Unsupervised Remaining Useful Life Prediction through Long Range Health Index Estimation based on Encoders-Decoders

Martin Hervé de Beaulieu * **Mayank Shekhar Jha** *
Hugues Garnier * **Farid Cerbah** **

* *Université de Lorraine, CRAN, CNRS UMR 7039, 54506 Vandoeuvre-les-Nancy, France (martin.herve-de-beaulieu@univ-lorraine.fr; mayank-shekhar.jha@univ-lorraine.fr; hugues.garnier@univ-lorraine.fr).*
** *Dassault Aviation 92552 Saint-Cloud, France (Farid.Cerbah@dassault-aviation.com)*

Abstract: The prediction of the Remaining Useful Life (RUL) is a critical step in Prognostics and Health Management (PHM) of systems under degradation. For efficient RUL predictions, most of the Artificial Intelligence (AI-based) methods perform direct mapping between raw sensor data input and RUL data as output targets for supervised learning. However, in the majority of the real-life cases, the available data are either incomplete or unlabeled, which calls for unsupervised methods. This paper proposes such an unsupervised RUL prediction method. Firstly, this method uses an autoencoder model to extract a Virtual Health Index (VHI) from sensors readings. Secondly, an LSTM-based (Long Short-Term Memory) encoder-decoder achieves VHI future predictions. Once the VHI prediction exceeds a pre-determined threshold, the RUL is recursively inferred. Such a method thus allows to obtain RUL predictions without using RUL-labeled data. This method is tested on C-MAPSS dataset. The results obtained are encouraging and offer new perspectives for real industrial applications.

Keywords: Predictive Maintenance, Prognostics and Health Management, State of Health, Autoencoders, Remaining Useful Life, Health Index, C-MAPSS, LSTM

1. INTRODUCTION

Condition-Based Maintenance (CBM) is a maintenance strategy which consists of monitoring the State of Health (SOH) of a system in real time in order to make appropriate maintenance decisions (Jardine et al. (2006)). The prediction of the future SOH of a system is one of the major challenge in CBM and is specifically addressed in the domain of Prognostics and Health Management (PHM) (Lee et al. (2014)). PHM uses two principal indicators, namely the Health Index (HI) and the Remaining Useful Life (RUL), which play a major role in predictive maintenance.

Condition monitoring signals can be captured from various sensors such as vibration, pressure, temperature, etc. Each of these signals contains both information about the health of the system and measurement noise. The Health Index is indicative of the SOH. It is typically constructed using inherent information within condition monitoring signals (Lei et al. (2018)). HI is a crucial indicator in PHM as it represents the SOH of the system and reveals the degradation process as accurately as possible. In addition, HI reduces the amount of information to the most essential. Moreover, it must be predictable in order to project in the future the health of the system and foresee the maintenance actions to be carried out over time (Lei (2016)).

As such, the process of HI construction is a major concern. There are several methods for constructing HI among which two approaches can be distinguished: Physics Health Index (PHI) and Virtual Health Index (VHI) (Lei et al. (2018). PHIs conserve the physical meaning of the collected sensor signals they are extracted from. PHIs are based on signal processing methods such as Root Mean Square (RMS) of vibration signals in time domain, Fast Fourier Transform in frequency domain, etc. On the other hand, VHIs are constructed by using multiple sensor readings fused into a single indicator (Hu et al. (2012)). Consequently, they no longer have any possibility of physical interpretation but consist of an implicit representation of the degradation trends. One way to obtain such VHIs is to use deep learning-based methods which are well adapted for the processing of multivariate non-stationary data and the extraction of inherent information.

Another important indicator is the Remaining Useful Life (RUL), which is defined as the remaining operating time before the End of Life (EOL) of the system (Si et al. (2011)). It can be written as follow:

$$RUL_k = t_{EOL} - t_k \quad (1)$$

where t_{EOL} is the EOL and t_k is the current time.

Most of the previous works based on deep learning techniques focus on RUL prediction by direct mapping from the values measured by the sensors. Furthermore, it turns out that most of the AI-based approaches leverage RUL-labeled data to perform such a mapping in a supervised manner. Such studies are based on various types of neural networks, including Convolutional Neural Networks (Babu et al. (2016), Li et al. (2018)), Long Short-Term Memory (LSTM) (Wu et al. (2018), Wang et al. (2018)), LSTM with attention (da Costa et al. (2019), Zhang et al. (2020)), and Temporal Convolutional Networks (Wenqiang et al. (2019), Zhou et al. (2020)).

On the other hand, autoencoders have proven to be very efficient in extracting VHI from raw sensors data. It was shown that for bearing vibration signals, the auto-extracted features were better than the handcrafted features (Hu et al. (2016)). In particular, the features obtained from autoencoders have monotonicity and clear trendability characteristic that are essential for RUL prediction. Gensler et al. (2016) also exploited the capabilities of autoencoders to perform feature extraction before predicting RUL, in a two-step architecture rather similar to the one presented in this paper.

However, in most of the real-life cases, the true RUL is not available. Indeed, obtaining the true RUL requires to conduct experiments (accelerated degradation tests) until the End Of Life (EOL) of the system is reached, which is often time consuming and costly. As most of the AI-based existing methods for prognostics rely heavily on the availability of labeled ground truth RUL data, the usefulness of such methods remain limited in the absence of such a ground truth RUL in real use cases.
The contribution of this paper lies in the RUL prediction based on long-range prediction of health indicator, namely the VHI, that avoids a direct dependence on labeled RUL target data. To that end, a vanilla autoencoder is employed for extraction of univariate VHIs from the multivariate sensor data in an unsupervised manner. This is followed by efficient long-term prediction of system health (quantified by VHI) into the future using a deep LSTM-based encoder-decoder structure. Finally, RUL predictions are generated in a recursive manner based on the predicted long term VHIs. As such, the overall prognostic methodology proposed here remains independent to labeled RUL data. The first stage corresponds to the capture of the spatial distribution of the VHI (feature correlation) while the second stage is the capture of the temporal evolution of the VHI. Such an unsupervised framework has been developed by Malhotra et al. (2016), but with a curve matching method for the RUL inference instead of a deep learning RUL prediction algorithm.

This study is part of a collaboration between the CRAN research centre and Dassault Aviation whose final objective is to develop a function for monitoring the condition of business jet aircraft systems by analyzing data recorded in flight.

The rest of the paper is organized as follows. Vanilla and LSTM-based Encoder-Decoders, along with the global architecture of the proposed framework are introduced in Section 2. Application results to the C-MAPSS dataset, including technical operations and choices made are presented in Section 3. Conclusion and possible future work are discussed in Section 4.

2. PROPOSED FRAMEWORK

In this section, the necessary background about the proposed framework is introduced.

2.1 Vanilla autoencoder

Encoder-decoder structure consists of two components. First, an encoder produces an encoding function

$$z = f_{\theta_e}(x) \quad (2)$$

that creates a compressed representation of the input x called "latent space" and denoted by z. Next, a decoder produces a decoding function

$$y = g_{\theta_d}(z). \quad (3)$$

The overall learning of the encoder-decoder can therefore be represented by the following nested function

$$y = g_{\theta_d}(f_{\theta_e}(x)). \quad (4)$$

Autoencoder structure is a special case of encoder-decoder architecture where the target space y is the same as the input space x. The goal of this structure is to reconstruct the input $x : y = x'$. The loss is therefore obtained via a function of x and x':

$$J_{AE}(\theta_e, \theta_d) = \sum L(x, x') = \sum L(x, g_{\theta_d}(f_{\theta_e}(x))) \quad (5)$$

where L is a loss function such as the mean squared error (Bengio et al. (2013)).

Such a structure can be a Fully Connected Network (FCN), with one or more hidden layers, trained to reproduce its input as output by forcing the computations to flow through a bottleneck representation, namely the latent space. As the latent space has a limited size, the network prioritizes learning the most meaningful features that allow an accurate reconstruction of the input. It is worth noting that autoencoders lead to unsupervised learning since they do not need labeled data to train on.

2.2 LSTM-based Encoder-Decoder

Recurrent neural networks (RNNs) are a special kind of neural networks that uses prior knowledge and not only current input to predict the output. Thus, RNNs are particularly useful for sequential data learning such as time series. An RNN can be thought of as multiple recurrent standard cells whose states are affected by both past states and current input. A particularly widespread version of the standard cell is the "Long-Short Term Memory" (LSTM), introduced by Hochreiter and Schmidhuber (1997).

RNN-based autoencoder is a special case of autoencoders where the encoder and the decoder parts are recurrent networks. It was first proposed by Cho et al. (2014) for sequence-to-sequence tasks, particularly for Neural Machine Translation (NMT). They are usually refered

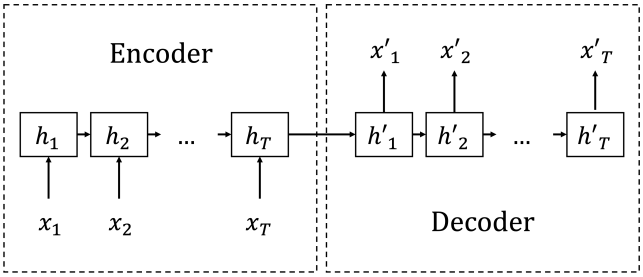

Fig. 1. Traditional structure of RNN-based Encoder-Decoder.

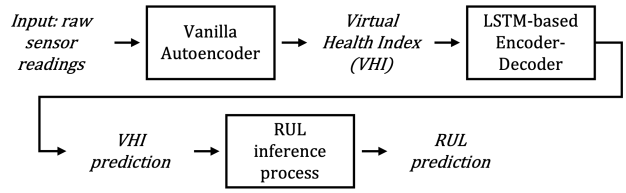

Fig. 2. Overall proposed RUL prediction procedure.

to as RNN-ED. The encoder RNN transforms the input sequence into a fixed-dimensional vector representation, while the decoder RNN maps it to the target sequence which, in our case, is the source sequence. Figure 1 shows an RNN-ED structure proposed by Malhotra et al. (2017).

Given a source time series $X = \{x_1, x_2, ..., x_T\}$, h_t is the hidden state of the encoder at time t. The encoder will capture relevant information as it encodes the time series, and as the last point T of the time series is reached, the hidden state h_T is the vector representation of the complete time series X (corresponding to the latent space in traditional autoencoder). The decoder has the same structure as the encoder and takes the final encoding hidden state h_T as initial decoding hidden state. It then reconstructs the original time series: $X' = \{x'_1, x'_2, ..., x'_T\}$. The structure is trained to minimize the reconstruction error between the source time series and the target time series, that is:

$$E = \sum_{i=1}^{N} \sum_{t=1}^{T} (x_t - x'_t) \tag{6}$$

for N time series of length T.
Instead of reconstructing the input time series, such a structure can also be used in order to predict future values of the source time series (See Du et al. (2020)), for instance predicting next HI values. Additionally, it is worth noting that the RNN encoder and decoder can take the form of several types of structures such as stacked-RNN and/or bidirectional RNN, as proposed by Yu et al. (2019).

2.3 Proposed RUL prediction procedure

The proposed procedure that leads to the RUL prediction is presented in Figure 2. It is essentially made of two steps.

First step. The first step is to extract a Health Index from a set of sensors readings. It is thus a dimensionality reduction problem, which is solved by using a vanilla autoencoder structure in a temporally-independent manner. At each timestep of the multidimensional input time series, the vector of the sensors readings is compressed into the latent space of the autoencoder and then reconstructed by the decoder part. The autoencoder is pruned in the testing phase, in order to conserve only the encoding funnel part, to reduce the dimensionality of the input data. This operation is repeated in a loop on the whole time series, in order to shift from a multivariate series to a univariate series. The structure is shown in Figure 3. As it will be introduced in Section 3, 7 sensors are selected as input. Thus, the encoder is made of 5 fully connected layers of size 7, followed by 3 layers of size 3, 2 and 1. The decoder is symmetrical to the encoder. A dropout of 0.2 is applied between each layer (See Srivastava et al. (2014)). The selected activation function is the hyperbolic tangent ($tanh$).

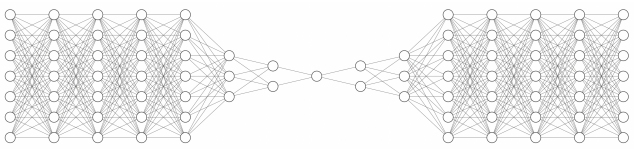

Fig. 3. Autoencoder structure for VHI extraction (step 1).

The newly created series is then normalized between 0 and 1 using a min-max scaler defined as follows:

$$HI' = \frac{HI - min(HI)}{max(HI) - min(HI)} \tag{7}$$

The univariate and normalized time series that is obtained from such a process is a Virtual Health Index, fusing the information from different input sensors.

Second step. Based on this new univariate VHI time series, an LSTM-based encoder-decoder is constructed to predict the future evolution of the VHI. Such a structure is ideal for this task as it leverages the recurrent nature of the data, and has already proved its efficiency on this kind of work cases (see Yu et al. (2019)). We choose a deep structure of stacked LSTM Encoder-Decoder (3 layers of 120 hidden units) in order to extract the deep temporal patterns of the VHI series.

The future values of VHI are continuing to be predicted until they reach the selected threshold, denoted as VHI_{EOL}. Once this threshold is reached, it implies that the EOL of the system has been reached. The RUL can then be deduced recursively by counting the number of iterations that have been necessary.

To that end, a sliding time window process is adopted. It consists of dragging a window of fixed size T along the VHI time series. The prediction of the future VHI values is made for a prediction window of length P. This prediction process is then repeated until the maximum value of the prediction window reaches the selected threshold VHI_{EOL}. After each prediction, the first S values of the prediction window are affected to a new input window, thus sliding step by step and using the previous predictions as new input, S being the stride value. The proposed RUL prediction process can be summarized by Algorithm 1.

Such a process gives the RUL value for one input window. The operation is then repeated for each input window available, until all the data is processed. This results in a complete RUL time series for the selected system. At the

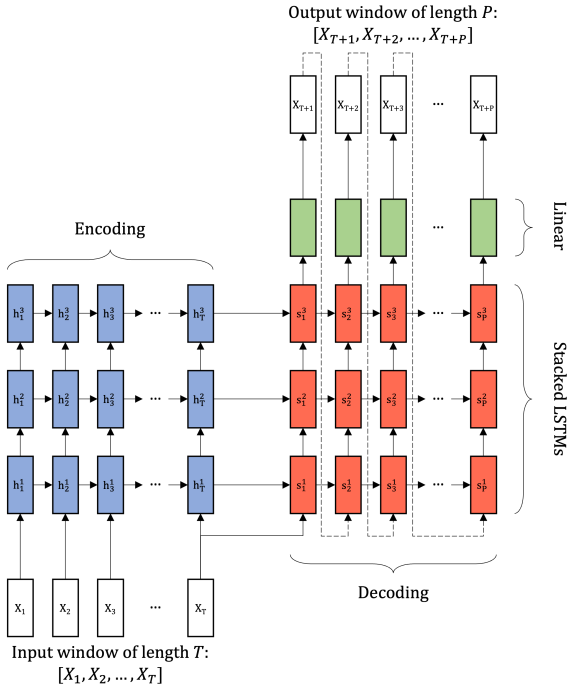

Fig. 4. Overview of the LSTM-based Encoder Decoder.

Hyperparameter	Value
Input window length (T)	50
Prediction window length (P)	50
Stride (S)	5
HI Threshold (VHI_{EOL})	0.86
Min HI Value for adaptive offset correction	0.7
Min offset value for adaptive offset correction	0.05

Table 1. Hyperparameters of the proposed method.

beginning of the process, the input window is temporally distant from the end of the life of the studied unit, which implies a higher variability in the prediction of the VHI than at the end.

The LSTM-based Encoder-Decoder that is used for the VHI prediction task is depicted in Figure 4.

The hyperparameters have been chosen empirically in the training phase, to optimize the RMSE of the predicted RUL. They can be found in Table 1. The input and prediction window lengths were chosen as long as possible in order to favour long-term predictions with clearer trends.

During the training phase, the LSTM-based Encoder-Decoder structure only sees VHI windows whose values are contained between 0 and 1. However, during the test phase, for a prediction window that must be generated at the very end of the time series it is possible that predicted values exceed 1. In such case, as the neural network has never seen such values, it tends to output prediction windows with correct dynamics but whose values necessarily remain below 1. To solve this problem, an adaptive offset correction mechanism is proposed. It consists of correcting the observed offset in an adaptive way, i.e. based on conditions that ensure correction is performed only when necessary. The conditions are as follows: first, the correction can be done only when the VHI is over 0.7, which guarantees that such operation is done only at the end of the VHI time series. Then, only negative offsets (i.e. when the prediction is under-estimated) can be corrected.

Algorithm 1: RUL prediction process for one input window

Input: A VHI window of length T denoted by X.
Output: A RUL value (scalar).
$i = 0$;
$Y \leftarrow$ VHI predicted window of length P based on input X;
while $max(Y) < VHI_{EOL}$ **do**
  $i \leftarrow i + 1$;
  $X \leftarrow X[S:] + Y[0:S]$; `/* S is the stride */`
  $Y \leftarrow$ new VHI predicted window based on new input X;
end
$index \leftarrow argmax(Y < VHI_{EOL})$;
$RUL \leftarrow S \times i + index$;

3. EXPERIMENTAL STUDY ON C-MAPSS

3.1 C-MAPSS Dataset

The C-MAPSS dataset contains turbofan engines degradation data which have been obtained by using a simulator developed by NASA, in closed-loop configuration (See Saxena et al. (2008)).

The C-MAPSS dataset actually consists of four distinct subsets referred to as FD001, FD002, FD003 and FD004, each containing a training set composed of complete degradation sequences of a number of turbines, and a test set. For each of the four subsets, both turbine populations (training and test) are assumed to belong to the same distribution. The fault modes in the subsets vary between one in FD001 and FD002 and two in FD003 and FD004, while the operating conditions vary between one in FD001 and FD003, to six, based on different combinations of altitude (0 to 42000 feet), throttle resolver angle (20 to 100) and Mach (0 to 0.84) in FD002 and FD004. In this work, only the first dataset (FD001) is used for the demonstration of the concept.

3.2 Data preparation

From the 21 available sensors, only the 7 most meaningful sensors are kept, based on a detailed observation of the shape of the different time series acquired by the sensors, as it was done by Wang et al. (2008). This rich subset is composed of sensors 2, 3, 4, 7, 11, 12, 15. Then, in order to ensure an equal contribution from all sensor readings, standard scaling is applied, as follows:

$$Norm(x^s) = \frac{x^s - \mu^s}{\sigma^s} \tag{8}$$

where s is the selected sensor, μ^s and σ^s are the mean and the standard deviation for the selected sensor. An example of the seven input sensors for turbine n°29 from FD001 training set can be seen in Figure 5. In this figure, the time series collected by the 7 sensors were normalized according to Equation 8 and plotted together on a common graph.

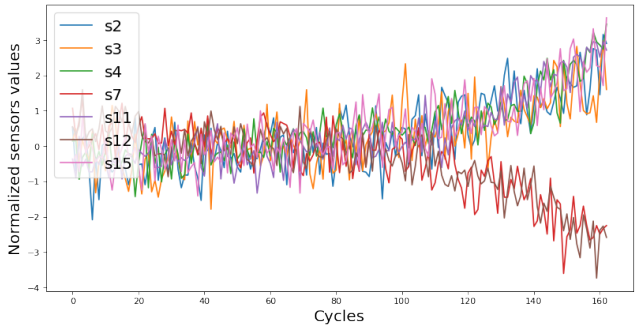

Fig. 5. Normalized sensors readings of one turbine from FD001 training set (turbine n°29).

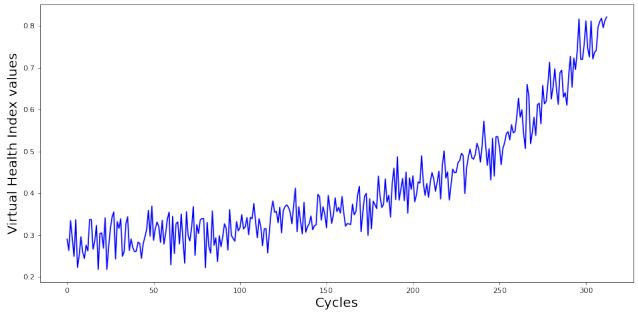

Fig. 6. Virtual Health Index of one turbine from the test set (turbine n°69).

3.3 Model evaluation

To evaluate the quality of the RUL prediction, the Root Mean Square Error (RMSE) has been selected as performance metric. RMSE is the most used indicator in the literature to compare RUL prediction methods on C-MAPSS and it offers a good representativeness of our model performances. It could be complemented with additional indicators such as the score function (Saxena et al. (2008)) but for reasons of space, only the RMSE will be used here. RMSE is computed as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}, \tag{9}$$

for n predictions, where d_i is the difference between the predicted and the actual RUL:

$$d_i = \hat{RUL}_i - RUL_i. \tag{10}$$

3.4 Experiment results

First step. The extracted VHI of one turbine from the FD001 test set can be seen in Figure 6. It presents, through a visual examination, a clear increasing trend with good global monotonicity feature. It is worth noting that all the turbines from the test set are normalized together. As such, all the VHI values are contained in the interval $[0, 1]$ but might not reach its bounds. For instance, one can see in Figure 6 that the maximum value is 0.82, not 1. This affects the selection of the value of the threshold in the RUL prediction procedure.

Second step. A complete RUL trajectory for one turbine of the FD001 test set can bee seen in Figure 7. The

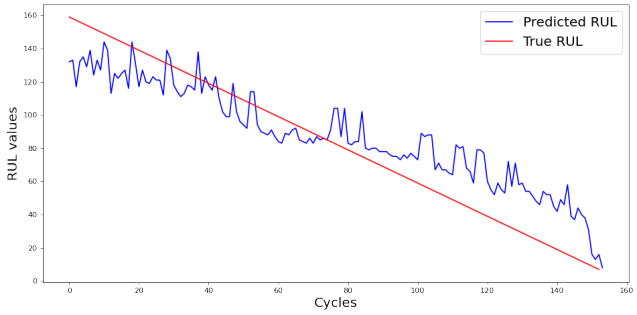

Fig. 7. Example of predicted RUL trajectory (turbine n°34 from FD001 test set).

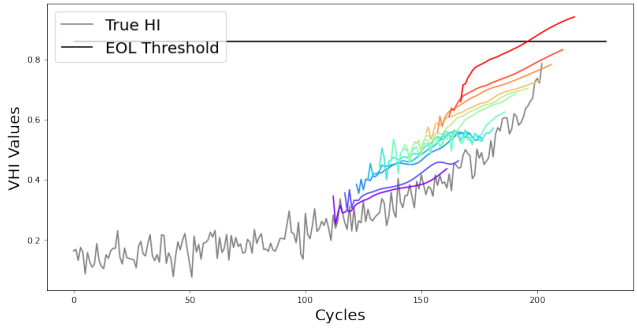

Fig. 8. Example of VHI predictions, for turbine n°34 (from FD001 test set), at timestep 62. The rainbow curves correspond to the successive prediction windows of the VHI (length $P = 50$).

prediction of the VHI for the same turbine, at timestep 62, is available in Figure 8. This gives an example of RUL calculation based on the VHI prediction. As shown in Figure 8, 12 predictions of windows of size $P = 50$ are needed to reach the threshold $VHI_{EOL} = 0.86$, with a stride of $S = 5$. The threshold is reached, in the last prediction window, at the 29th (denoted as *index*) value of the time window. This results in a RUL of:

$$RUL = 12 \times S + index = 12 \times 5 + 29 = 89. \tag{11}$$

As a minimum input window of length $T = 50$ is required, predictions could not be realized for a few turbines of the test set whose length does not exceed this value. The RMSE is calculated for each of the remaining turbines, thus giving a mean RMSE for the test set of 44.7 with a standard deviation of 27.4.

4. CONCLUSION AND FUTURE WORK

The methodology proposed in this paper accomplishes prediction of RUL in an unsupervised manner avoiding a direct dependence on the availability of labeled RUL data for training. It is particularly relevant as in most of the real industrial use-cases, data are incomplete and/or unlabeled. In this work, as a first proof of concept, vanilla autoencoders have been employed for an unsupervised extraction of health index (VHI) from raw sensor readings followed by the long range prediction of the latter using a LSTM-based Encoder-Decoder. This proposed two-steps prediction process has been applied to C-MAPSS data (FD001).

The results obtained, although not up to par with those coming of supervised techniques, are very encouraging as the direct mapping between raw sensor and RUL is avoided. Such a difference in performance between the existing supervised approaches and the proposed unsupervised (RUL independent) approach arises mainly due to non-usage of RUL data as target labels. Future perspective involves improvement of VHI extraction quality using more efficient autoencoder structures as well as efficient long-range VHI predictions under variable operating conditions especially on FD002 and FD004 subsets of C-MAPSS.

REFERENCES

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

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

da Costa, P.R.d.O., Akcay, A., Zhang, Y., and Kaymak, U. (2019). Attention and long short-term memory network for remaining useful lifetime predictions of turbofan engine degradation. *International Journal of Prognostics and Health Management*, 10, 034.

Du, S., Li, T., Yang, Y., and Horng, S.J. (2020). Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing*, 388, 269–279.

Gensler, A., Henze, J., Sick, B., and Raabe, N. (2016). Deep learning for solar power forecasting—an approach using autoencoder and LSTM neural networks. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)*, 002858–002865. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Hu, C., Youn, B.D., Wang, P., and Yoon, J.T. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering & System Safety*, 103, 120–135.

Hu, Y., Palmé, T., and Fink, O. (2016). Deep health indicator extraction: A method based on auto-encoders and extreme learning machines. In *PHM 2016, Denver, USA, 3-6 October 2016*, 446–452.

Jardine, A.K., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7), 1483–1510.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical systems and signal processing*, 42(1-2), 314–334.

Lei, Y. (2016). *Intelligent fault diagnosis and remaining useful life prediction of rotating machinery*. Butterworth-Heinemann.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., and Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical systems and signal processing*, 104, 799–834.

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

Malhotra, P., TV, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G. (2016). Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. *arXiv preprint arXiv:1608.06154*.

Malhotra, P., TV, V., Vig, L., Agarwal, P., and Shroff, G. (2017). Timenet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*.

Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management*, 1–9. IEEE, Denver, Colorado, USA.

Si, X.S., Wang, W., Hu, C.H., and Zhou, D.H. (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1), 1–14.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.

Wang, J., Wen, G., Yang, S., and Liu, Y. (2018). Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network. In *2018 Prognostics and System Health Management Conference*, 1037–1042. IEEE, Chongqing, China.

Wang, T., Yu, J., Siegel, D., and Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *International Conference on Prognostics and Health Management*, 1–6. IEEE, Denver, Colorado, USA.

Wenqiang, J., Jian, C., and Yi, C. (2019). Remaining useful life prediction for mechanical equipment based on temporal convolutional network. In *2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 1192–1199. IEEE, Changsha, China.

Wu, Y., Yuan, M., Dong, S., Lin, L., and Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, 275, 167–179.

Yu, W., Kim, I.Y., and Mechefske, C. (2019). Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems and Signal Processing*, 129, 764–780.

Zhang, H., Zhang, Q., Shao, S., Niu, T., and Yang, X. (2020). Attention-based LSTM network for rotatory machine remaining useful life prediction. *IEEE Access*, 8, 132188–132199.

Zhou, D., Li, Z., Zhu, J., Zhang, H., and Hou, L. (2020). State of health monitoring and remaining useful life prediction of lithium-ion batteries based on temporal convolutional network. *IEEE Access*, 8, 53307–53320.