A Transformer-based approach to Remaining Useful Life Estimation Problem

Mattia D'Urso
mattiadu[at]edu.aau.at

Abstract

The work reported in this paper is part of the Master's thesis of the author. The thesis aims to develop a predictive maintenance system using time series and deep learning methods for electronic testing equipment at Infineon Technologies AG. The final approach introduced is evaluated in this paper using the C-MAPSS benchmark dataset to compare it with Remaining Useful Life Estimation state-of-the-art methods. The code is available at `https://github.com/mattiadurso/Predictive-Maintenance`.

Keywords: Time Series, Deep Learning, Time Series Forecasting, Remaining Useful Life.

1 Introduction

Remaining Useful Life Estimation (RUL) is a well-known problem in which the goal is to predict the time to the next failure of a mechanical or electronic component. This knowledge enables more accurate maintenance than run-to-fail or preventive approaches by being applied just before the component failure and not too late or too soon, respectively. Historically, many statistical methods were proposed to address the forecasting problems [11], but nowadays Deep Learning (DL) models outperform statistical and traditional Machine Learning (ML) models in many fields [3, 14], including predictive maintenance (PdM), in case enough high-quality historical data is available. DL methods are more flexible on data requirements and can also handle multivariate inputs and outputs. Recently, many approaches have been used to solve the PdM problem. Among these, there are the evergreen Multi-Layer Perceptron (MLP) and the Convolutional Neural Network (CNN) [14]. The former method is also the base for the design of the Recurrent Neural Network (RNN) models like the Long- and Short-Term Memory (LSTM).

2 Related work

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, generated by the homonym system [7], is the selected dataset to evaluate the approach introduced. Some approaches tested on this benchmark in the last five years are based on CNNs and RNNs architectures. Among the first are the deep CNN (DCNN) proposed by Li et al. [9], the distributed attention based temporal convolutional network (DA-TCN) by Song et al. [15], and the temporal deep degradation network (TDDN) by Qin et al. [12]. The latter approach, more popular for sequences, includes the bi-directional LSTM (Bi-LSTM) by Wang et al. [17], the LSTM plus neural Turing machine by Falcon et al. [6], the Bi-LSTM with attention by Liu et al. [10] and the LSTM with filtering by Asif et al. [1].

3 Proposed model

In this paper, an architecture based on Transformer [16] is proposed. While it is an architecture made of two parts, the encoder, and the decoder, this work uses only the first part. The choice of an Attention-based architecture is motivated by incredible performances shown in the last years. Figure 1 shows the architecture introduced in this paper and its three main parts. In the first part, the input is resized to a larger space [2] to enlarge the degrees of freedom of the latent space, then the Positional Encoding (PE), described by Vaswani

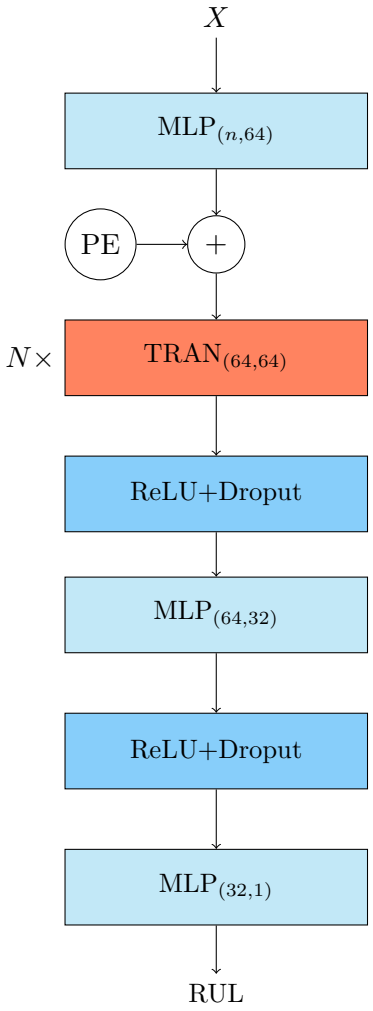

Figure 1: Transformer-based model architecture, where X is the input. For each layer the input and the output size used are indicated.

et al. [16], is added. The second part consists of N (tested up to 4) vanilla Transformer-encoder layers. In the third part, an MLP layer followed by a ReLU activation function and a Dropout layer computes the input and reduces its size. The last step is repeated twice. Notice that, it is possible to use this architecture for multi-class classification too by changing the final output size.

4 Experimental Results

4.1 C-MAPSS Dataset

As reported in Section 2, the chosen benchmark is the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, generated by the homonym system [7]. The dataset consists of 23 different sensor measurements for each of the four sub-datasets called FD001, FD002, FD003, and FD004. In this work only the FD001 is used. According to [6], some constant features ($\sigma^2 \approx 0$) can be removed. Specifically, the considered features are sensors 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 20, and 21. In this work, the degradation is assumed to be linearly decreasing, and not piecewise like in other works like [6]. Figure 2 shows the expected degradation behavior in green.

4.2 Experimental Setup

The resulting data is scaled with a Min-Max scaler defined as

$$X' = \frac{X - \max(X)}{\max(X) - \min(X)},$$

where X' is the resulting scaled data, X represents the raw data and $\max(\cdot)$ and $\min(\cdot)$ denote the maximum and the minimum function, respectively. Then, data is organized following a sliding window approach. The hyper-parameters used are reported in Table 1.

Table 1: Hyper-parameters description.

Hyper-parameters	value
Window lenght	64
Batch size	40
Hidden size	64
Epochs	100
Learning rate	$10^{\{-2,-3,-4\}}$
Weight decay	$0, 10^{\{-3,-5\}}$
Dropout	0, .2, .4, .6

To train the model and to achieve optimal results, the Mean Squared Error (MSE) loss function was used. It is defined as

$$MSE = \mathbb{E}\big[(A - P)^2\big],$$

where A denotes the actual values, P indicates the predicted values and $\mathbb{E}[\cdot]$ represents the mathematical expectation function. The optimizer and the scheduler used are, respectively, Adam [8] and ReduceOnPlateau[1].

[1]More info at pytorch.org

4.3 Metrics

Since this work is comparable (dataset, models, and aim) to Serradilla et al. [14], the metrics are chosen accordingly. The first metric is the Root Mean Squared Error (RMSE) which is a function measuring the distance between the actual, and the predicted values defined as

$$RMSE = \left[\mathbb{E}\left[(A-P)^2\right]\right]^{\frac{1}{2}},$$

where A are the actual values and P are the predicted values. The second metric is the Scoring function (Score) proposed by Saxena et al. [13], and it is an error function designed to favor a more conservative prediction. The function is defined as

$$S = \sum_{t=1}^{N} s_t,$$

where

$$s_t = \begin{cases} e^{-\frac{h_t}{13}} - 1, \text{ if } h_t < 0 \\ e^{\frac{h_t}{10}} - 1, \text{ if } h_t \geq 0 \end{cases},$$

and where $h_t = A_t - P_t$, A_t indicates the actual values, P_t denotes the predicted values at time t.

4.4 Results

The final results show that the proposed architecture with $N = 4$ Transformer-encoder layers achieves the best RMSE score and the third-best Score among the compared solutions. Surprisingly the model with $N = 2$ Transformer-encoder layers achieves the second-best Score and RMSE.

Figure 2 shows an example of the forecasting capabilities of the TRAN_4 model. The considered time series belong to the test set, hence it is never seen by the model. The green line represents the assumed linear degradation, the orange line is the trend of the prediction, the blue line is the inference of the model, and the dashed line indicates the failure of the machine. A conservative prediction, as the one in Figure 2, is exactly the target to achieve since it allows the maintenance few cycles before they fail.

Table 2: The table summarizes the results achieved by the proposed architectures compared to the literature. The best results are in bold, and the proposed architectures' results are underlined. The subscript indicates the number of the Transformer-encoder layer.

Architecture	Year	Score	RMSE
DCNN[9]	2018	2.74×10^2	12.61
BiLSTM[17]	2018	2.95×10^2	13.65
LSTM+NTM[6]	2020	2.42×10^2	12.50
DA-TCN[15]	2020	2.29×10^2	11.78
BiLSTM+Att.[10]	2021	2.42×10^2	12.50
TDDN[12]	2022	2.14×10^2	9.47
LSTM+Filter.[1]	2022	$\mathbf{1.00 \times 10^2}$	7.78
TRAN_1	2022	$\underline{1.85 \times 10^2}$	$\underline{7.87}$
TRAN_2	2022	$\underline{1.47 \times 10^2}$	$\underline{7.57}$
TRAN_4	2022	$\underline{1.58 \times 10^2}$	$\underline{\mathbf{7.36}}$

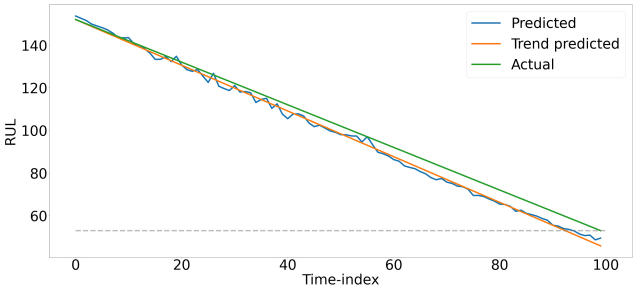

Figure 2: Example of TRAN_4's inference.

5 Conclusions

The tests show a performing model that achieves state-of-the-art performances in the FD001 dataset. It is clear that the core of this architecture, the Transformer-encoder, is extremely powerful and can compete with different state-of-the-art architectures in a large variety of tasks such as image classification [5], object detection [4], machine translation [16], and many others.

For the future, more tests with different architectures (e.g. CNN) in the first and third parts are necessary to find the best-performing combination. It may be interesting to test the model on FD002, FD003, and FD004 datasets and other benchmark datasets like the famous Electricity Transformer Temperature (ETT) [18] to find out where it ranks against other methods.

References

[1] Owais Asif et al. "A Deep Learning Model for Remaining Useful Life Prediction of Aircraft Turbofan Engine on C-MAPSS Dataset". In: *IEEE Access* 10 (2022), pp. 95425–95440.

[2] Yoshua Bengio. "Practical recommendations for gradient-based training of deep architectures". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

[3] Konstantinos Benidis et al. "Deep Learning for Time Series Forecasting: Tutorial and Literature Survey". In: *ACM Computing Surveys (CSUR)* (2018).

[4] Nicolas Carion et al. "End-to-end object detection with transformers". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 213–229.

[5] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[6] Alex Falcon et al. "A Neural Turing Machine-based approach to Remaining Useful Life Estimation". In: *2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2020, pp. 1–8.

[7] D. Frederick, J. DeCastro, and J. Litt. *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)*. Cleveland, Ohio 44135, USA: National Aeronautics and Space Administration (NASA), Glenn Research Center., 2007.

[8] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[9] Xiang Li, Qian Ding, and Jian-Qiao Sun. "Remaining useful life estimation in prognostics using deep convolution neural networks". In: *Reliability Engineering & System Safety* 172 (2018), pp. 1–11.

[10] Yuefeng Liu et al. "Prediction of remaining useful life of turbofan engine based on optimized model". In: *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE. 2021, pp. 1473–1477.

[11] Spyros Makridakis. "A survey of time series". In: *International Statistical Review/Revue Internationale de Statistique* (1976), pp. 29–70.

[12] Yuwen Qin et al. "Remaining Useful Life Prediction Using Temporal Deep Degradation Network for Complex Machinery with Attention-based Feature Extraction". In: *arXiv preprint arXiv:2202.10916* (2022).

[13] Abhinav Saxena and Kai Goebel. "Turbofan engine degradation simulation data set". In: *NASA Ames Prognostics Data Repository* (2008), pp. 1551–3203.

[14] Oscar Serradilla et al. "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects". In: *Applied Intelligence* (2022), pp. 1–31.

[15] Yan Song et al. "Distributed attention-based temporal convolutional network for remaining useful life prediction". In: *IEEE Internet of Things Journal* 8.12 (2020), pp. 9594–9602.

[16] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[17] Jiujian Wang et al. "Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network". In: *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. IEEE. 2018, pp. 1037–1042.

[18] Haoyi Zhou et al. "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting". In: *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*. Vol. 35. 12. AAAI Press, 2021, pp. 11106–11115.