 2008.

[19] S. Selcuk. Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9):1670–1679, 2017.

[20] Z. Shi and A. Chehade. A dual-lstm framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, 205:107257, 2021.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[22] J. Wang, G. Wen, S. Yang, and Y. Liu. Remaining useful life estimation in prognostics

using deep bidirectional lstm neural network. In *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pages 1037–1042, 2018.

[23] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi. An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation. *Computers in Industry*, 115:103182, 2020.

[24] J. Yan, Y. Meng, L. Lu, and L. Li. Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. *IEEE Access*, 5:23484–23491, 2017.

[25] H. Yang, F. Zhao, G. Jiang, Z. Sun, and X. Mei. A novel deep learning approach for machinery prognostics based on time windows. *Applied Sciences*, 9(22), 2019.

[26] M.-Y. You, F. Liu, W. Wang, and G. Meng. Statistically planned and individually improved predictive maintenance management for continuously monitored degrading systems. *IEEE Transactions on Reliability*, 59(4):744–753, 2010.

[27] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE transactions on neural networks and learning systems*, 28(10):2306–2318, 2016.

[28] C. Zhang, X. Yao, J. Zhang, and H. Jin. Tool condition monitoring and remaining useful life prognostic based on a wireless sensor in dry milling operations. *Sensors*, 16(6), 2016.

[29] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li. Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement*, 152:107377, 2020.

[30] W. Zhang, D. Yang, and H. Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3):2213–2227, 2019.

[31] Y. Zhang, R. Xiong, H. He, and M. G. Pecht. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, 67(7):5695–5705, 2018.

[32] C. Zhao, X. Huang, Y. Li, and M. Yousaf Iqbal. A double-channel hybrid deep neural network based on cnn and bilstm for remaining useful life prediction. *Sensors*, 20(24), 2020.

[33] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95, 2017.

[34] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li. Predictive maintenance in the industry 4.0: A systematic literature review. *Computers Industrial Engineering*, 150:106889, 2020.