Prognostics in Aeronautics with Deep Recurrent Neural Networks

Marcia Baptista[1], Helmut Prendinger[2], and Elsa Henriques[3]

[1,3] *University of Lisbon, Instituto Superior Tecnico, Lisbon, Portugal*
marcia.baptista@ist.utl.pt
elsa.henriques@flad.pt

[2] *National Institute of Informatics, Tokyo, Japan*
helmut@nii.ac.jp

ABSTRACT

Recurrent neural networks (RNNs) such as LSTM and GRU are not new to the field of prognostics. However, the performance of neural networks strongly depends on their architectural structure. In this work, we investigate a hybrid network architecture that is a combination of recurrent and feed-forward (conditional) layers. Two networks, one recurrent and another feed-forward, are chained together, with inference and weight gradients being learned using the standard back-propagation learning procedure. To better tune the network, instead of using raw sensor data, we do some preprocessing on the data, using mostly simple but effective statistics (researched in previous work). This helps the feature extraction phase and eases the problem of finding a suitable network configuration among the immense set of possible ones. This is not the first proposal of a hybrid network in prognostics but our work is novel in the sense that it performs a more comprehensive comparison of this type of architecture for different RNN layers and number of layers. Also, we compare our work with other classical machine learning methods. Evaluation is performed on two real-world case studies from the aero-engine industry: one involving a critical valve subsystem of the jet engine and another the whole reliability of the jet engine. Our goal here is to compare two cases contrasting micro (valve) and macro (whole engine) prognostics. Our results indicate that the performance of the LSTM and GRU deep networks are significantly better than that of other models.

1. INTRODUCTION

Traditionally, the process of developing prognostics models involves a high degree of expert knowledge and technical skill (Medjaher, Camci, & Zerhouni, 2012). In contrast, data-driven approaches to prognostics involve constructing computational models that result from an analysis of sensor data without explicit knowledge of the underlying physical behavior (Jardine, Lin, & Banjevic, 2006).

The development of models that can capture both spatial and temporal patterns is of great importance to the community of prognostics since to predict the remaining useful life of the equipment it is necessary to deal with multiple sensor signals. One possible way to handle this data is by the use of recurrent neural networks (RNN). An RNN is a kind of artificial neural network that is based on the idea of the neural memory. Here, neural memory is the ability to remember input from previous time steps. By having this ability, the network can process sequential data.

Even though RNNs are not new to prognostics and have been around for some time, only recently, with the emergence of deep learning, have recurrent networks been subject to closer examination. Deep learning brought the possibility to develop larger networks (i.e. with more layers) with consequently more representative power. As a result, larger RNNs have started to be proposed in prognostics. Examples of RNNs here are: the standard network (Elman, 1990; Jordan, 1997), the Long-Short Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997) and the more recent Gated Recurrent Unit (GRU) network (Cho et al., 2014).

Defining network architecture is an important decision that can greatly influence performance. In this work, and to take full advantage of the capabilities of the RNN, we propose a hybrid neural network that combines RNN layers with multilayer perceptron (MLP) layers. The RNN is responsible for high-level feature extraction while the MLP performs the remaining useful life (RUL) prediction at each time step. To accelerate training, raw features are preprocessed using simple and general statistical functions before they were fed into the network.

Our work brings some innovative contributions in the sense that is an extensive comparison of networks. Since we compare three RNN type of layers (standard, LSTM, and GRU) and we also vary the number of layers (1 layer or 6 stacked layers) we build and compare 6 kinds of models. With this, we aim to (i) investigate which is the best RNN layer for our architecture and to (ii) confirm that deeper (with 6 stacked RNN layers) networks can attain better prediction results than shallow ones (with a single layer).

To make our work more comprehensive, we also compare the proposed models against the classical methods (X. Wu et al., 2008) of random forests (RF) (Breiman, 2001), support vector machines (SVM) (Cortes & Vapnik, 1995; Schölkopf & Burges, 1998), neural networks (NN) (Reed & Marks, 1989), k-nearest neighbors (KNN) (Aha, Kibler, & Albert, 1991) and generalized linear models (GLM) (Nelder & Wedderburn, 1972). The goal here is to show the superiority of the proposed deep architecture over traditional machine learning methods. We should note that this comparison is done in two large-scale real-world datasets.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes in detail the proposed architecture. In Section 4 we provide a detailed treatment of the two real-world datasets, which are used to establish the validity of the proposed architecture. Finally, conclusions and future work are in Section 5.

2. RELATED WORK

The motivation for this paper is to examine different RNN structures for prognostics. This section starts with a general review of deep learning models in prognostics. Then, we focus on papers that propose RNN models and discuss their relation to our research.

The prognostics of the future health of equipment consists in knowing/gnosis that a future health phenomenon (e.g. a failure event) will occur in a prospective time, or that the health state at issue will prevail. This perception amounts to a prediction of the equipment end of life (EoL) or remaining useful life (RUL). Traditionally, prognostics have relied on model-based approaches to estimate the damage and mitigate system risk (Oppenheimer & Loparo, 2002; Adams, 2002; Chelidze & Cusumano, 2004; Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008; Saha & Goebel, 2009; M. Daigle, Saha, & Goebel, 2012). These approaches use domain knowledge of the system, including its components and how they fail, to describe the underlying physical phenomena in a physics-of-failure (PoF) model (M. J. Daigle & Goebel, 2011). These prognostics tools are however expensive and time-consuming which lends strength and credibility to the investigation of new approaches and tools in the field. Moreover, when complex engineering systems are involved with intricate and non-linear interactions between their components and with the domain where they operate, the development of such models may even exceed knowledge and technologies currently available.

As an alternative to model-based tools, data-driven approaches (Atherton, 1999; Gupta & Ray, 2007; Goebel, Saha, Saxena, Celaya, & Christophersen, 2008) are applied to prognostics when sufficient data exist to establish the damage space (Goebel, Saha, & Saxena, 2008). Deep learning is a novel kind of data-driven methods that attempts to somewhat relief the dependence of classical machine learning on feature extraction methods. In deep learning, features can be learned from data using a general-purpose learning procedure (LeCun, Bengio, & Hinton, 2015). This characteristic is especially important to prognostics, where performance is severely dependent on the quality of features and most feature extraction is still performed by machine learning experts in conjunction with domain experts (Yan & Yu, 2015; Brownlee, 2014). Some methods have been proposed in the literature for deep learning prognostics. Mainly, the techniques used include Auto-encoder (AE) and its variants, Restricted Boltzmann Machines and its variants including Deep Belief Network (DBN) and Deep Boltzmann Machines (DBM), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) (Zhao et al., 2019).

Recurrent Neural Networks is a learning model that is suitable for dealing with cross-sectional time series (i.e., independent sets of single time-series). These networks can represent time series as they include a recurrent connection in each of their computational units. The output (activation) of a unit is feedback to itself with a weight and a unit time delay at each time step, which provides the model with a memory of its past activation and enables it to learn the temporal dynamics of sequential data.

Unlike feedforward networks, in which computations are performed within a one-time frame, RNNs map inputs to outputs over multiple time steps. Observing that the RNN undergoes multiple transformations not only feedforward (from input to output within a time step) but also recurrently (across multiple time steps), two definitions of depth can be applied: the traditional, feedforward depth and the recurrent depth (S. Zhang et al., 2016). From these two distinct definitions follows the notion that recurrent neural networks are the “deepest” of all neural networks (Schmidhuber, 2015).

Several works have adopted RNNs in prognostics. The work in (Atherton, 1999) employs a shallow variant of the standard RNN (Elman, 1990; Jordan, 1997). The network is used to predict machine deterioration using vibration data. In (Heimes, 2008), a hybrid RNN model is proposed for RUL prediction of turbofan engines, with the architecture being tuned using differential evolution and weights being set using an Extended Kalman Filter-based algorithm. The network architecture is simple, utilizing only 24 inputs, with three layers of feed-forward connections, and three layers of recurrent connections (standard RNN layer). The authors, however, do not disclose much information about the architecture as it is proprietary.

Due to their specific topology, the LSTM is an RNN better able to learn short term and long term temporal dependencies that has yielded positive results in prognostics. For example, In (Z. Zhang, Lu, Zhou, & Liao, 2018) authors show that the LSTM can outperform auto-regressive methods. The model in (Zhao, Yan, Wang, & Mao, 2017) combines convolution neural work with bi-directional LSTM for RUL estimation. In (Guo, Li, Jia, Lei, & Lin, 2017), an LSTM-based model is used to predict the residual life of wind turbine generator bearings. In (Zhao, Wang, Yan, & Mao, 2016), shallow and deep LSTMs are developed to predict the wear condition of a cutting tool. Experimental results suggest that both shallow and deep LSTMs can outperform several state-of-art baseline methods and that deep LSTMs are the best performing methods. As this work is mainly focused on the LSTM architecture it lacks a comparison of other architectures such as the gated recurrent unit (GRU).

Gated recurrent units (GRUs) (Cho et al., 2014) are an RNN model with similar performance to that of LSTM (Chung, Gulcehre, Cho, & Bengio, 2014) but with fewer parameters and therefore a faster learning process. GRUs are used with success in (Song, Li, Peng, & Liu, 2018). In (Yuan, Wu, & Lin, 2016), the GRU is compared with the standard RNN and the LSTM. Their experiment results show the standard LSTM outperforms the others. From their paper it is however not clear how many RNN layers were used and the specific details of their architecture.

3. NETWORK ARCHITECTURE

In this section, we introduce the recurrent network architecture for remaining useful life (RUL) estimation. The main criteria justifying the choice of this architecture is to have a solution able to deal with the temporal aspect of sensor data. Fig. 1 illustrates the general architecture and its main building blocks. Formally, the goal is to provide for each time step t, with t running from 1 to T, a predicted RUL $\hat{y}_t$ given a set of sensor inputs $\{x_1, ..., x_n\}$. The main assumption here is that several layers of recurrent algorithms, followed by fully-connected layers can give better flexibility and performance to deal with RUL forecasting.

To train the neural networks we use the conventional back-propagation learning procedure. This algorithm needs an adaptive step size method. After some empirical testing, with RMSProp, Adam and other optimizers, we found the Adam optimizer to be suited to our problem as the adaptive step size method.

Table 1. Preprocessing functions.

Input	Description	Equation
p_1	Average amplitude	$\frac{1}{k}\sum_{i=1}^{k} s(i)$
p_2	Standard deviation	$\left(\frac{\sum_{i=1}^{k}(s(i)-p_1)^2}{k-1}\right)^{\frac{1}{2}}$
p_3	Root mean square amplitude	$\left(\frac{1}{k}\sum_{i=1}^{k} s(i)^2\right)^{\frac{1}{2}}$
p_4	Squared mean root absolute amplitude	$\left(\frac{1}{k}\sum_{i=1}^{k} \lvert s(i)\rvert^{\frac{1}{2}}\right)^2$
p_5	Kurtosis coefficient	$\frac{\sum_{i=1}^{k}\left(s(i)-p_1\right)^4}{(k-1)p_2^4}$
p_6	Skewness coefficient	$\frac{\sum_{i=1}^{k}\left(s(i)-p_1\right)^3}{(k-1)p_2^3}$
p_7	Peak value	$\max \lvert s(i)\rvert$
p_8	Peak factor	$\frac{p_7}{p_3}$
p_9	Margin factor	$\frac{p_7}{p_4}$
p_{10}	Waveform factor	$\frac{p_3}{\frac{1}{k}\sum_{i=1}^{k}\lvert s(i)\rvert}$
p_{11}	Impulse factor	$\frac{p_7}{\frac{1}{k}\sum_{i=1}^{k}\lvert s(i)\rvert}$

3.1. Input layer

As shown in Fig. 1, the proposed architecture comprises four different types of layers, i.e. input, recurrent, fully-connected and the output layer. The input layer receives at each iteration a vector $\boldsymbol{x}$ of two dimensions: the time ($t = \{1, \ldots, T\}$) and the sensor ($n = \{1, \ldots, N\}$) dimension. The time dimension specifies the number T of time steps to be passed to the model while the sensor dimension specifies the number N of sensor inputs.

The size of the sensor dimension (N parameter) is fixed. However, the size of the time dimension (T parameter) varies. We do this to prevent the mixing of data from different asset samples. As shown in Fig. 1, the model considers the existence of several samples, where each sample contains the data corresponding to the lifetime of an asset. Data from two samples should not be mixed to avoid situations where the network uses past information about a given asset to predict the RUL of another asset. If the data were to be continuously processed with a fixed-size time window this would most likely happen. Given the variability of the lifetime of each asset, it is not possible to select a time window size that guarantees that distinct assets are not mixed within the same network input. If this situation was not prevented, performance could degrade unrealistically. What we do is we divide the data of one asset into T chunks and we have a final chunk with the remaining temporal size. There is no need for padding as the network can deal with variable size sequences.

Please note that the number of sensor inputs does not equal the number of raw sensory signals in the two cases. This follows from the volume of our data: the raw data collected from the sensors are time series with a high sampling frequency. If

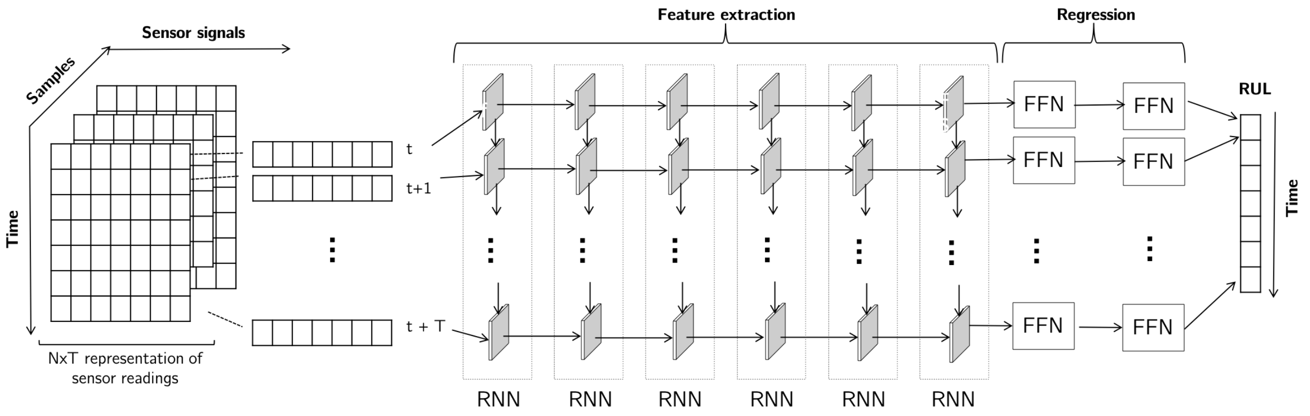


Figure 1. Architecture. The computation of the predicted remaining useful life (RUL) for a sample set of sensor signals is a dynamic process. First, the sensor readings of each sample are fed into the recurrent neural network (RNN) layers time at a time. Then, the hidden states of the recurrent layers are used to compute high-level features. Two layers of fully connected neurons (FFN) are used to compute the remaining useful life (RUL) of each time step based on the extracted features.

the data were to be processed at each time point, the computation would be too expensive. Accordingly, the time-series are processed by applying the functions shown in Tab. 1 (Q. Wu, Yang, & Zhou, 2012) on segments of data to produce new time domain inputs. The application of the functions in Tab. 1 is done on a flight by flight basis, similarly to what experts in the field do. Four functions (p_1, p_3, p_4, p_7) are used to capture the amplitude and energy of each signal while the remaining ones aim to reflect the distribution of each signal over the time domain. Also note that this preprocessing of the data is especially important for the performance of the model: as the recurrent layers are deep both in time and in space, they are computationally expensive. Preprocessing the raw data using simple and general functions allows us to have more recurrent layers and hence a faster feature extraction by the network.

Please note that the used preprocessing functions are general enough to be applied to most prognostics cases. The goal here is not the extensive extraction of features by-hand, but to propose a set of useful methods that can speed up the training process of the deep learning models. The extraction of the high-level features continues to be a responsibility of the networks. What we advocate here is the use of generic methods to speed the network's work.

3.2. Recurrent Layers

The present approach is one of the deepest architectures proposed for RUL estimation, as it can involve up to six layers of recurrent layers. Despite the benefits of such a deep architecture, there is also the risk of model overfitting. To prevent this situation and allow the model to generalize well to unseen data we employ a set of strategies. First, we use a dropout layer (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) after each of the recurrent layers. The key idea here is to randomly drop units (along with their connections) from the network during training. Second, we employ L2-norm regularization (Ng, 2004). L2 works by adding a penalty on the norm of the weights to the loss. Third, early stopping is used (Cataltepe, Abu-Mostafa, & Magdon-Ismail, 1999). The monitor object of early stopping is the performance on the training set.

Different kind of recurrent layers can be selected for the same model. This is another distinguishing trait of our architecture: a model can be constituted by the standard RNN layer, the LSTM layer, and the GRU layer. LSTMs extend RNNs with the use of memory cells instead of recurrent units. LSTM memory cells make use of the mechanism of gating: each cell is updated according to the activation of different gates that control which operation is performed on the cell memory: write (input gate), read (output gate) or reset (forget gate). GRU extends the RNN by an update gate and a reset gate. The update gate determines how much the inputs can change the new state while the reset gate determines to what extent the old state needs to be erased. The GRU memory cell only has two gates while the LSTM cell has three gates. This makes the design of the GRU more simple. Here, the standard RNN stood out for its simplicity, LSTM was selected for its popularity and GRU for its training performance.

3.3. Prediction and output layers

After feature extraction, the extracted features are weighted and combined in two fully-connected layers as shown in Fig. 1. This represents the prediction part of the model. Finally, there exists one output neuron for each time step in the input layer.

The fact that we are chaining a recurrent neural network (RNN) with a fully connected network (FFN) brings several advantages. The main advantage of RNN layers is that they are capable of extracting useful patterns from temporal data. In turn, the main advantage of FFNs is that they are "input agnostic". No particular assumptions need to be made about the input as in the case of RNNs (sequences) or CNNs (images). In a certain way, fully connected architectures act as "universal approximators" capable of learning any function. This generality is important for prognostics where each case study has its peculiarities.

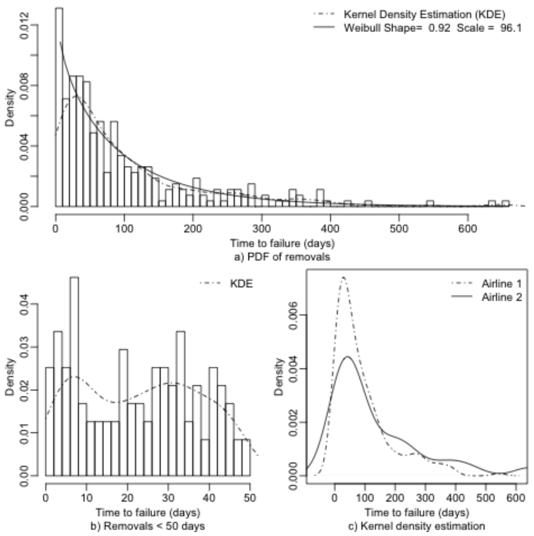
4. METHODOLOGY AND RESULTS

In this section, we evaluate the performance of the proposed approach. First, the datasets of two case studies are described. Then, details about the experimental setup are provided. Finally, results are presented and discussed.

4.1. Datasets

This study uses two real-world datasets from the aeronautics sector: one related to the whole reliability of a modern aero-engine (DS-1) and another related to a critical component of the engine (DS-2). Regarding the first dataset, DS-1, the data describe the evolution of the performance of a set of commercial jet engines between approximately ten years in different intervals of time for each engine. Formally, the data consists of a cross-sectional time series in the sense that for each engine, there is a multi-variate series that represents the temporal progression of the engine sensor signals. These signals are measured at three different flight phases: 1 measurement is taken at take-off, another at climb and 3 other at cruise. Overall, we analyze around 3GB of raw data. In addition to performance signals, there is also information about the engine overhauls. An engine overhaul can be defined as a comprehensive inspection that involves removing and disassembling the engine, testing all its sub-systems, cleaning and replacing parts as needed and then reassembling the engine (Seemann, Langhans, Schilling, & Gollnick, 2010). The dataset includes fixed-interval and condition-based overhauls.

The second dataset DS-2 describes the reliability of a set of engine bleed valves. These valves are critical systems (de Pádua Moreira & Nascimento, 2012) as, if not working as expected, they can make the compressor "stall" (meaning it abruptly ceases operating and stops turning, at least briefly). If one of these valves happens to fail, large amounts of power can be lost, even enough to result in an airplane on the ground (AOG) scenario. Mostly due, not to the valve itself, but to the complexity of the system where the valve operates, it is not always easy to recognize fault existences (M. Baptista, de Medeiros, et al., 2017). To address the problem of this kind of RUL prediction, we study real data of several valve unscheduled removals recorded between 2010 and 2015 from commercial aircraft of three airlines. Here, by *removal* we mean a maintenance and repair action where the equipment is removed from the aircraft and restored to its original condition or replaced by a new/repaired unit. Fig. 2 shows the probability density function (PDF) of time between two consecutive unscheduled removals. As illustrated, the maintenance events are highly dispersed, with the time between removals ranging from 0 to 543 days. In addition to the removal events of the valves, the data set comprises 100 GB of data collected from aircraft sensors as time series with a sampling frequency of 1 Hz as well as information about environmental conditions during flight.



a) PDF of removals

b) Removals < 50 days

c) Kernel density estimation

Figure 2. Probability Density Function (PDF) of removal times (DS-1).

4.2. Research Question

The goal of our experiments is to show the effectiveness of the proposed RNN approach for RUL prediction. Concretely, we empirically evaluate the performance of a deep standard recurrent neural network (DSRNN), a deep long-short term memory (DLSTM) and a deep gated recurrent unit (DGRU) models with other data-driven algorithms that have been applied to the datasets in earlier works (M. Baptista, de Medeiros, et al., 2017; M. L. Baptista et al., 2017; M. Baptista, Sankararaman, et al., 2017) namely, generalized linear model (GLM), neural network (NN), random forests (RF), k-nearest neighbors (KNN) and support vector regression (SVR). To study the influence of deep learning we also empirically evaluate the performance of the proposed deep models (DSRNN, DLSTM, and DGRU) against their shallow versions (SSRNN, SLSTM, and SGRU).

4.3. Software and Hardware Specifications

All experiments were run on a personal computer with Intel Core i7-4500U (1.80GHz) CPU, 4GB memory and Ubuntu 14.04. All code is written in Python 3.6 with scientific computing library "Theano" and deep learning library "Keras".

4.4. Performance Metrics

A considerable number of performance metrics have been used in prognostics (Saxena et al., 2008). In this study, we feature a subset of these. The used accuracy and robustness metrics include, for example, the root mean squared error (RMSE), and the median absolute deviation (MAD) respectively. An extensive list of the considered metrics and calculation methods is provided in Table 2. We encourage the reader to get more details of these metrics from (Tang, Orchard, Goebel, & Vachtsevanos, 2011).

4.5. Evaluation Methods

The 10-fold cross-validation scheme is used to verify the learning ability of the algorithm to generalize to unseen data, i.e. the testing dataset. This scheme is used to evaluate the proposed models and the baseline classical machine learning models.

4.6. Results

In this section we present the results of testing the three proposed RNN algorithms (DRNN, DLSTM, and DGRU) and their shallow versions (SRNN, SLSTM and SGRU) as well as five traditional data-driven methods (NN, GLM, RF, KNN and SVR) on DS-1 (whole engine) and DS-2 (bleed valve case). Table 3 and Table 4 illustrate the comparison results across the two datasets in terms of bias, accuracy and robustness. It can be observed that the deep methods consistently achieve the best values in accuracy and robustness. For instance, the deep LSTM (DLSTM) predicts the RUL of the engine with an average absolute error (MAD) of 43 cycles (see Table 3) and the deep standard network predicts the RUL of the valve with an average error (MAD) of around 5 cycles (see Table 4). The performance of the machine learning methods are not so impressive, with the best methods having larger absolute errors. For example, the generalized linear model, has a MAD of 70 cycles in the engine case (DS-1) (63% more than the DLSTM), and the support vector regression (SVR) has a MAD of 9 cycles in the valve case (DS-2) (80% more than the DSRNN). As expected, in scenarios where machine learning already provides good estimates (DS-1), the deep algorithm leads to less significant performance differences.

Not surprisingly, deep networks were able to outperform their shallow versions for both datasets. This can be seen by comparing the performance of shallow neural learning against the performance of deep learning in Table 3 and Table 4. The difference of performance is however not as extreme as in the previous comparison. In DS-1, the best performing shallow model, the shallow RNN (SRNN), has a MAD of 50 cycles (16% more than the DLSTM). In DS-2, the best performing shallow model, the shallow GRU (SGRU), has a MAD of 8 cycles (60% more than the DSRNN). These results indicate that shallow RNNs can outperform machine learning despite their relative simplicity. We hypothesize that this is due to the fact that any RNN layer can capture temporality in a more effective way than classical machine learning methods.

Among the three proposed methods, we consider that DGRU achieved the best overall performance on dataset DS-1 (engine) and DLSTM on the dataset DS-2 (valve). Since there was no clear winner, this result suggests that performance depends significantly of the industrial application. The most adequate learning method appears to depend on the health management system and its operating environment. From our experiments, however, It seems that if well configured, the deep layers of LSTM and GRU can yield better overall results than standard RNN. This is not surprising as these methods are more sophisticated.

Interestingly, the relative accuracy (RA) of the best performing models was somewhat similar for both scenarios (40% for DS-1 and 31% for DS-2). The models had similar predictive ability even though the first model was more powerful. This is an intriguing result as it was expected that macro prognostics (DS-1:engine case)would be more difficult than micro prognostics (DS-2:valve case). To explain these results, we hypothesize that the difficulty of the prognostics task depends mostly on the input data and the intricacies of its failure patterns and not the macro or micro aspect of the task.

The performance over time of the proposed deep models is shown in Figure 3 and 4 for datasets DS-1 and DS-2. In the x-axis of plots a) to c) the fraction of time to end-of-life is shown (t_λ) while the y-axis represents different evaluation metrics. From an inspection of the top plots, it can be seen that deep recurrent neural networks excel at capturing failure patterns close to the end-of-life while the other more traditional methods exhibit more difficulties in achieving the same level of accuracy near the equipment end of life. Regarding robustness, the bottom plots ascertain the slightly more robust nature of the predictions of the proposed models.

Overall, these results lend recognition to the use of recurrent neural networks and the further exploration of deep learning methods in prognostics.

5. CONCLUSIONS

It is widely recognized by the community that deep learning models can outperform other methods. However, it is still not so clear the extent to which recurrent neural network (RNN) algorithms can promote better prediction models. The RNN models are powerful in the sense that they can explicitly capture the temporality of the data and hold a memory between calculations, a unique property that most artificial neural networks do not hold. This capability makes them worthy of further exploration, especially in the field of prognostics. We aimed here to provide two concrete and complex real-world industrial cases on the topic.

Table 2. Performance metrics.

Metric	Abbr	Formula
Mean Error	ME	$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$
Median Error	MdE	$\text{median}(\{\hat{y}_i - y_i\}_{i=1}^{n})$
Root Mean Squared Error	RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$
Relative Accuracy	RA	$1 - \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{\hat{y}_i - y_i}{y_i}\right\rvert$
Mean Absolute Error	MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$
Median Absolute Error	MdAE	$\text{median}(\{\lvert\hat{y}_i - y_i\rvert\}_{i=1}^{n})$
Median Absolute Deviation	MAD	$\text{median}\left(\{\lvert y_i - \text{median}(\{y_i\}_{i=1}^{n})\rvert\}_{i=1}^{n}\right)$
Sample Standard Deviation	SSD	$\sqrt{\frac{\sum_{i=1}^{n}((\hat{y}_i - y_i) - \text{ME})^2}{n-1}}$

Note: the term n stands for number of $\{y_i\}_{i=1}^{n}$ observations in the testing set. For each observation y_i, the model outputs the $\hat{y}_i$ prediction. Here, variable y_i means the time to next condition event at time index i. All measures are given in days except the MAPE (%).

Table 3. Results for DS-1 (engine) (ordered by RMSE/ best in bold).

	Bias		Accuracy				Robustness	
	ME	MdE	RMSE	RA	MAE	MdAE	MAD	SSD
			Classical Machine Learning					
RF	149.32	164.60	234.19	22.89	208.87	192.93	81.28	120.62
GLM	153.18	170.40	223.59	23.64	205.02	197.70	70.41	91.68
NN	136.51	138.36	214.78	24.48	194.81	182.63	71.62	102.50
KNN	113.29	128.90	211.22	25.30	184.26	179.82	78.80	115.25
SVR	31.97	60.22	150.56	31.93	135.00	132.56	75.43	88.52
			Shallow Neural Learning					
SLSTM	-58.75	-51.89	113.47	37.13	98.53	90.24	61.24	78.21
SSRNN	-33.63	-20.34	111.38	37.06	95.25	88.40	50.46	79.28
SGRU	-56.17	-51.38	109.99	36.65	95.22	89.15	59.26	81.07
			Deep Learning					
DLSTM	-45.18	-34.22	108.48	36.22	100.38	94.83	**42.68**	**67.90**
DSRNN	-29.75	-30.52	106.79	38.58	96.44	87.81	48.56	74.20
DGRU	-28.51	-12.48	**102.98**	**40.07**	**89.16**	**80.13**	50.24	72.22

Table 4. Results for DS-2 (bleed valve) (ordered by RMSE/ best in bold).

	Bias		Accuracy				Robustness	
	ME	MdE	RMSE	RA	MAE	MdAE	MAD	SSD
			Classical Machine Learning					
NN	20.88	19.78	48.77	15.12	38.57	33.38	21.39	35.03
RF	20.87	24.82	37.03	22.72	32.63	31.62	13.03	19.04
GLM	18.33	28.35	35.64	20.22	32.93	33.17	9.82	16.28
KNN	19.51	27.82	35.72	20.48	31.86	31.90	13.00	15.94
SVR	20.60	27.91	35.13	21.25	32.09	33.00	8.82	12.71
			Shallow Neural Learning					
SGRU	13.12	7.89	31.35	29.99	28.27	27.04	7.71	13.84
SSRNN	14.11	12.88	31.10	27.16	27.79	26.29	7.76	12.98
SLSTM	13.70	10.59	30.78	27.56	27.79	25.56	8.85	13.76
			Deep Learning					
DGRU	9.91	7.45	29.17	27.09	24.80	24.28	5.56	10.13
DSRNN	2.27	6.68	23.43	32.02	20.91	18.74	**5.12**	10.13
DLSTM	-8.80	-2.54	**16.86**	**30.87**	**14.38**	**14.94**	5.82	**9.70**

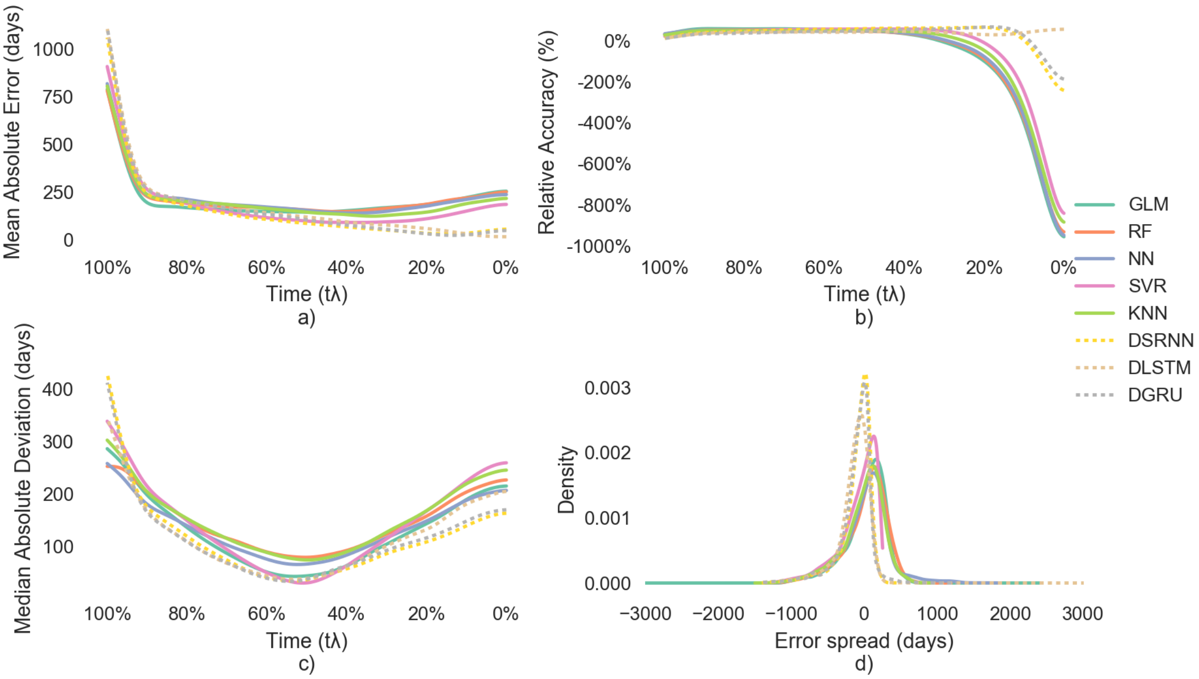


Figure 3. Performance on DS-1 (engine). The performance of machine learning and deep learning models is compared in a) mean absolute error, b) relative accuracy, c) median absolute deviation and d) error distribution. Here, $t\lambda$ is the fraction of time until the condition event.

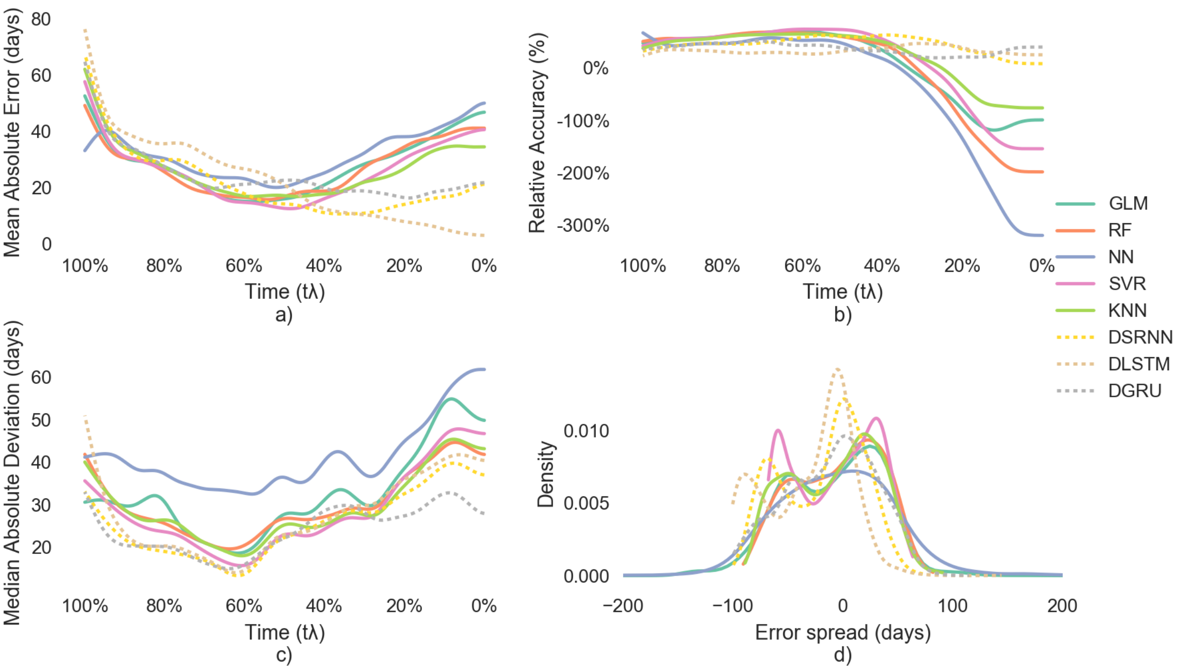


Figure 4. Performance on DS-2 (bleed valve). The performance of machine learning and deep learning models is compared in a) mean absolute error, b) relative accuracy, c) median absolute deviation and d) error distribution. Here, $t\lambda$ is the fraction of time until the health event.

Our goal with this work was to show that data-driven approaches based on deep recurrent neural networks can be more suited to prognostics than other methods. The novelty of our approach lied in a comprehensive analysis and comparison of data-driven techniques in two related real-world maintenance problems. The results are promising, indicating that deep learning and recurrent neural networks are powerful tools for further exploration in prognostics.

To conclude we wish to also shed some light on the limitations and constraints of these approaches. First, it is important to note that these are data-driven approaches and are therefore always dependent on the volume and quality of the data. Second, deep learning models tend to be black-box tools and are, to an even greater extent than classical machine learning, not easy to understand. Third, these models, especially recurrent networks, are difficult to train. These dimensions are future research directions worth exploring. Also, another research direction will be to combine the relative strengths of different deep learning techniques using more complex hybrid approaches, ensemble, and other system combination techniques.

REFERENCES

Adams, D. E. (2002). Nonlinear damage models for diagnosis and prognosis in structural dynamic systems. In *Aerosense* (pp. 180–191).

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37–66.

Atherton, D. (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks. *Journal of vibration and acoustics*, *121*, 355–362.

Baptista, M., de Medeiros, I. P., Malere, J. P., Nascimento, C., Prendinger, H., & Henriques, E. M. (2017). Comparative case study of life usage and data-driven prognostics techniques using aircraft fault messages. *Computers in Industry*, *86*, 1–14.

Baptista, M., Sankararaman, S., de Medeiros, I. P., Nascimento Jr, C., Prendinger, H., & Henriquesa, E. M. (2017). Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling. *Computers & Industrial Engineering*.

Baptista, M. L., de Medeiros, I. P., Malere, J. P., Nascimento, C. L., Prendinger, H., & Henriques, E. (2017). Aircraft on-condition reliability assessment based on data-intensive analytics. In *Aerospace conference, 2017 ieee* (pp. 1–12).

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Brownlee, J. (2014). Discover feature engineering, how to engineer features and how to get good at it. *Machine Learning Process*.

Cataltepe, Z., Abu-Mostafa, Y. S., & Magdon-Ismail, M. (1999). No free lunch for early stopping. *Neural computation*, *11*(4), 995–1009.

Chelidze, D., & Cusumano, J. P. (2004). A dynamical systems approach to failure prognosis. *Journal of Vibrations and Acoustics*, *126*(1), 2–8.

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

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Daigle, M., Saha, B., & Goebel, K. (2012). A comparison of filter-based approaches for model-based prognostics. In *Aerospace conference* (pp. 1–10).

Daigle, M. J., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *International journal of prognostics and health management*, *2*(2), 84–99.

de Pádua Moreira, R., & Nascimento, C. L. (2012). Prognostics of aircraft bleed valves using a svm classification algorithm. In *Aerospace conference* (pp. 1–8).

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Goebel, K., Saha, B., & Saxena, A. (2008). A comparison of three data-driven techniques for prognostics. In *62nd meeting of the society for machinery failure prevention technology (mfpt)* (pp. 119–131).

Goebel, K., Saha, B., Saxena, A., Celaya, J. R., & Christophersen, J. P. (2008). Prognostics in battery health management. *IEEE instrumentation & measurement magazine*, *11*(4).

Guo, L., Li, N., Jia, F., Lei, Y., & Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, *240*, 98–109.

Gupta, S., & Ray, A. (2007). Real-time fatigue life estimation in mechanical structures. *Measurement Science and Technology*, *18*(7), 1947.

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

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

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

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. *Advances in psychology*, *121*, 471–495.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Medjaher, K., Camci, F., & Zerhouni, N. (2012). Feature extraction and evaluation for health assessment and failure prognostics. In *Proceedings of first european conference of the prognostics and health management society, phm-e'12.* (pp. 111–116).

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, *135*(3), 370-384.

Ng, A. Y. (2004). Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on machine learning* (p. 78).

Oppenheimer, C. H., & Loparo, K. A. (2002). Physically based diagnosis and prognosis of cracked rotor shafts. In *Aerosense 2002* (pp. 122–132).

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Reed, R., & Marks, I. (1989). Neural smithing. supervised learning in feedforward artificial neural networks. *Cambridge, MA: MIT Press*.

Saha, B., & Goebel, K. (2009). Modeling li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society* (pp. 2909–2924).

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–17).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Schölkopf, B., & Burges, C. J. (1998). *Advances in kernel methods: support vector learning*. MIT press.

Seemann, R., Langhans, S., Schilling, T., & Gollnick, V. (2010). Modeling the life cycle cost of jet engine maintenance. *Technische Universität Hamburg-Harburg (TUHH), Hamburg*.

Song, Y., Li, L., Peng, Y., & Liu, D. (2018). Lithium-ion battery remaining useful life prediction based on grurnn. In *2018 12th international conference on reliability, maintainability, and safety (icrms)* (pp. 317–322).

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

Tang, L., Orchard, M. E., Goebel, K., & Vachtsevanos, G. (2011). Novel metrics and methodologies for the verification and validation of prognostic algorithms. In *Aerospace conference, 2011 ieee* (pp. 1–8).

Wu, Q., Yang, X., & Zhou, Q. (2012). Pattern recognition and its application in fault diagnosis of electromechanical system. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, *9*(8), 2221–2228.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... others (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*(1), 1–37.

Yan, W., & Yu, L. (2015). On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. In *Proceedings of the annual conference of the prognostics and health management society.*

Yuan, M., Wu, Y., & Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *Aircraft utility systems (aus), ieee international conference on* (pp. 135–140).

Zhang, S., Wu, Y., Che, T., Lin, Z., Memisevic, R., Salakhutdinov, R. R., & Bengio, Y. (2016). Architectural complexity measures of recurrent neural networks. In *Advances in neural information processing systems* (pp. 1822–1830).

Zhang, Z., Lu, J., Zhou, G., & Liao, X. (2018). Research on tool wear prediction based on lstm and arima. In *Proceedings of the 2018 international conference on big data engineering and technology* (pp. 73–77).

Zhao, R., Wang, J., Yan, R., & Mao, K. (2016). Machine health monitoring with lstm networks. In *Sensing technology (icst), 2016 10th international conference on* (pp. 1–6).

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237.

Zhao, R., Yan, R., Wang, J., & Mao, K. (2017). Learning to monitor machine health with convolutional bidirectional lstm networks. *Sensors*, *17*(2), 273.

BIOGRAPHIES

Marcia L. Baptista (BS and MSc. in Informatics and Computer Engineering. Instituto Superior Tecnico, Lisbon, Portugal, September 2008) holds a PhD from the Engineering Design and Advanced Manufacturing (EDAM) program under the umbrella of MIT Portugal. Her research focuses on the development of prognostics techniques for aeronautics equipment. Her research interests include data-driven modeling, prognostics, and deep learning.

Helmut Prendinger received his Master and Doctoral degrees in Logic and Artificial Intelligence from the University of Salzburg in 1994 and 1998, respectively. Since 2012, he is a full professor at the National Institute of Informatics (NII), Tokyo, after joining NII in 2004 as Associate Professor. Previously, he held positions as a research associate (2000 2004) and JSPS postdoctoral fellow (1998 2000) at the University of Tokyo, Dept. of Information and Communication Engineering, Faculty of Engineering. In 1996-1997, he was a junior specialist at the University of California, Irvine. His research interests include artificial intelligence including machine learning, intelligent user interface, cyber-physical systems, and the melding of real and virtual worlds, in which areas he has published more than 220 peer-reviewed journal and conference papers. His vision is to apply his research to establish the IT infrastructure for Unmanned Aerial Vehicles, or “drone”. He is a member of IEEE and ACM.

Elsa Maria Pires Henriques has a doctorate in Mechanical Engineering and is associated professor at Instituto Superior Tecnico in the University of Lisbon. She is responsible for the “Engineering Design and Advanced Manufacturing (LTI/EDAM)” post-graduation. During the last fifteen years, she has participated and/or coordinated several national and European R&D projects in collaboration with different industrial sectors, from tooling to automotive and aeronautics, mainly related to manufacturing, life cycle based decisions and management of complex design processes. She has a large number of scientific and technical publications in national and international conferences and journals. She was a national delegate in the 7th Framework Programme of the EU.