Estimating the Remaining Useful Life via Neural Sequence Models: a Comparative Study

Giovanni D'Agostino[1], Alex Falcon[1,2], Oswald Lanz[3], Giorgio Brajnik[1], Carlo Tasso[1], and Giuseppe Serra[1]

[1] University of Udine, Via delle Scienze, 206, 33100 Italy
[2] Fondazione Bruno Kessler, Via Sommarive, 18, 38123 Italy
[3] Free University of Bozen-Bolzano, Piazza Domenicani, 3, 39100 Italy

Abstract. The prediction of the Remaining Useful Life of a machine component represents a strategic problem in predictive maintenance, which may have important consequences for a company. Recent approaches to this problem leverage data-driven methodologies based on deep learning, achieving impressive results. In particular, due to the temporal nature of the sensor measurements detailing the life of a component, neural sequence models are often chosen to automate the feature extraction process. In this paper, we investigate several of these models on a particle filtration system. The experiments performed present the good prediction capabilities of these models, highlighting some of them for their accuracy. Nonetheless, the qualitative analysis shows that when the fault is farther away, most of these models tend to have unstable predictions. These results motivate some future research directions which are discussed in the conclusions.

Keywords: Remaining useful life estimation · Neural sequence models · Deep learning

1 Introduction

Recently, the United States Department of Energy reported that most of the companies in the USA follow a reactive maintenance strategy, that is they wait for a machine component to fail instead of properly maintaining it [1]. To avoid the replacement of extremely costly components, being able to accurately estimate when a failure is going to happen, that is to estimate its Remaining Useful Life (RUL), represents a strategic problem which is often put at the core of predictive maintenance [2].

Commonly, the methods for the RUL estimation task methods are either model-based or data-driven. In the former case, the predictions are made by physical or mathematical models which simulate the degradation of the machine under analysis [3, 4]; however, because of the need for domain expertise and extensive verification, they are highly complex, expensive, and need to be designed in a case-by-case manner. The methods developed with the second methodology rely on historical sensor data to build an approximate degradation model by leveraging handcrafted features [5, 6]. Although powerful and often applicable to heterogeneous domains, they still rely on domain expertise in order to perform the feature engineering step. Recently, data-driven methodologies which use deep learning gained a lot of attention [7, 8], thanks to their automatic feature extraction step, which works directly on raw data, and ease of application to different domains. In particular, neural sequence models are often chosen because they inherently discover hidden patterns in temporally-related data [9, 10].

This paper investigates the prediction capabilities of neural sequence models. The experiments are performed on a public dataset from particle filtration systems [11], which are often deployed in manufacturing companies dealing with food and beverage, semiconductor and electronic components, and many more. The evidences presented in this paper highlight the accuracy of some of these models when modelling the evolution of the health state of the analyzed machine. Nonetheless, the qualitative analysis shows that this prediction is less accurate when the fault is far away.

2 Related Work

Model-based and data-driven methods. The problem of correctly estimating the RUL has been strategic for several decades [12, 13]. Traditional approaches can be divided in model-based and data-driven. The former use mathematical or physical models of the degradation phenomena, e.g. [14, 15], thus requiring an in-depth understanding of the underlying system and the failure modes. Instead, data-driven methods build a degradation model solely based on historical sensor data. Methods based on statistics, e.g. [5, 16], and Artificial Intelligence, e.g. [6, 17], are popular examples. Instead of relying on a deep understanding of the underlying system, data-driven methods leverage handcrafted features which are extracted from the raw data. However, such a feature engineering step can be time consuming and may still rely on domain knowledge.

Deep Learning-based methods. A major advantage of deep learning consists in the automatic extraction of the features, as opposed to handcrafted ones. Initial approaches with these techniques used Multilayer Perceptrons (MLP) to estimate the RUL directly from raw data [18]. However, since the sensor measurements are taken periodically, they likely have temporal dependencies, making neural sequence models a more suitable choice. RNNs were used in [18], yet they can fail at remembering information from long time series. To overcome this issue, memory-based networks were used to store key knowledge over time: for instance, LSTMs [7, 19] and GRUs [20, 21] were often used. More recently, NTMs also showed potential in this field by using a memory bank and learnable operations to access and modify it [8, 22].

3 Methodology

An overview of the methodology followed in this study is shown in Figure 1. In particular, the time series are first sliced into shorter windows, normalized

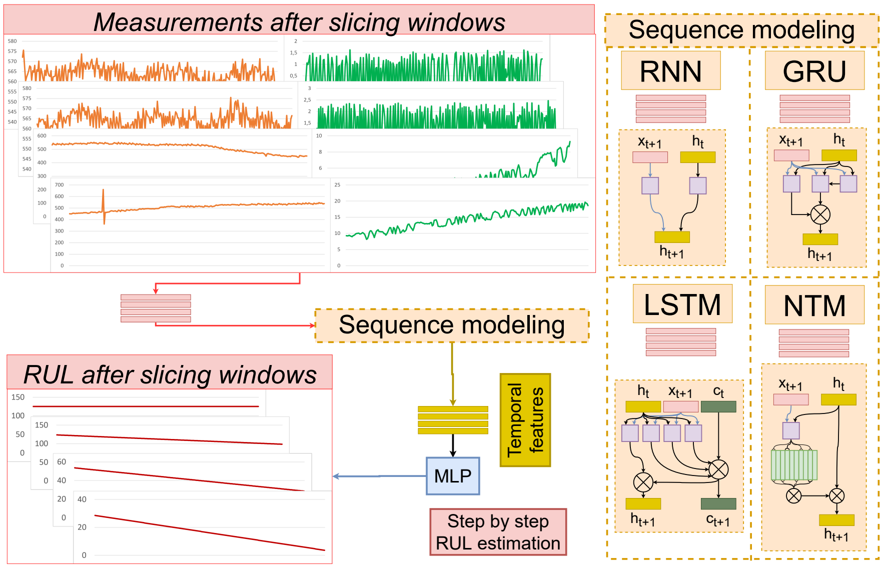

Fig. 1. Graphical overview of our methodology. The series of sensor measurements are first sliced into short windows. A sequence model is used to automatically extract relevant features from the raw data. Finally, a MLP is used to estimate the RUL values.

through MinMax, and then labelled with a piece-wise degradation function [18] with a max RUL of 125, as in [23]. The sequences are then modelled by means of a neural sequence model, including Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), Gated Recurrent Units (GRU), and Neural Turing Machines (NTM). Then, a mapping function between the automatically extracted features and the RUL values is learned by using an MLP. Finally, the network weights are optimized by using the Mean Square Error of the predictions. In the following, these sequence models are briefly described.

Recurrent Neural Network. The data typically considered in the Prognostics and Health Management field is composed of long time series measurements of sensors data. To model temporally-related sequential data and the evolution of its intrinsic characteristics, RNNs have shown good performance in extrapolating hidden patterns in data. RNNs are a class of artificial neural networks which compute the t-th output by using the $t-1$-th output together with the t-th element of the input sequence. RNNs are affected by two problems when calculating the gradient of the cost function over long input sequences: the vanishing gradient problem, in which the value of the gradient gradually converges to zero (thus "vanishing"), and the exploding gradient problem, in which its value tends to infinite (thus "exploding") [24, 25].

Long Short-Term Memory. Due to the length of the sequences considered in the RUL estimation problem, the gradient issues affecting RNNs need to be paid attention to; the LSTM Networks were introduced to mitigate such issues.

The flow of information in an LSTM network is controlled by three gates, called *input*, *output*, and *forget*. Moreover, two memory states, called *hidden* and *cell*, are recurrently updated by the LSTM. In particular, the input gate decides whether to update the cell state by using the current input, the forget gate decides whether to keep or forget the information from the previous hidden state, and the output gate decides how to update the hidden state given the information stored in the cell state.

Gated Recurrent Unit. GRUs were introduced in [26] as a variant of the LSTM networks architecture; in fact, GRUs have only two gates: the reset and the update gate. Differently from LSTMs, GRUs do not possess a cell state, and the reset gate is applied directly to the previous hidden state, therefore performing a similar task as the input and output gates in LSTMs. It follows that GRUs have less training parameters than LSTMs, thus using less memory and executing faster; nonetheless, LSTMs may be more accurate on larger datasets.

Neural Turing Machine. The NTM was originally proposed in [10] and later applied to the predictive maintenance field in [22, 8]. It is inspired by classical Turing Machines: in fact, it comprises a tape-like memory and updates it by means of read and write operations which are guided by a controller. Differently from LSTMs and GRUs, the NTM has an array of memory vectors, therefore enlarging its mnemonic capabilities and possibly reducing the likelihood of overwriting previously learnt concepts. This is also made possible by the usage of learnable read and write operations, which consider contextual information to decide which locations to use and to which extent the information contained therein should be updated.

4 Experimental Results

4.1 Analyzed dataset

The PHM Society 2020 Data Challenge **(PHM20)** [11] public dataset is used to perform the experiments because it offers sensor measurements comprising failures in a particle filtration system, which is often used in food and beverage manufacturing, pharmaceutical industries, etc. In this dataset, the measurements come from an experimental rig. Contaminants in the liquids passing through the system may clog it, and the challenge objective is to anticipate when such an occurrence will happen. In particular, the clogging can be identified when the pressure difference is higher than 20 psi. Each of the 32 experiments (24 for training, 8 for validation) in the dataset include concentration (40%-47.5%) and size (45-53μm, or 63-75μm) of the contaminant particles, and are thousands of steps long with a sampling rate of 10 Hz. For each time step, three measurements are taken: flow rate, upstream and downstream pressures. In addition, we also consider the concentration value and the size of the particles. In this work, the RUL is 0 when the pressure difference becomes higher than 20 psi for the first time. Finally, we use the validation experiments as the test data, and further split the training data with an 80/20 ratio to create a validation set.

Temp. ctx	30	45	60	70	140	210	280	350
RNN	9.3/6.5	9.4/6.7	8.3/5.7	9.2/6.4	9.8/6.8	11.1/8.5	10.7/8.0	11.1/8.0
LSTM	10.4/6.8	10.3/6.8	7.0/4.8	9.3/6.3	7.5/5.2	8.4/5.6	10.2/7.3	8.1/5.8
GRU	**9.0**/6.2	**6.9**/**4.7**	**6.6**/**4.3**	7.8/5.5	**6.2**/**4.4**	6.9/4.6	5.9/4.5	**6.2**/**4.5**
NTM	**9.0**/**5.8**	7.3/4.6	7.1/4.5	**6.8**/**4.4**	6.7/4.5	**5.5**/**3.7**	**5.4**/**3.7**	6.9/5.0

Table 1. 5-runs average RMSE/MAE values on the test set. Overall best is underlined.

4.2 Training settings and model evaluation

The experiments are performed using PyTorch 1.7.1. With our hardware (RTX A5000 and i7-9700K), a training run takes around 50 minutes for the NTM, for which a CUDNN implementation is not currently available, and 6-8 minutes for the other models. We used the following hyperparameters: batch size 100, learning rate 5e-3, 64 neurons in the MLP, and all the hidden sizes are set to 64. In this study, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) were chosen to assess the prediction accuracy of each of the sequence models. A key difference consists in the higher sensitivity of RMSE when it comes to prediction errors which highly deviate from the mean value.

4.3 Quantitative comparison

Since sequence models perform the prediction based on a sequence of observations, varying the size of such a temporal context may highly influence the prediction error. Therefore, an experiment is conducted by using different sizes for it. Three observations can be drawn from Table 1. Firstly, by using shorter contexts, e.g. 30-45, which likely lack of crucial information, all the models make unstable predictions, leading to high RMSE. Secondly, by increasing it, more information is likely to be found, and the prediction error steadily decreases. In particular, by modelling the sequences with a GRU or a NTM and using a context of 280 steps, the lowest error is achieved (5.9 RMSE and 4.5 MAE, and 5.4 RMSE and 3.7 MAE). Lastly, while really long sequences may contain additional and potentially useful information, they are also harder to be modelled: as a consequence, the prediction error increases.

4.4 Qualitative analysis

Figure 2 compares the prediction made by the four models (context of 280) on a full experiment from the test set (groundtruth shown in red). It shows that all the models are highly precise when the RUL is close to 0, indicating that the fault is evident by looking at the sensor measurements. Conversely, the farther from the fault, the higher the uncertainty: this clearly indicates the difficulty of anticipating such an event, although the GRU and the NTM are quite precise, especially if compared to the noisy predictions made by the RNN and the LSTM.

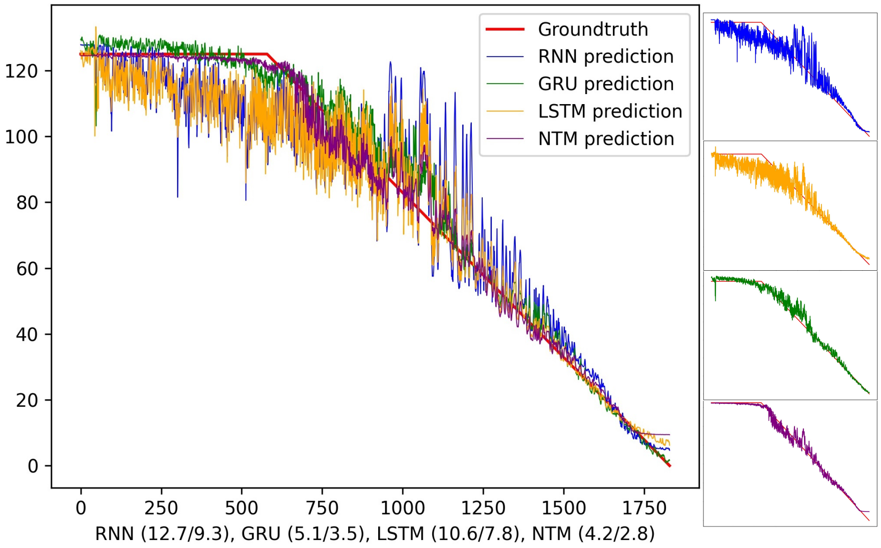

Fig. 2. Predictions made by the four models on a full experiment from the test set. RMSE and MAE values are shown below. Best viewed in color.

5 Conclusions

Being able to predict when a fault may occur in a industrial machine is fundamental. To achieve this goal, a precise predictive model is required and the availability of historical data often shifts the attention to data-driven methodologies, and in particular to the use of deep learning techniques to automatically extract useful features from raw sensor measurements. Given that faults develop over time, in this study we investigated the predictive capabilities of several neural sequence models in a particle filtration system. Quantitatively, we observed that all the models achieve modest prediction accuracy, although the GRU and the NTM perform better than the others. Considering that these models are designed with a lot of care on the technique used to access their memory state, further research is needed to improve the operations used to access and update the memory, while at the same time strive for more attention on the contents put into it. Qualitatively, we presented evidence that all the models are accurate when the fault is close, but they become more and and more uncertain the farther it is. Consequently, neural sequence models may become aware of a fault when it is far too close, therefore it may be difficult to perform a preemptive action. Therefore, future work may also focus on improved training procedures which put more emphasis on detecting when the fault starts to develop, which represents a critical point for a predictive system. Finally, Transformer-based approaches [27] could also be used for future research on RUL estimation.

References

1. U. S. D. of Energy, Operations & maintenance best practices: A guide to achieving operational efficiency, Tech. rep. (2020).
URL https://www.energy.gov/sites/prod/files/2020/04/f74/omguide_complete_w-eo-disclaimer.pdf
2. A. K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, MSSP 20 (7) (2006) 1483–1510.
3. W. Teng, C. Han, Y. Hu, X. Cheng, L. Song, Y. Liu, A robust model-based approach for bearing remaining useful life prognosis in wind turbines, IEEE Access 8 (2020) 47133–47143.
4. N. Li, Y. Lei, T. Yan, N. Li, T. Han, A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability, IEEE Transactions on Industrial Electronics 66 (3) (2018) 2092–2101.
5. C. Ordóñez, F. S. Lasheras, J. Roca-Pardiñas, F. J. de Cos Juez, A hybrid arima–svm model for the study of the remaining useful life of aircraft engines, Journal of Computational and Applied Mathematics 346 (2019) 184–191.
6. Y. Pan, R. Hong, J. Chen, W. Wu, A hybrid dbn-som-pf-based prognostic approach of remaining useful life for wind turbine gearbox, Renewable Energy 152 (2020) 138–154.
7. J. Xia, Y. Feng, C. Lu, C. Fei, X. Xue, Lstm-based multi-layer self-attention method for remaining useful life estimation of mechanical systems, Engineering Failure Analysis 125 (2021) 105385.
8. A. Falcon, G. D'Agostino, O. Lanz, G. Brajnik, C. Tasso, G. Serra, Neural turing machines for the remaining useful life estimation problem, Computers in Industry 143 (2022) 103762.
9. S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
10. A. Graves, G. Wayne, I. Danihelka, Neural turing machines, arXiv preprint arXiv:1410.5401 (2014).
11. P. Society, Phm society 2020 data challenge, https://phm-europe.org/data-challenge-2020 (last accessed: 23 April 2022) (2020).
12. C. S. Byington, S. E. George, G. W. Nickerson, Prognostic issues for rotorcraft health and usage monitoring systems, A Critical Link: Diagnosis to Prognosis (1997) 93.
13. F. L. Greitzer, L. J. Kangas, K. M. Terrones, M. A. Maynard, B. W. Wilson, R. A. Pawlowski, D. R. Sisk, N. B. Brown, Gas turbine engine health monitoring and prognostics, in: International Society of Logistics (SOLE) 1999 Symposium, Las Vegas, Nevada, Vol. 30, 1999, pp. 1–7.
14. N. Bolander, H. Qiu, N. Eklund, E. Hindle, T. Rosenfeld, Physics-based remaining useful life prediction for aircraft engine bearing prognosis, in: Annual Conference of the PHM Society, Vol. 1, 2009.
15. A. Coppe, M. J. Pais, R. T. Haftka, N. H. Kim, Using a simple crack growth model in predicting remaining useful life, Journal of Aircraft 49 (6) (2012) 1965–1973.
16. Y. Zhou, M. Huang, Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and arima model, Microelectronics Reliability 65 (2016) 265–273.
17. H.-Z. Huang, H.-K. Wang, Y.-F. Li, L. Zhang, Z. Liu, Support vector machine based estimation of remaining useful life: current research status and future trends, Journal of Mechanical Science and Technology 29 (1) (2015) 151–163.

18. F. O. Heimes, Recurrent neural networks for remaining useful life estimation., in: ICPHM, 2008.
19. S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long short-term memory network for remaining useful life estimation, in: ICPHM, IEEE, 2017, pp. 88–95.
20. M. Baptista, H. Prendinger, E. Henriques, Prognostics in aeronautics with deep recurrent neural networks, in: PHM Society European Conference, Vol. 5, 2020.
21. Q. Luo, Y. Chang, J. Chen, H. Jing, H. Lv, T. Pan, Multiple degradation mode analysis via gated recurrent unit mode recognizer and life predictors for complex equipment, Computers in Industry 123 (2020) 103332.
22. A. Falcon, G. D'Agostino, G. Serra, G. Brajnik, C. Tasso, A neural turing machine-based approach to remaining useful life estimation, in: ICPHM, 2020, pp. 1–8.
23. K. Ince, E. Sirkeci, Y. Genç, Remaining useful life prediction for experimental filtration system: A data challenge, in: PHM Society European Conference, Vol. 5, 2020.
24. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (2) (1994) 157–166.
25. S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001).
26. K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259 (2014).
27. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.