Aircraft Engines Remaining Useful Life Prediction with an Adaptive Denoising Online Sequential Extreme Learning Machine

Tarek Berghout
Laboratory of Automation and Manufacturing Engineering, University of Batna 2, Batna 05000, Algeria
berghouttarek@gmail.com

Leïla-Hayet Mouss
Laboratory of Automation and Manufacturing Engineering, University of Batna 2, Batna 05000, Algeria
h.mouss@univ-batna2.dz

Ouahab Kadri
Department of Computer Science, University of Batna 2, Batna 05078, Algeria
o.kadri@univ-batna2.dz

Lotfi Saïdi
SIME_Lab (LR 13ES03), Université de Tunis, Tunisia;
Institut de Recherche Dupuy de Lôme (UMR CNRS 6027), University of Brest, 29238 Brest, France
Lotfi.Saidi@esstt.rnu.tn

Mohamed Benbouzid
Institut de Recherche Dupuy de Lôme (UMR CNRS 6027), University of Brest, 29238 Brest, France
Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China
Mohamed.Benbouzid@univ-brest.fr

Abstract: Remaining Useful Life (RUL) prediction for aircraft engines based on the available run-to-failure measurements of similar systems becomes more prevalent in Prognostic Health Management (PHM) thanks to the new advanced methods of estimation. However, feature extraction and RUL prediction are challenging tasks, especially for data-driven prognostics. The key issue is how to design a suitable feature extractor that is able to give a raw of time-varying sensors measurements more meaningful representation to enhance prediction accuracy with low computational costs. In this paper, a new Denoising Online Sequential Extreme Learning Machine (DOS-ELM) with double dynamic forgetting factors (DDFF) and Updated Selection Strategy (USS) is proposed. First, depending on the characteristics of the training data that comes from aircraft sensors, robust feature extraction using a modified Denoising Autoencoder (DAE) is introduced to learn important patterns from data. Then, USS is integrated to ensure that only the useful data sequences pass through the training process. Finally, OS-ELM is used to fit the non-accumulative linear degradation function of the engine and to address dynamic programming by trucking the new coming data and forgetting gradually the old ones based on the proposed DDFF. The proposed DOS-ELM is tested on the public dataset of commercial modular aeropropulsion system simulation (C-MAPSS) of a turbofan engine and compared with OS-ELM trained with ordinary Autoencoder (AE), basic OS-ELM and pervious works from the literature. Comparison results prove the effectiveness of the new integrated robust feature extraction scheme by showing more stability of the network responses even under random solutions.

Keywords: ELM, OS-ELM, forgetting mechanism, denoising autoencoder, updated selection strategy, C-MAPPS, remaining useful life.

Nomenclature

AE	Autoencoder	m	Total number of training mini-batches
AOS-ELM	Adaptive OS-ELM	X	Inputs of each mini-batch
CBM	Conditional based maintenance	T	Targets of each mini-batch
C-MAPSS	Commercial modular aeropropulsion system simulation	W	Input weights
DAE	Denoising autoencoder	B	Biases vector
DDFF	Double dynamic forgetting factors	N	Number of training samples
DOS-ELM	Denoising online sequential ELM	K	Gain matrix
ELM	Extreme learning machine	β	Output weights
HPC	High pressure compressor	H	Hidden layer
HPT	High pressure turbine	e	Approximation error
LPC	Low pressure compressor	C_r	Corruption noise
LPT	Low pressure turbine	γ	Magnitude of the noise
OS-ELM	Online sequential ELM	$(_{OS})$	Subscript indicating that the variable belongs to OS-ELM
RLS	Recursive least squares	$(_{AE})$	Subscript indicating that the variable belongs to AE
RUL	Remaining useful life	$(^{-1})$	Superscript refers to pseudo-inverse of the matrix
SLFN	Single hidden layer feedforward neural network	$(^{T})$	Superscript refers to the transpose matrix
SVD	singular value decomposition	λ	Dynamic forgetting factor
k	Index of the training mini-batch	P	Covariance matrix
		d	Difference between estimated RUL and the desired one

1. Introduction

Nowadays, and as a result of the remarkable evolution in growth, variety, and velocity of data due to the advancement in sensors technology, the estimation of RUL of components or subsystems based on available historical data becomes highly recommended and worth enough to motivate researchers towards new prediction approaches [1], [2], [3].

RUL estimation of aircraft engines for engine health assessment is a very crucial task in Conditional Based Maintenance (CBM) operations [4]. CBM operations are related to the actual health state of the equipment or subsystems under operating conditions. As a result, the accurate earlier prediction of failures involves an informed maintenance decision making, which can avoid probable disasters [5].

The complexity of RUL estimation in term of data-driven approaches lies to the multiple accumulated sequences of higher dimensional time-varying data. This flooded accumulation can easily obscure the loss function of the training model from converging to the desired error. The best solution for this problem or time-varying data problems, in general, is to integrate a dynamic tracking and selection strategy in the data-driven models to resist against any variation in the new coming chunks [6], [7], [8], [9]. However, such a solution is feasible under the constraint that the training data are not contaminated with noise with unknown source or behavior [10], [11].

Therefore in this paper, a new data-driven approach based on DOS-ELM with double dynamic forgetting factors (DDFF) and updated selection strategy (USS) is introduced. The proposed algorithm aims to achieve an accurate RUL estimation by proposing these solutions:

- Denoising autoencoder (DAE) trained with Online Sequential Extreme Learning Machine (OS-ELM) for best features representations and noise reduction under time-varying data;
- An OS-ELM for the RUL estimation;
- Dynamic tracking ability of new data is integrated into both DAE and OS-ELM using a proposed DDFF;
- USS is utilized to select only the new expressive samples for the training process.

The proposed approach is validated on the public dataset C-MAPPS of turbofan engines [12], the results are compared to OS-ELM trained with an ordinary autoencoder and basic OS-ELM, where higher performances are recorded.

The remaining parts of this paper are organised as follows: in section 2, an investigation about some important related works is introduced. A brief description of the used dataset is presented in section 3. Section 4 elaborates on the proposed algorithm used in this work. Section 5 illustrates and explains experimental results, where the performances of the proposed training process are showcased and compared to other data-driven methods. Finally, the paper is concluded in Section 6.

2. Related works

Many recent approaches were introduced to attempt an accurate RUL estimation and enhancing reliability by reducing unneeded maintenance operations and streamline activities [5]. Different architectures such as hybrid models, ensemble algorithms and deep learning were involved as a decision-making tool for the planned maintenance policy.

For instance, Bai *et al.* [11] introduced a new fast training approach for RUL prediction based on a "single-batch" ELM after an appropriate features selection and noise filtering for accurate estimation. Bektas *et al.* [10] proposed multi-regime normalization approaches followed by filtering methods for best features representation and noise reduction before feeding the feed-foreword neural network for RUL prediction. Lu *et al.* [11] improved the recursive learning of the OS-ELM by reducing the noise produced from Recursive Least Squares (RLS) linear approximation with Kalman filter objective function and proposed a new weights adaptation method to accurately predict the RUL in aircraft engines. Djeziri *et al.* [14] proposed a data-driven method based on data augmented in simulation to take into consideration every possible trends of the degradation system process, then, the dataset is used for an offline estimation of the RUL within confidence bound. The robustness of the proposed RUL estimation approach to the changes in Condition Monitoring is performed online by updating the model parameters.

It is worth to be mentioned that all cited works can be classified as hybrid models depending on human intervention whose aim is to predict RUL accurately based on available historical data via an accurate features reconstruction, filtering or the best parameters tuning of the training model. However, the effect of the future accumulated time-varying mini-batches on the behavior of the approximation function and the divergence of training parameters has not been discussed.

Recently, a family of deep learning models has emerged intending to learn meaningful representation at higher level of data complexity. Ma *et al.* [15] used recent deep learning techniques to ensure that contaminated noise in historical sensors measurements is reduced by training a stack of denoising autoencoders with multiple types of noise before feeding the training model of RUL estimation. Xia *et al*. [16] constructed two-stage deep neural networks for robust feature extraction based on a stack of DAEs and shallow neural networks for estimating the RUL. Li *et al.* [17] used a time scaling window for better features representations combined with a deep convolution neural network for an accurate RUL prediction. The deep learning models used in these works addresses dynamic programming and less human intervention by updating the training model in every driven mini-batch. However, multiple hidden layers with integrated features selections paradigm based on conventional training algorithms like backpropagation or contrastive divergence require more algorithmic complexity. Besides, blinded updating with any driven sequences without looking to its importance on the training model can easily drive the approximation function away from the desired target, regardless to the learning rate tuning issues and local minima problems.

Throughout this analysis, the main open challenges of RUL prediction are:

- Updating the training model toward the new health state of the studied equipment by giving the well-needed attention to the new driven mini-batches;
- Measurements noise reduction or canceling to guarantee a reliable source of information;
- Reducing time-consuming during the training process.

In the recent decade, OS-ELM which is one of the ELM variants proposed for a Single Hidden Layer Feed-forward Neural network (SLFN), has been widely used in prediction fields due to its data-driven interaction, fast training and fewer parameters tuning based on a recursive best fit of linear function approximation [18]. Zhang *et al*. [19] developed a deep multilayer denoising ELM to extract meaningful representations and it is applied to image classification and proved its accuracy. Cheng *et al*. [20] build a feature space based on a new deep denoising ELM with sparse representation of the hidden layers based on Singular Value Decomposition (SVD) for better learning of features from the training set. Li *et al.* [6] introduced their novel OS-ELM with dynamic tracking ability of newly coming data for the sake of addressing a data-driven approach for an accurate prediction of a gas utilization ratio. As well as they proposed an updated selection strategy to reduce the training samples by selecting only the important ones for the training process. In a recent paper by Chin *et al.* [7] an adaptive online sequential ELM (AOS-ELM) is proposed to predict the frequency-dependent sound pressure level data of several boxes onboard of the offshore platform, it was shown that AOS-ELM allows a gradual increase in the dataset that is hard to find during the initial design stage of the offshore platform.

In the framework of the above-mentioned works, we propose a very fast data-driven neural network with only a single hidden layer. The network is designed to tune all the hidden nodes parameters for appropriate feature extraction and learning. USS and DDFF are proposed to adapt the neural network dynamically with the variation of newly arrived chunks of data attempting to achieve better accuracy with minimum training samples and less human intervention.

The new algorithm is different in terms of adaptive learning, feature extraction and architecture. This learning model is capable of dynamic adaptation according to environmental variables and the physical state of the studied systems by paying attention to the new arrived samples. In addition to that, it allows a robust extraction of features by pushing the hidden layer of the SLFN towards a more significant mapping, while reducing the algorithmic complexity and the computation time by keeping only one hidden layer.

3. Dataset description

In the commercial modular aeropropulsion system simulation (C-MAPSS) software, the simulated engine is a tow spool turbofan engine with high level of thrust up to 400340 (Newton), which appears in the diagram of Fig. 1. The engine operates under operating conditions ranged from sea level to 12192 (meters) of altitude and ambient temperature vary from -51C° to 39 C°. In this type of engine, the contaminated air generated by the fan is compressed in two stages passing by low and high-pressure compressors (LPC) and (HPC). After that, in the combustion chamber the compressed air is heated to produce enough thrust to drive both low and high-pressure turbines, LPT and HPT. The thrust power will be produced by both air generated by the fan in the bypass stream and the air entering the engine core [4].

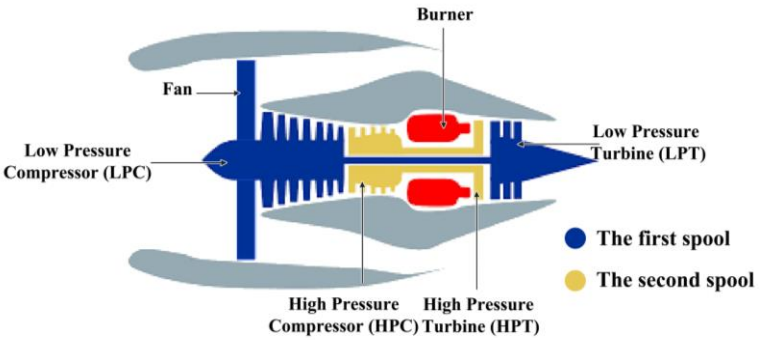

Fig. 1. diagram of the studied engine [12].

Retrieved data from C-MAPSS software is provided for the public as a benchmark for comparing and studying RUL data-driven predictions models for aircraft engines [12]. The dataset describes different faults modes under different engine operating conditions and it consists of four different subsets (FD001, FD002, FD003, and FD004) where each subset is divided into training and testing sets.

The dataset contains a multivariate time series (life cycles) of 26 features (engine number, time cycles, operating conditions, sensors measurements) of 100 different degradation profiles of similar engines. At the beginning of each degradation profile, the engine is considered as normally functioning under the initial operating conditions. And then, at a certain level of operating cycles the engine starts losing its performances towards the failure mode. Fig. 2 illustrates the attitude of sensors measurements against the linear piecewise degradation function that is defined according to PHM data challenge for each engine from the dataset [5], [21].

The different colors in the sensors measurements represent different types of information recorded from multiple sensors from the C-MAPPS turbofan engine simulator. The most attractive challenge about the dataset is that sensors measurements are contaminated with noise from different sources. In our work, only the first dataset FD001 is used to evaluate the proposed algorithm.

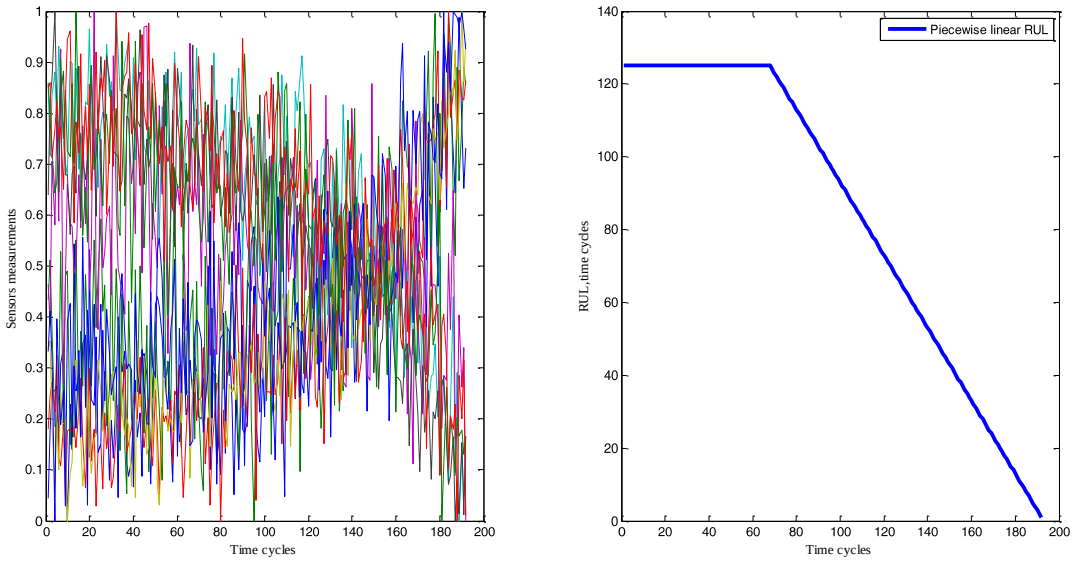

Fig. 2. Example of data variation in one degradation profile from the first dataset FD001.

4. Proposed algorithm

To introduce our proposed DOS-ELM, we have to establish first the most important basic learning rules of OS-ELM, by elucidating steps and mathematical equations used during training, the equations and methodology of OS-ELM are used from [16] with some changes in the notations to make the learning rules of the DOS-ELM better illustrated. Concerning the basic rules of the ordinary autoencoder are used from [21] and those associated with the DAE were inspired from [22].

Before any training process for both algorithms we should normalize the training and testing inputs in the range [0 1] based on min-max normalization.

4.1. OS-ELM

The learning rules of OS-ELM based on a given set of training mini-batches $\{X_{k+1}, T_{k+1}\}_{k=0}^{m}$ where m represents the number of the training mini-batches are:

Initial phase (k=0):

- Randomly generate the input weights matrix and the biases vector (W,B) normalized between -1 and 1;
- Calculate the hidden layer H_0 according to equation (1) where G is the activation function.
- Calculate the initial covariance matrix P_0 according to equation (2);
- Determine the initial output weight matrix β_0 as given in equation (3);

Updating phase (k=k+1):

- Calculate the hidden layer H_{k+1} for the new mini-batch as shown in equation (1).
- Update the output weight matrix β_{k+1} as given in equation (4) and using equations (5), (6), and (7).

$$H_{k+1} = G(W \cdot X_{k+1} + B) \tag{1}$$

$$P_0 = (H_0^T H_0)^{-1} \tag{2}$$

$$\beta_0 = P_0 H_0^T T_0 \tag{3}$$

$$\beta_{k+1} = \beta_k - P_{k+1} H_{k+1}^T e_{k+1} \tag{4}$$

$$e_{k+1} = T_{k+1} - H_{k+1}\beta_k \tag{5}$$

$$P_{k+1} = P_k - K_{k+1} H_{k+1}^T P_k \tag{6}$$

$$K_{k+1} = \frac{P_k H_{k+1}}{H_{k+1}^T P_k H_{k+1}^T} \tag{7}$$

The updating process is repeated for any coming training mini-batches.

4.2. Proposed DOS-ELM

The proposed training scheme of DOS-ELM for RUL prediction is shown in Fig. 3. Two learning networks are used to train an over-complete hidden layer of the DOS-ELM. The DAE is used for the sake of learning meaningful representation from the training inputs by reducing the noise coming with them same as mentioned in [23]. The DAE reconstruction function attempts to reconstruct the original inputs from the corrupted ones.
DDFF is proposed based on [6] and integrated into the DAE and OS-ELM updating phases to make the DOS-ELM training model dynamically discard old training data by giving the attention to only the new ones. The DDFF functions are illustrated in equation (16) for both the DAE and DOS-ELM. The USS strategy is proposed based on [6] to make the training model accepts only the useful mini-batches during training.

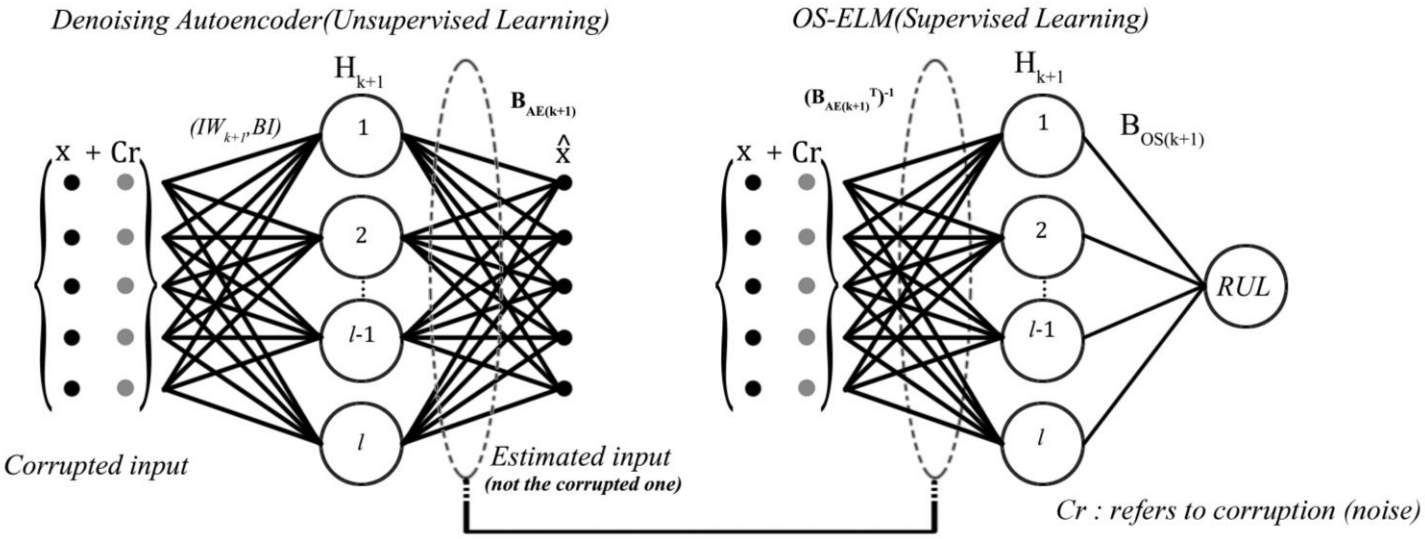

Fig. 3. Proposed DOS-ELM for RUL prediction.

The training steps of the proposed DOS-ELM for the same given training set of OS-ELM are:

Initial phase (k=0):

- Randomly generate inputs weights and biases the same as in OS-ELM;
- Generate a random noise Cr_{k+1} from a stochastic distribution normalized between 0 and 1;
- Corrupt the initial mini-batch X_{k+1} by adding the noise with the user desired magnitude $\gamma \in [0,1]$, and calculate the hidden layer H_{k+1} as shown in equation (8);
- Calculate the initial output weights $\beta_{AE(k+1)}$ and $\beta_{os(k+1)}$ of the DAE and DOS-ELM as given in equation (10) and (11), respectively;
- Calculate the initial reconstruction error $e_{AE(k+1)}$ and prediction $e_{os(k+1)}$ error from equations (14) and (15).

Updating phase(k=k+1):

- Replace the input weights of the encoder as illustrated in equation (9) (this formula is proposed and will be discussed after representing the remaining training steps);
- Calculate the new reconstruction and prediction errors;
- Update both $\beta_{AE(k+1)}$ and $\beta_{os(k+1)}$ according to the USS condition presented in equations (12)and (13) based on the new updated formulas of estimation error, covariance and gain where the DDFF is integrated with all of them;
- Finally, the forgetting factor should always be adjusted according to lower band and upper bands constraints which equal to 0 and 1, respectively.

$$H_{k+1} = G(W_{k+1}(X_{k+1} + \gamma Cr_{k+1}) + B) \tag{8}$$

$$W_{k+1} = \begin{cases} W, k=0 \\ (\beta_{AE(k+1)})^{-1}, k>0 \end{cases} \tag{9}$$

$$\beta_{AE(0)} = P_0 H_0^T X_0 \tag{10}$$

$$\beta_{os(0)} = P_0 H_0^T T_0 \tag{11}$$

$$\beta_{AE(k+1)} = \begin{cases} \beta_{AE(k)} - P_{AE(k+1)} H_{k+1}^T e_{AE(k+1)}, e_{OS(k+1)} \leq e_{OS(k)} \\ \beta_{AE(k)}, e_{OS(k+1)} \geq e_{OS(k)} \end{cases} \tag{12}$$

$$\beta_{OS(k+1)} = \begin{cases} \beta_{OS(k)} - P_{OS(k+1)} H_{k+1}^T e_{OS(k+1)}, e_{OS(k+1)} \leq e_{OS(k)} \\ \beta_{OS(k)}, e_{OS(k+1)} > e_{os(k)} \end{cases} \tag{13}$$

$$e_{AE(k+1)} = X_{k+1} - H_{k+1}\beta_{AE(k)} \tag{14}$$

$$e_{OS(k+1)} = T_{k+1} - H_{k+1}\beta_{OS(k)} \tag{15}$$

$$P_{AE(k+1)} = \frac{1}{\lambda_{AE}}(P_{AE(k)} - K_{AE(k+1)}H_{k+1}^{T}P_{AE(k)}) \tag{16}$$

$$P_{OS(k+1)} = \frac{1}{\lambda_{os}}(P_{OS(k)} - K_{OS(k+1)}H_{k+1}^{T}P_{OS(k)}) \tag{17}$$

$$K_{AE(k+1)} = \frac{P_{AE(k)}H_{k+1}}{\lambda_{AE} + H_{k+1}^{T}P_{AE(k)}H_{k+1}^{T}} \tag{18}$$

$$K_{OS(k+1)} = \frac{P_{OS(k)}H_{k+1}}{\lambda_{OS} + H_{k+1}^{T}P_{OS(k)}H_{k+1}^{T}} \tag{19}$$

$$\lambda_{AE(k+1)} = \lambda_{min} + (1 - \lambda_{min})e^{-\mu\left\|e_{AE(k+1)}\right\|} \tag{20}$$

$$\lambda_{os(k+1)} = \lambda_{min} + (1 - \lambda_{min})e^{-\mu\left\|e_{os(k+1)}\right\|} \tag{21}$$

It is mentioned in training rules of autoencoder in [21] that the encoding process can be achieved using equation (22), but depending on the same formula that is mentioned in [21] and shown in equation (23) it is mathematically proven that equation (23) will give a better representation, this formula has already been tested and its accuracy proved [25].

$$H = X\beta_{AE}^{T} \tag{22}$$

$$X = H\beta_{AE} \Rightarrow H = X(\beta_{AE}^{T})^{-1} \tag{23}$$

Further details about the training rules of the proposed approach can be presented in the pseudo code of Algorithm 1.

Algorithm 1. DOS-ELM

Inputs: $X,T,l,G,\lambda_{\min},\gamma,m$

Outputs: β_{AE},β_{os}

%% Initial phase (*k=0*)

Generate inputs weights and biases $\{W,B\}$;

Add noise and calculate the hidden layer : H_{k+1} $H_{k+1} = G(W_{k+1}(X_{k+1} + \gamma Cr_{k+1}) + B)$;

Calculate the initial covariance matrix of the hidden layer: $P_0 = (H_0^{T}H_0)^{-1}$;

Determine the initial output weights of the DAE : $\beta_{AE(0)} = P_0H_0^{T}X_0$;

Determine the initial output weights of the DOS-ELM : $\beta_{OS(0)} = P_0H_0^{T}T_0$;

Calculate the initial reconstruction error $e_{AE(k+1)}$: $e_{AE(k+1)} = X_{k+1} - H_{k+1}\beta_{AE(k)}$;

Calculate the initial prediction error $e_{os(k+1)}$: $e_{OS(k+1)} = T_{k+1} - H_{k+1}\beta_{OS(k)}$;

%% Updating phase (*k=k+1*)

For *k=1:m;*

Update the input weights: $W_{k+1} = \begin{cases} W, k=0 \\ (\beta_{AE(k+1)})^{T}, k>0 \end{cases}$;

Calculate the new reconstruction error $e_{AE(k+1)}$ of the DAE: $e_{AE(k+1)} = X_{k+1} - H_{k+1}\beta_{AE(k)}$;

If $e_{AE(k+1)} > e_{AE(k)}$ **;** *%% USS conditions*

$\lambda_{AE(k+1)} = \lambda_{min} + (1 - \lambda_{min})e^{-\mu\left\|e_{AE(k+1)}\right\|}$;*%% Update forgetting parameter*

If $\lambda_{AE(k+1)} > 1$ *%% Forgetting parameter constraints*

$\lambda_{AE(k+1)}$ =1;

Else if $\lambda_{AE(k+1)}$ <0

$\lambda_{AE(k+1)}$ =0;

End

$K_{AE(k+1)} = \frac{P_{AE(k)}H_{k+1}}{\lambda_{AE}+H_{k+1}^{T}P_{AE(k)}H_{k+1}^{T}}$;%% *Update gain matrix.*

$P_{AE(k+1)} = \frac{1}{\lambda_{AE}}(P_{AE(k)} - K_{AE(k+1)}H_{k+1}^{T}P_{AE(k)})$;%% *Update covariance matrix.*

$\beta_{AE(k+1)} = \beta_{AE(k)} - P_{AE(k+1)}H_{k+1}^{T}e_{AE(k+1)}, e_{OS(k+1)} \le e_{OS(k)}$; %% *Update output weights matrix.*

End

Calculate the new prediction error $e_{os(k+1)}$: $e_{OS(k+1)} = T_{k+1} - H_{k+1}\beta_{OS(k)}$;

If $e_{os(k+1)} > e_{os(k)}$ %% *USS conditions*

$\lambda_{os(k+1)} = \lambda_{min} + (1 - \lambda_{min})e^{-\mu\left\|e_{os(k+1)}\right\|}$; %% *Update forgetting parameter.*

If $\lambda_{os(k+1)}$ >1 %% *Forgetting parameter constraints.*

$\lambda_{os(k+1)}$ =1;

Else if $\lambda_{os(k+1)}$ <0;

$\lambda_{os(k+1)}$ =0;

end

$K_{OS(k+1)} = \frac{P_{OS(k)}H_{k+1}}{\lambda_{OS}+H_{k+1}^{T}P_{os(k)}H_{k+1}^{T}}$; %% *Update gain matrix.*

$P_{OS(k+1)} = \frac{1}{\lambda_{os}}(P_{OS(k)} - K_{OS(k+1)}H_{k+1}^{T}P_{OS(k)})$; %% *Update covariance matrix.*

$\beta_{OS(k+1)} = \beta_{OS(k)} - P_{OS(k+1)}H_{k+1}^{T}e_{OS(k+1)}, e_{OS(k+1)} \le e_{OS(k)}$; %% *Update output weights matrix.*

End

End

5. Numerical simulations and discussions

Before going further for any explanation, we have to mention that the comparative analysis is carried out using the hyper-parameters in Table 1, where these parameters values are experimentally assigned by retraining of the proposed DOS-ELM several times. These parameters were obtained simply by exhaustive searching based on a grid search through a manually specified set of hyperparameters.

Table 1. Training hyper-parameters.

λ_{min}	μ	l	G	*Mnin-batch size*
0.98	0.001	100	ReLU	Varied size (life cycle size)

It is mentioned in the OS-ELM learning rules that it is capable of learning data in pieces of various or fixed sizes [6]. As C-MAPSS training set is already organized in separate life cycles with different sizes, there is no need to divide it again. In fact, avoiding such unnecessary operations will reduce the computation time and the algorithmic complexity.

The proposed DOS-ELM is applied on the FD001 dataset of the C-MAPSS turbofan engine. The performances of the training algorithms are evaluated according to two metric functions in addition to training time; the score function and Root Mean Squared Error (RMSE) expressed in equations (21) and (22), respectively where proposed in PHM data challenge [12]. In PHM data challenge, which is a competition open to all potential conference attendees, the main

objective goal is to estimate RUL using machine learning techniques and depending on C-MAPSS dataset which is retrieved from a physical model. More details about this competition and its rules can be found in 'DASHlink ' the a collaborative sharing network for researchers in the data mining and systems health fields in [26].

The proposed score function used as an objective function to measure the exactitude of the prediction model by increasing the penalty of late and early predictions. Late prediction are considered more harmful then early ones and this is the reason behind setting its penalty factor equal 13 rather than 10 for early ones that may produce no harm but consume more maintenance resources. According to [12], These penalties represent costs of early or late predictions, and corresponding values are determined by the application at hand and risk postures. For this dataset, those penalties were determined at these numbers based on background information and were used for consistent evaluations thereafter. And because the precision function is weighted for some maintenance reasons, it will not be able to process the actual difference between the estimated RUL and the desired RUL. therefore, an additive metric such as a RMSE should be considered to confirm the reliability of the score results.

$$s = \begin{cases} \sum_{i=1}^{N} e^{-d/13} - 1, d_i \geq 0 \\ \sum_{i=1}^{N} e^{-d/10} - 1, d_i < 0 \end{cases} \tag{21}$$

$$R^{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d^2} \tag{22}$$

Before illustrating the final results of this experiment it is better to explain the effect of each contribution separately on the training of DOS-ELM passing by DAE, DDFF and USS strategy.

5.1. Denoising process

In this experiment, we have used one single type of noise. The noise is generated from Gaussian normal distribution according to the user-desired ratio and magnitude for training mini-batches.
To make the values of the input samples still normalized between 0 and 1 to satisfy the ELM learning constraints even after corruption, the noise is normalized between 0 and 1 before it is multiplied with the user desired magnitude factor γ and mixed with input mini-batches. After that, this mixture normalized again with min-max normalization between 0 and 1. Fig. 4 clearly addresses the input corruption process with $\gamma = 0.25$ and a corruption ratio equal to 0.60 for a chosen time series from the training set.

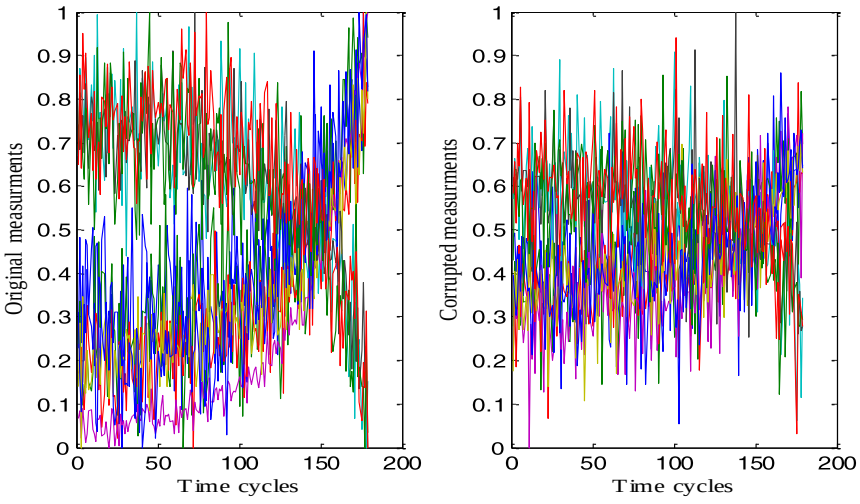

Fig. 4. Training data inputs corrupted with noise.

Before comparing the algorithm with the previously mentioned ones, we have tested the effect of the corruption ratio and magnitude on the prediction accuracy after multiple runs of the training model and the results are shown in Table 2.

Results from Table 2 show that by increasing the magnitude of noise or the corruption ratio gradually, the prediction model starts losing its aptitude for an accurate approximation. However, comparing these results later to the original version of the algorithm with no denoising scheme, the results will explain that the additive noise will help improve the accuracy of the prediction at certain threshold, and after that, the learning samples will be distorted and important patterns will begin to vanish.

Table 2. The effect of magnitude on the prediction of RUL.

Noise ratio	γ	R^{MSE} of testing	S
0.01	**0.01**	**12.3273**	**198.7692**
0.05	0.01	12.6805	204.5811
0.40	0.09	13.6003	295.7252
0.60	0.09	14.3125	319.5467
0.80	0.09	14.1594	383.9508
0.80	0.15	15.5208	719.5608
0.95	0.25	16.1717	798.3142
1.00	0.60	28.9643	975.1546

5.2. Forgetting mechanism

In this experiment, two dynamic forgetting functions are integrated into both DAE and DOS-ELM to control the approximation function of RUL estimation. The forgetting factors are changed dynamically to adapt the DOS-ELM to the new coming data. Figure 5 illustrates the changing of both dynamic forgetting factors according to the time-varying mini-batches.

We can see from the illustrated behaviors of DDFF functions in Fig. 5 for both DAE and DOS-ELM that each one of the forgetting factors increases gradually from λ_{min} towards λ_{max} This velocity of forgetting can be controlled using the sensitivity factor μ which is related to the number of mini-batches decided by the user (the higher the number of mini-batches the lower the sensitivity factor must be chosen).

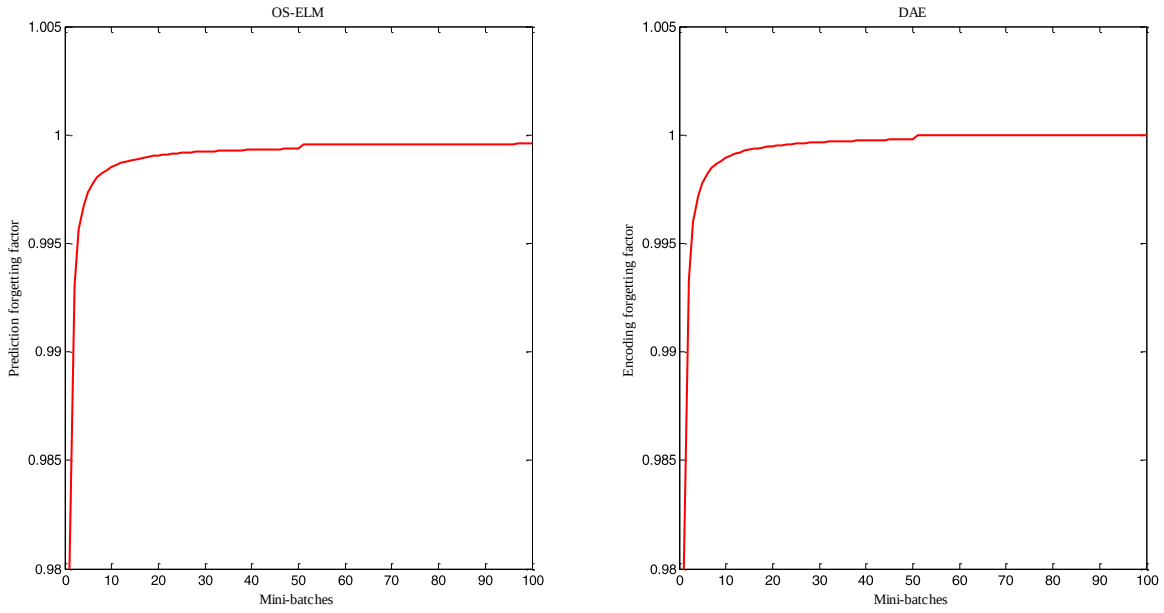

Fig. 5. Behavior of DDFF during the training of both the DAE and OS-ELM.

In the two curves which behave differently, and unlike the old OS-ELM forgetting mechanism which generally depends on a single assigned static hyperparameter [27], the new dynamic forgetting mechanism uses a forgetting factor which changes in form of exponential function, allows the learning weights to evolve gradually to adapt the learning model to the most significant variations of the concerned samples.

5.3. Updated Selection Strategy

The proposed USS is inspired from [6] were it was used only for supervised learning to reduce the number of accumulated mini-batches during training by ignoring the unwanted ones. In our main work the USS is integrated for both supervised and unsupervised learning paths. The decision of updating both β_{AE} and β_{OS} depends on the RMSE of reconstruction and prediction errors that are shown previously in equation (22).

Figure 6 proves that the USS algorithm is working as it is suggested in equations (12) and (13), the red points represent the RMSE of RUL error for each selected mini-batches. The error in each selected mini-bath is higher than the RMSE of the previous one, which satisfies USS function constraints.

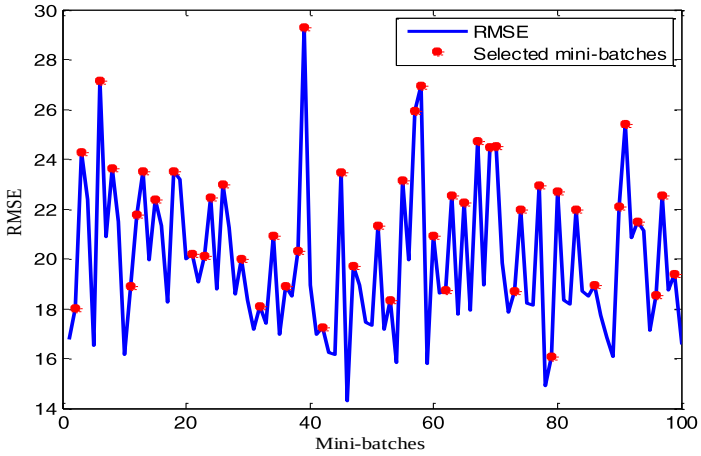

Fig. 6. USS strategy response to the undesirable mini-batches.

The main objective of these selected samples is to reduce the risk of overfitting as well as the computation time during the processing of these samples by keeping the same precision which will be explained in the following sections.

5.4. Results Comparison

To ensure that the new training rules of adaptation (USS and DDFF) given for a single hidden layer and the denoising algorithm (DAE) are capable of improving both OS-ELM online learning and the adaptive learning predictions, The performances of the proposed DOS-ELM are compared with basic OS-ELM and OS-ELM trained with an ordinary autoencoder with same way as the proposed methodology and only without USS strategy and DDFF.

The results of Table 3 show that DOS-ELM allows a more precise prediction than the other variants despite the fact that this variant uses all the training samples and that DOS-ELM only reads 47 mini-batches thanks to the USS strategy.

In a previous work [28], the results were obtained on the basis of a mapping of specific characteristics carried out via a stack of ordinary autoencoders. The learning rules were reinforced by a temporal memory called time difference. However, considering the new denoising system, new learning rules further improve the results due to the reduced noise of the training samples. Consequently, this confirm that the new learning paradigm can be a worth recommending approach for future data-driven prediction.

Table 3. Comparison of scoring results.

Methods	RMSE	Score	Training time (s)	Number of mini-batches
Basic OS-ELM	13.1792	230.9170	3.6972	100
OS-ELM with ordinary Autoencoder	12.8721	213.6580	**2.1216**	100
DOS-ELM	**12.2919**	**189.7695**	7.4568	**47**

In Fig. 7 and concerning the ordinary OS-ELM and the DOS-ELM variant, the metric scoring functions behaviors have more sparseness towards higher values then the others in the proposed DOS-ELM**.** This confirms the robustness of patterns learning from the new feature mapping of the denoising autoencoder with the adaptive learning paradigm.

During the training process, the denoising autoencoder try to push the hidden layers to learn the important patterns from corrupted samples, which consequently increases the quality of the representations of the features leading to guarantee an accurate estimation. Adaptive learning with a forgetting mechanism and an updated selection strategy allows dynamic trucking and reduction of training data which contributes in the minimization of structural risks and overfitting.

In this case also, and by comparing the error values of both late and early predictions, The RMSE in Fig. 7 explains that the RUL estimated in early predictions is more accurate than that in late predictions which provides clarification on the specification of predictions during the use of recursive least squares learning.

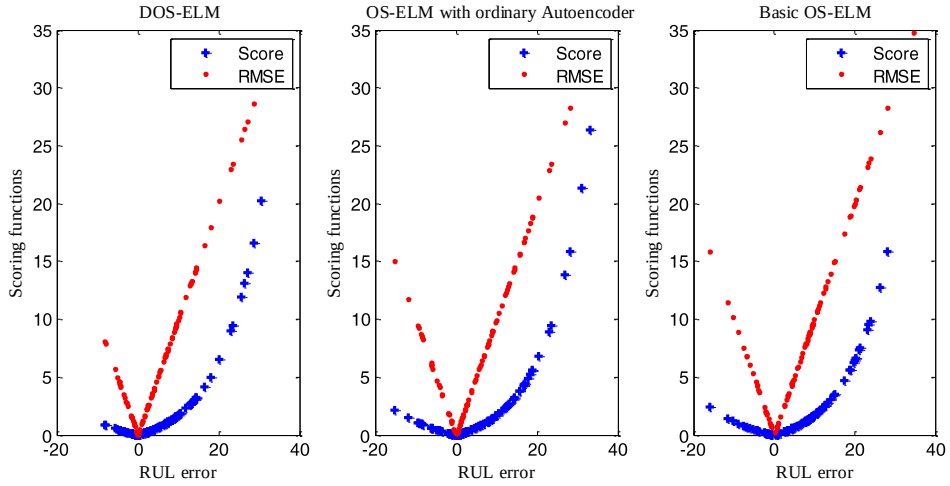

Fig. 7. Scoring functions of the studied data-driven approaches.

For graphic illustrations of the behavior of network responses to sensor measurements in each degradation profile, a set of life cycles of the turbofan engine are randomly chosen from the test set of FD001. Figure 8 gives a comparative example of curve fitting of RUL target function with the studied algorithms. It is observed that the DOS-ELM have less late prediction distance from the desired RUL and gives a better curve fit for the RUL function then the other algorithms compared with. These less late prediction errors explain the importance of the adaptive learning and filtering process to reduce the risks associated with the poorly planned maintenance policy.

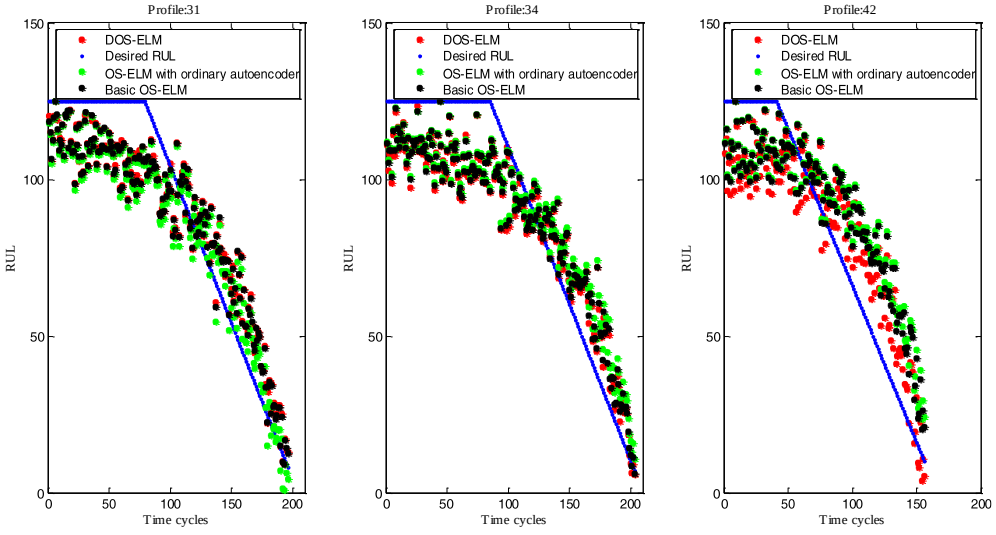

Fig. 8. Responses of the studied networks to run-to-failure measurements for a set of engines from the test set.

In ELM theories [29], and regardless of overfitting and other ill-posed problems, the more hidden nodes there are, the more precision can be achieved. Therefore, in order to explain the results of Fig. 8 and study the accuracy of the proposed algorithm to confirm its credibility in different circumstances and random parameters, Figure 9 is elucidates the

performances of the studied models during the incrimination of the number of hidden nodes by following both the validation and the training paths.

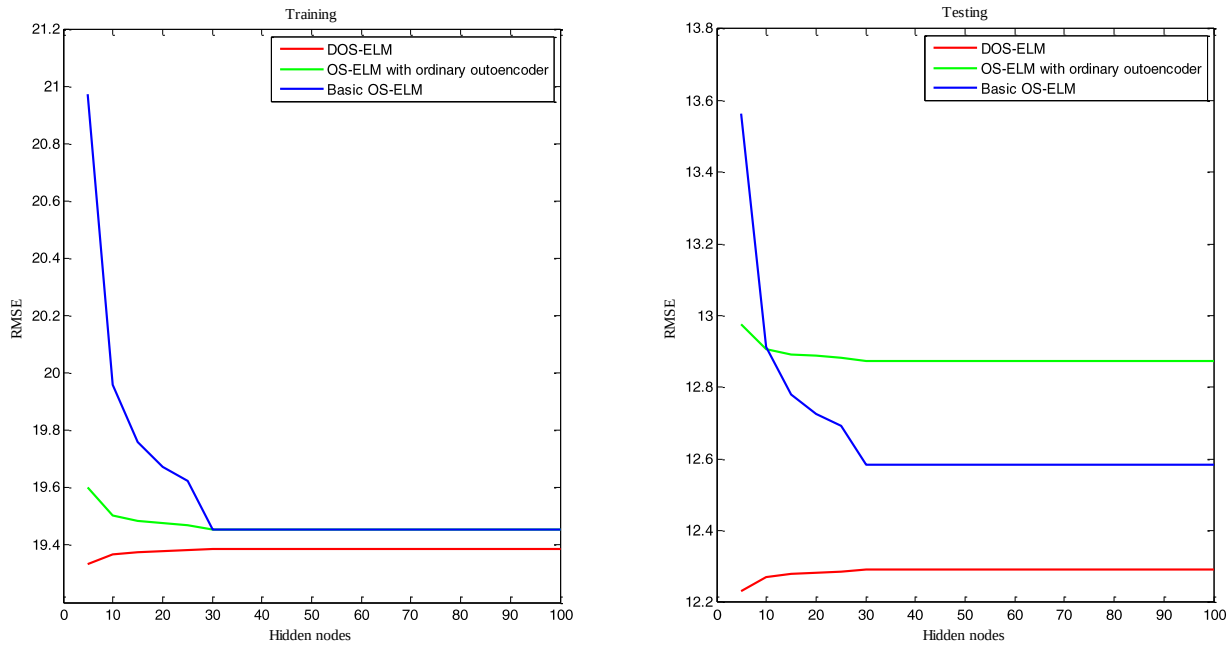

Fig. 9. Training and testing RMSE during incremental learning.

The results of Fig. 9 confirm that the adopted denoising process forces the hidden layers of the neural network to extract robust characteristics allowing more meaningful representations while leading to a more precise estimation.

In Fig. 9, and by showing that OS-ELM with an ordinary autoencoder outperforms basic OS-ELM during training and testing when hidden nodes are less than 10, this proves that learning with autoencoders requires fewer hidden nodes and soon it becomes more structural risk due to noisy data. The new denoising system shows more resistance against structural risk and allows maximum precision with only a few hidden nodes, which represents less calculation costs.

6. Conclusion

A new data-driven approach has been presented in this work. The adopted learning rules are based on OS-ELM one of ELM variants which are enhanced by both adaptive learning and a new denoising scheme. USS and DDFF are proposed to adapt the training weights towards data variation. Meanwhile, DAE is used to extract meaningful representations from driven patterns. Tests have been carried out with a single subset from the C-MAPPS turbofan engine dataset, where a single operating condition and a single fault mode were considered. One single type of noise has been used to train the denoising autoencoder and to study its effect on the data-driven prediction model. The results of the proposed approach explain that there is a certain level of noise that the training model can handle. Above this threshold value, the approximation function will start diverging from the desired response.

The USS explains that the data-driven model doesn't need all of the training mini-batches to achieve better prediction accuracy. In fact, a few training samples will be enough to satisfy both generalization and universal approximation. DDFF shows that it plays a key role in adapting the weights of DOS-ELM for the new coming time-varying mini-batches.

The algorithms have not been tested on other subsets of the dataset, for which more operating conditions and fault modes are available. Therefore, we encourage researchers to use different types of noises with the adopted C-MAPSS dataset to give a more general conclusion and to explain the limitation of the proposed

data-driven model. In addition to that, and since the dataset caries a multivariate time series which is recorded in sequences, it would be better if an integration of 'time memory 'is considered for adaptive learning.

References

[1] J. Zhu, N. Chen, and W. Peng, "Estimation of Bearing Remaining Useful Life Based on Multiscale Convolutional Neural Network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3208–3216, 2019.

[2] X. Wang and M. Han, "Improved extreme learning machine for multivariate time series online sequential prediction," *Eng. Appl. Artif. Intell.*, vol. 40, pp. 28–36, 2015.

[3] S. Xiang, Y. Qin, C. Zhu, Y. Wang, and H. Chen, "Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction," *Eng. Appl. Artif. Intell.*, vol. 91, no. October 2019, p. 103587, 2020.

[4] C. Ordóñez, F. Sánchez Lasheras, J. Roca-Pardiñas, and F. J. de C. Juez, "A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines," *J. Comput. Appl. Math.*, vol. 346, pp. 184–191, 2019.

[5] Giduthuri Sateesh Babu, "Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9642, pp. 214–228, 2016.

[6] Y. Li, S. Zhang, Y. Yin, W. Xiao, and J. Zhang, "A novel online sequential extreme learning machine for gas utilization ratio prediction in blast furnaces," *Sensors (Switzerland)*, vol. 17, no. 8, 2017.

[7] C. S. Chin and X. Ji, "Adaptive online sequential extreme learning machine for frequency-dependent noise data on offshore oil rig," *Eng. Appl. Artif. Intell.*, vol. 74, no. April, pp. 226–241, 2018.

[8] B. Yin, M. Dridi, and A. El Moudni, "Recursive least-squares temporal difference learning for adaptive traffic signal control at intersection," *Neural Comput. Appl.*, vol. 31, pp. 1013–1028, 2019.

[9] T. Matias, F. Souza, R. Araújo, N. Gonçalves, and J. P. Barreto, "On-line sequential extreme learning machine based on recursive partial least squares," *J. Process Control*, vol. 27, pp. 15–21, 2015.

[10] O. Bektas, J. A. Jones, S. Sankararaman, I. Roychoudhury, and K. Goebel, "A neural network filtering approach for similarity-based remaining useful life estimation," *Int. J. Adv. Manuf. Technol.*, vol. 101, no. 1–4, pp. 87–103, 2019.

[11] J.-M. Bai, G.-S. Zhao, and H.-J. Rong, "Novel direct remaining useful life estimation of aero-engines with randomly assigned hidden nodes," *Neural Comput. Appl.*, vol. 1, 2019.

[12] A. Saxena, M. Ieee, K. Goebel, D. Simon, and N. Eklund, "Damage Propagation Modeling for Aircraft Engine Prognostics," in *INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT*, 2008.

[13] F. Lu, J. Wu, J. Huang, and X. Qiu, "Aircraft engine degradation prognostics based on logistic regression and novel OS-ELM algorithm," *Aerosp. Sci. Technol.*, vol. 84, no. September, pp. 661–671, 2019.

[14] M. A. Djeziri, S. Benmoussa, and M. E. Benbouzid, "Data-driven approach augmented in simulation for robust fault prognosis," *Eng. Appl. Artif. Intell.*, vol. 86, no. August 2018, pp. 154–164, 2019.

[15] J. Ma, H. Su, W. L. Zhao, and B. Liu, "Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning," *Complexity*, vol. 2018, 2018.

[16] M. Xia, T. Li, T. Shu, J. Wan, C. W. De Silva, and Z. Wang, "A Two-Stage Approach for the Remaining Useful Life Prediction of Bearings Using Deep Neural Networks," *IEEE Trans. Ind. Informatics*, vol. 15, no. 6, pp. 3703–3711, 2019.

[17] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliab. Eng. Syst. Saf.*, vol. 172, pp. 1–11, 2018.

[18] G. Huang, N. Liang, H. Rong, P. Saratchandran, and N. Sundararajan, "On-Line Sequential Extreme Learning

Machine Review of Extreme Learning Ma- chine (ELM),” in *INternational Conference on Computational Intelligence*, 2005, no. Ci.

[19] N. Zhang, S. Ding, and Z. Shi, “Denoising Laplacian multi-layer extreme learning machine,” *Neurocomputing*, vol. 171, pp. 1066–1074, 2016.

[20] X. Cheng, H. Liu, X. Xu, and F. Sun, “Denoising deep extreme learning machine for sparse representation,” *Memetic Comput.*, vol. 9, no. 3, pp. 199–212, 2017.

[21] F. O. Heimes, “Recurrent Neural Networks for Remaining Useful Life Estimation,” in *International Conference on Prognostics and Health Management*, 2008.

[22] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. C. Soh, “Stacked Extreme Learning Machines.,” *IEEE Trans. Cybern.*, vol. PP, no. 99, p. 1, 2014.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion,” *J. Mach. Learn. Res.*, vol. 11, no. 3, pp. 3371–3408, 2010.

[24] P. Vincent and H. Larochelle, “Extracting and Composing Robust Features with Denoising Autoencoders,” 2008.

[25] B. Tarek, H. Mouss, O. Kadri, L. Saïdi, and M. Benbouzid, “Aircraft Engines Remaining Useful Life Prediction using an Improved Online Sequential Extreme Learning Machine,” *Appl. Sci.*, 2020.

[26] “DASHlink - Prognostics Data Challenge.” [Online]. Available: https://c3.nasa.gov/dashlink/projects/15/. [Accessed: 07-Oct-2020].

[27] W. Guo, T. Xu, K. Tang, J. Yu, and S. Chen, “Online Sequential Extreme Learning Machine with Generalized Regularization and Adaptive Forgetting Factor for Time-Varying System Prediction,” *Math. Probl. Eng.*, vol. 2018, 2018.

[28] T. Berghout, L. Mouss, O. Kadri, L. Saïdi, and B. Mohamed, “Aircraft Engines Remaining Useful Life Prediction with an Improved Online Sequential Extreme Learning Machine,” *Appl. Sci.*, vol. 10, no. 3, 2020.

[29] G. Bin Huang, “What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt’s Dream and John von Neumann’s Puzzle,” *Cognit. Comput.*, vol. 7, no. 3, pp. 263–278, 2015.