



**HAL**  
open science

# A deep supervised learning approach for condition-based maintenance of naval propulsion systems

Tarek Berghout, Leïla-Hayet Mouss, Toufik Bentrchia, Elhoussin Elbouchikhi,  
Mohamed Benbouzid

## ► To cite this version:

Tarek Berghout, Leïla-Hayet Mouss, Toufik Bentrchia, Elhoussin Elbouchikhi, Mohamed Benbouzid.  
A deep supervised learning approach for condition-based maintenance of naval propulsion systems.  
Ocean Engineering, 2021, 221, pp.108525. 10.1016/j.oceaneng.2020.108525 . hal-03412732

**HAL Id: hal-03412732**

**<https://hal.science/hal-03412732>**

Submitted on 2 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# A Deep Supervised Learning Approach for Condition-based Maintenance of Naval Propulsion Systems

Tarek Berghout<sup>1</sup>, Leïla-Hayet Mouss<sup>1</sup>, Toufik Bentrchia<sup>1</sup>, Elhoussin Elbouchikhi<sup>2</sup>, and Mohamed Benbouzid<sup>3,4</sup>

<sup>1</sup>University of Batna 2, Laboratory of Automation and Manufacturing Engineering, 05000 Batna, Algeria

<sup>2</sup>ISEN Yncréa Ouest, L@bISEN, 29200 Brest, France

<sup>3</sup>University of Brest, UMR CNRS 6027 IRDL, 29238 Brest, France

<sup>4</sup>Shanghai Maritime University, 201306 Shanghai, China

t.berghout@univ-batna2.dz, h.mouss@univ-batna2.dz, elhoussin.elbouchikhi@isen-ouest.yncrea.fr, Mohamed.Benbouzid@univ-brest.fr

**Abstract:** In the last years, predictive maintenance has gained a central position in condition-based maintenance tasks planning. Machine learning approaches have been very successful in simplifying the construction of prognostic models for health assessment based on available historical labeled data issued from similar systems or specific physical models. However, if the collected samples suffer from lack of labels (small labeled dataset or not enough samples), the process of generalization of the learning model on the dataset as well as on the newly arrived samples (application) can be very difficult. In an attempt to overcome such drawbacks, a new deep supervised learning approach is introduced in this paper. The proposed approach aims at extracting and learning important patterns even from a small amount of data in order to produce more general health estimator. The algorithm is trained online based on local receptive field theories of extreme learning machines using data issued from a propulsion system simulator. Compared to extreme learning machine variants, the new algorithm shows a higher level of accuracy in terms of approximation and generalization under several training paradigms.

**Keywords:** Predictive maintenance; Decay detection; Extreme learning machine; Deep learning; Prognostic and health management; Naval propulsion systems.

## Nomenclature

**ANN:** artificial neural network

**C-DBN:** combinatorial DBN

**CNN:** convolutional neural network

**CODLAG:** COmbined Diesel eLectric And Gas propulsion plant

**DBN:** deep belief neural network

**ELM:** extreme learning machine

**ELM-LRF:** ELM with local receptive fields

**GT:** gas turbine

**GTC:** gas turbine compressor

**HP:** high-pressure turbine

**LP:** low-pressure turbine

**ML:** machine learning

**OS-ELM:** online sequential ELM

**RLS:** recursive least squares

**RMSE:** root means squared error

**RNN:** recurrent neural network

**SLFN:** single hidden layer feedforward neural network

**SVM:** support vector machine

**C:** regularization parameter

**H:** hidden layer

**I:** identity matrix

**M:** covariance matrix

**N:** number of neurons

**T:** targets

**$x_i$ :** single input observation

**X:** inputs

**a:** input weights

**b:** biases

**$d_{GT}$ :** GT decay coefficients

**$d_{GTC}$ :** GTC decay coefficients

**e:** estimation error

**k:** index of the training mini-batches

**l:** error rate

**L:** number of hidden layers

**n:** number of samples

**S:** mini-batch size

**$\beta$ :** output weights

**$\lambda$ :** forgetting factor

## 1. Introduction

Predictive maintenance can be considered as a concept strongly related to the computer-assisted maintenance management field. It consists in anticipating maintenance requirements in order to prevent alteration or secession of such a system earlier by following damages propagation [1], [2]. This prediction approach needs an accurate prognostic system through which failure conditions can be analyzed to directly address associated causes[3], [4]. Predictive maintenance makes it possible to predict technical incidents rather than suffer them [5]. Thanks to this strategy, it is thus possible to plan interventions in advance and thus avoid an unexpected immobilization of equipment or a machine. It therefore pushes things a step further than preventive maintenance.

Many modeling approaches can be involved to determine the operating behavior such as physical, numerical, and mathematical modeling [5]. However, the complexity and flexibility of recent systems, which are due to the massive evolution in industrial technologies, have resulted in difficult manipulation besides the lack of precision at the level of the aforementioned modeling tools. Hence, these shortcomings may lead to reduced reliability for health monitoring systems [6].

Motivated by both the remarkable advances in sensor technologies and the low implementation cost, Machine Learning (ML) tools are becoming more relevant for the generation of more reliable training models in comparison to conventional strategies [4], [7], [8].

In the literature, several ML tools with different training paradigms and architectures have been investigated. These studies are involved in the analysis of the faults of many industrial systems, in particular the most critical ones such as propulsion systems. In fact, a small defect or failure in any of the sub-components of these systems, can lead to probable catastrophes.

For instance, ensemble algorithms[9], deep learning [7], [10], [11] and hybrid architectures [12]–[14] were studied as part of the analysis of deterioration and propagation of damage for different operating systems including propulsion systems associated to marine vessels. In this regard, Coraddu *et al.* [5] constructed the numerical model of a COmBined Diesel eLectric And Gas propulsion plant (CODLAG) to generate data corresponding to fatigue growth of the propulsion system. Two different ML tools, namely Support vector machine (SVM) and recursive least squares (RLS) have been used to model data behavior under several sub-sampling methods jointly with different data fractioning paradigms. In this study, it has been proven that SVM was able to achieve more accurate results. Cipollini *et al.* [3] performed a comparative study to investigate the use of several supervised learning methods to determine the most suitable model for decay detection using the same training data of the previous work. Moreover, the use of hybrid models that combines two paths for both unsupervised and supervised learning, with the aim of providing more reliable learners under the lack of data have been presented by Cipollini *et al.* [1]. Tan *et al.* [2] used one-class SVM-based approach for a dataset classification attempting to keep the prediction accuracy as lower as possible even under the use of a small amount of samples during the training process. Furthermore, Tan *et al.* [6] proposed to study the same topic using a new ML approach based on the combination of a hybrid SVM and an isolation forest.

In the context of ML modeling, the availability of labeled training data allows obtaining well-generalized universal approximators. Nevertheless, these universal approximators suffer from insufficient precision when the recorded data representing the history of the operating systems suffer for the lack of labels, according to the theories of deep learning [15]. In addition, a multitude of time varying data collected from various and varied sensors that change over time and distribution makes dynamic adaptation really important to maintain the stability of training models [16].

The emergence of deep learning algorithms in ML has gained an excellent reputation as it can handle both generalization and approximation based on an unsupervised training process. Moreover, a fine tuning can be performed based on a small set of training samples [17], [10], [18], [19]. However, supervised training or fine tuning of deep networks based gradient descent algorithms such as backpropagation or contrastive divergence, or unsupervised learning approaches, could make the training process more difficult and not precise enough because of the following issues[20]–[24]:

- These training algorithms can easily be trapped within local minima and suffer from stopping criteria;
- Numerous components including iterative tuning, repeating learning sequences for Recurrent Neural Networks (RNN), multiple layers of convolutional mapping and pooling for Convolutional Neural Networks (CNNs), forward and backward propagation for autoencoders and restricted Boltzmann machines entail a huge amount of computational costs and training time, especially under big data environments;
- The learning rate, the dropout, regularization based L1 and L2 norms minimization, the overlapping step, the size of the local receptive field and many other hyperparameters should be adjusted in each iterative settings. This issue does not address the biological learning of the human brain and leads to more human intervention or some other combined random search algorithms;
- Data from the actual process are required to be divided by experts into training and test sets, as familiar in well-known benchmarks. Otherwise, another problem of randomized validation or k-fold cross-validation investigations should be achieved attempting to find the most suitable training samples avoiding the empirical risks or other ill-posed problems.

The training tool known as extreme learning machine (ELM) has brought many enhancements to the realm of artificial neural networks (ANN) [25]. The ELM has been proposed in an attempt to eliminate the barriers between human thinking and the ANN by discarding the employment of both iterative adjustment of the hidden nodes parameters during training besides the learning rate [26]. As a matter of fact, ELM aims to design a well generalized universal approximators by considering only the hyperparameters of the output layer [27].The ELM was first unleashed for the training of single hidden layer feedforward neural networks (SLFN) [26].Then, it spreads so as to adapt to all the paradigms of ANNs for different domains such as the current health assessment and predictive maintenance field [28], ranging from multilayer to deep and hierarchical architectures under both online or offline learning attitudes [25]. In addition, ELM provides easy integration of adaptive learning which deals with environmental variables as well as fast robust extractors of features such as denoising autoencoders, like those that have been investigated in previous work in [29].

In this work, the main objective is the design of an accurate learning and prediction tool based on ELM theories for the analysis of the damages propagation in naval propulsion plants. The main contributions of the present work as regarding the naval propulsion systems maintenance scheduling are as follows:

- The use of ELM theories to train a deep belief neural network (DBN) for both unsupervised learning and supervised fine tuning stages. This allows taking advantage of both deep architecture of the DBN precision under the lack of labeled data and the reduction in computational time;
- The integration of locally connected combinatorial neural sub-networks into the hidden nodes based on ELM local receptive fields theories (ELM-LRF). Hence, the hidden layers take advantage of the convolutional mapping of CNNs [30], [31];
- The introduction of a regularized online learning-based on online sequential ELM (OS-ELM) to address adaptive training, which prevents from structural risks [32].

The novelty of the proposed algorithms lies in the combination of all these characteristics in a very fast adaptive learning with both a convolutional mapping and a deep reconstruction of features with a stack of autoencoders in a single framework of a multilayer neural network.

The proposed combinatorial deep belief neural network (C-DBN) is evaluated as part of the prediction of the decay of the frigate using the same data taken from a simulation model of CODLAG [5]. The results of the evaluation are compared with basic variants of the proposed algorithm namely OS-ELM and ELM. The comparative study proves the effectiveness of the proposed approach over the ELM and OS-ELM learning rules.

The rest of the paper is organized as follows: Section 2 describes the materials used in this work. Section 3 presents the proposed methodology as well as the adopted algorithms. Section 4 is dedicated to experimental results analysis and discussions. Finally, section 5 concludes this paper and gives some directions for future works in this field.

## 2. System and dataset description

CODLAG system is a hybrid naval propulsion system with a double shaft arrangement that employs electric motors powered by diesel engines as illustrated by **Figure (1-a)**. The gas turbine (GT) depicted by the diagram of **Figure (1-b)** powers the shafts via an interconnected gearbox as an alternative drive source [6]. The GT consists of a single compressor (C) and a high pressure turbine (HP) designed for the production of exhaust gas in which it drives the low pressure (LP) turbine for the production of enough thrust power to move the vessel [5]. The numerical model of the engine has been previously simulated by Coraddu *et al.* [5] using Matlab/Simulink library and the recovered data were released for the public at the data repository of the University of California at Irvine [33]. More description about the modeling process and the vessel propulsion system can be found in [5].

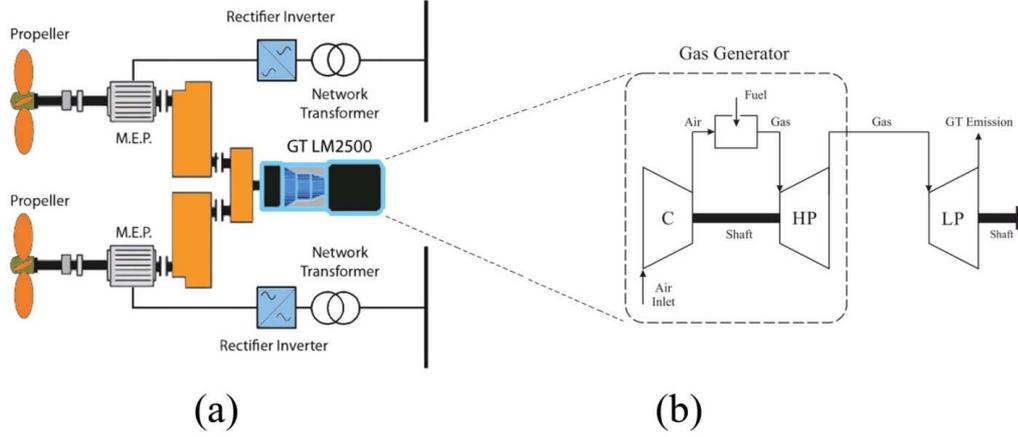


Figure 1. (a) CODLAG propulsion system, (b) diagram of the gas turbine [5].

The dataset describes both GTC and GT decay variation ( $d_{GTC}$ ,  $d_{GT}$ ) as defined by equations (1) and (2) with 16 different sensors measurements gathered in 11934 samples. All parameters, which are organized in a dataset are obtained from [19] and their computed statistical characteristics are presented in Table 1. Collected samples are recorded sequentially with a higher level of dynamic disturbance, which may drive any approximation function towards structural risks if the loss function is not optimally minimized without considering regularization.

$$d_{GTC} = 1 - i \cdot 0.001, i \in [0,1,2, \dots, 51]. \quad (1)$$

$$d_{GT} = 1 - i \cdot 0.001, i \in [0,1,2, \dots, 26]. \quad (2)$$

Table 1. Dataset features and their characteristics.

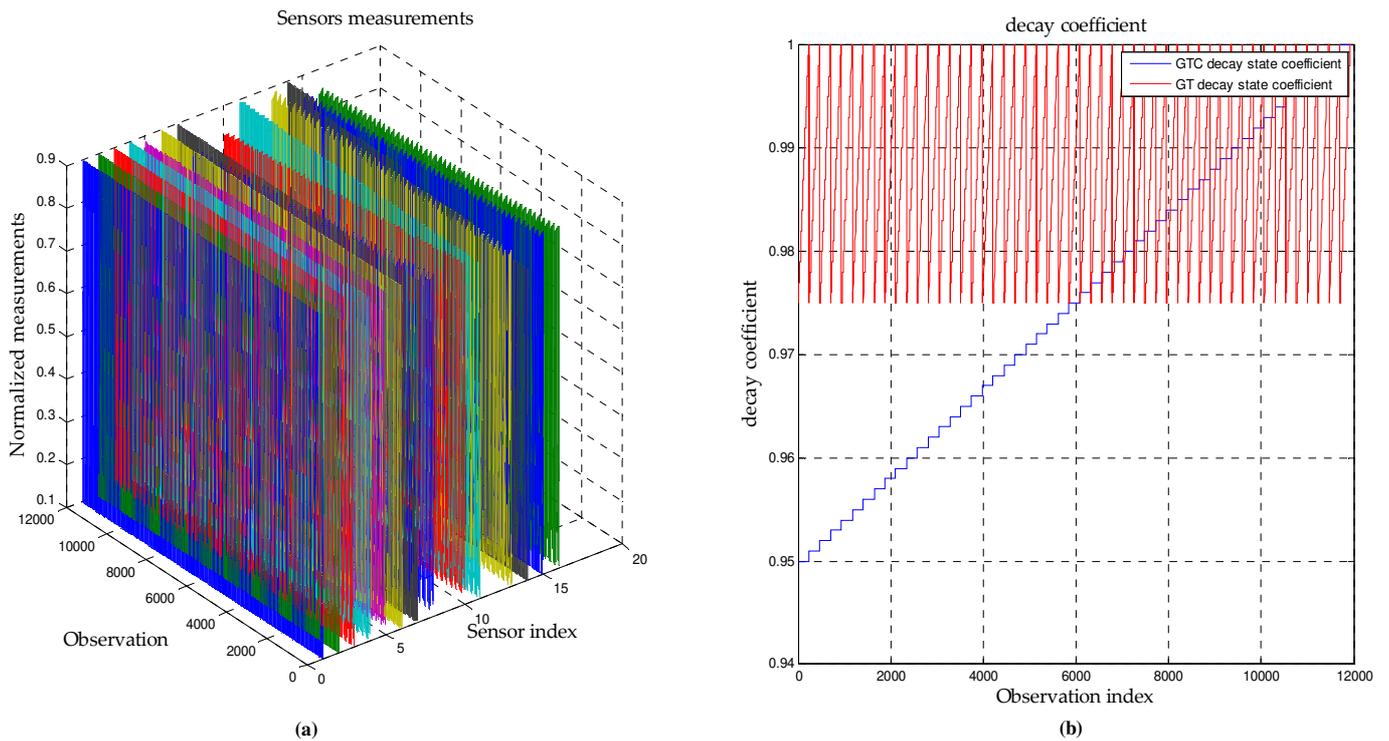
Index	Features	Unit	Min( $10^3$ )	Max ( $10^4$ )	Mean ( $10^4$ )	Standard deviation ( $10^4$ )
1	Lever position (lp)	[ ]	0.0011	0.0009	0.0005	0.0003
2	Ship speed (v)	[knots]	0.0030	0.0027	0.0015	0.0008
3	Gas Turbine shaft torque (GTT)	[kN m]	0.2535	7.2785	2.7247	2.2148
4	Gas Turbine rate of revolutions (GTn)	[rpm]	1.3077	0.3561	0.2136	0.0774
5	Gas Generator rate of revolutions (GGn)	[rpm]	6.5890	0.9797	0.8201	0.1091
6	Starboard Propeller Torque (Ts)	[kN]	0.0053	0.0645	0.0227	0.0200
7	Port Propeller Torque (Tp)	[kN]	0.0053	0.0645	0.0227	0.0200
8	HP Turbine exit temperature (T48)	[C]	0.4424	0.1116	0.0735	0.0174
9	GT Compressor inlet air temperature (T1)	[C]	0.2880	0.0288	0.0288	0
10	GT Compressor outlet air temperature (T2)	[C]	0.5404	0.0789	0.0646	0.0073
11	HP Turbine exit pressure (P48)	[bar]	0.0011	0.0005	0.0002	0.0001
12	GT Compressor inlet air pressure (P1)	[bar]	0.0010	0.0001	0.0001	0.0000
13	GT Compressor outlet air pressure (P2)	[bar]	0.0058	0.0023	0.0012	0.0005
14	Gas Turbine exhaust gas pressure (Pexh)	[bar]	0.0010	0.0001	0.0001	0.0000
15	Turbine Injection Control (TIC)	[%]	0	0.0093	0.0034	0.0026
16	Fuel flow (mf)	[kg/s]	0.0001	0.0002	0.0001	0.0001
17	GTC decay state coefficient ( $d_{GTC}$ ).	[ ]	0.0010	0.0001	0.0001	0.0000
18	GT Turbine decay state coefficient ( $d_{GT}$ ).	[ ]	0.0010	0.0001	0.0001	0.0000

According to [1], GTC and GT are not decaying if the decay coefficients of GTC and GT belong to the intervals  $[0.98,1]$  and  $[0.99,1]$ , respectively. Contrariwise, they can be considered as decaying if their decay coefficient belongs to  $[0.95,0.98]$  and  $[0.975,0.99]$ , respectively.

As indicated in [5], the decay variation for both GTC and GT is defined according to equations (1) and (2), respectively.

**Figure 2** illustrates the dataset issued from different sensors and the corresponding decay variations in both GTC and GT as delivered by the simulation model. It depicts the normalized data versus observation and sensors indices (**Table 1**). This figure shows the huge amount of the collected data and the difficulty to analyze such a database in order to get any information about the operating and health state of both sub-systems, namely GTC and GT.

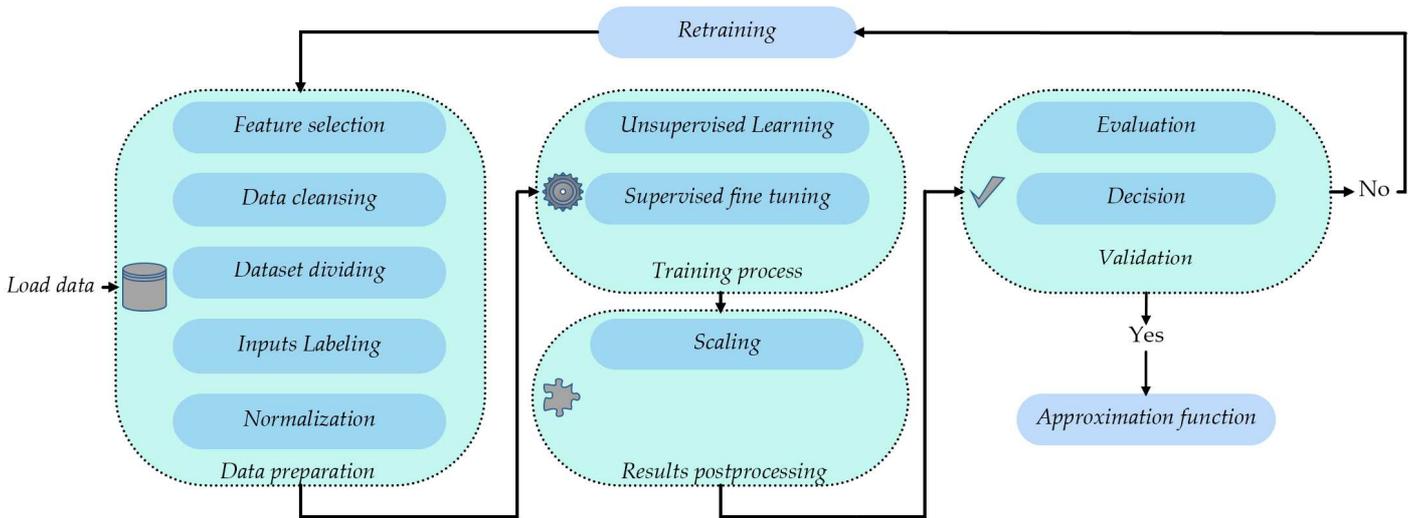
As it is known from deep learning theories, in such a big data problem, the well-designed deep learning model has more meaningful feature representations than ordinary feature mapping. Hence, the reduction of the complexity of data and the extraction of meaningful patterns was one of the main reasons that led to combine two well known deep learning techniques (convolutional mapping, feature reconstruction) at the same time within one algorithm with a very fast learning scheme of OS-ELM.



**Figure 2.** (a) Normalized measurements versus number of observations and sensors index (b) Variation of decay coefficient with respect to observation index.

### 3. Methodology

The flowchart in **Figure 3** depicts the proposed methodology for training model construction. It can be noticed from this figure that the training stage is composed of four distinctive parts: dataset preparing, model training, results post processing and model validation. These consecutive stages are further described in the following sub-sections.



**Figure 3. Flowchart of the proposed methodology.**

### 3.1. Dataset preparing

Raw data from the current dataset cannot be directly exploited by the prediction model for training or prediction purposes due to the incompatibility between the measurement scales of the sensors and their unknown degree of correlation. Therefore, simple standardization and some correlation investigations must be performed to ensure the accuracy and the effectiveness of the training process.

In the current study and according to the ELM learning constraints, the input learning samples  $x_i$  are scaled to fit within the range  $[0, 1]$  using the min-max normalization formula shown in equation (3). After that, the correlation matrix is used to decide which parameters are critical for the elaboration of the decay prediction model.

$$x = \frac{x_i - x_{min}}{x_{max} - x_{min}}. \quad (3)$$

The heat map in **Figure (4-a)** shows the correlation density between dataset features before and after selection. It shows, for example, that sensors T1 and P1, corresponding to the inlet air temperatures and inlet air pressure of the GT compressor, respectively, can probably lead the model to an overfitting or to other ill-posed problems due to the non-correlation between them under the same criteria of decay variation. Consequently, these measurements must be eliminated to increase the correlation of the data as it can be seen in the **Figure (4-b)**.

Labels of each observation are defined according to the equations (1) and (2). Regarding the current study, the dataset is divided randomly into training and testing sets according to different ratios.

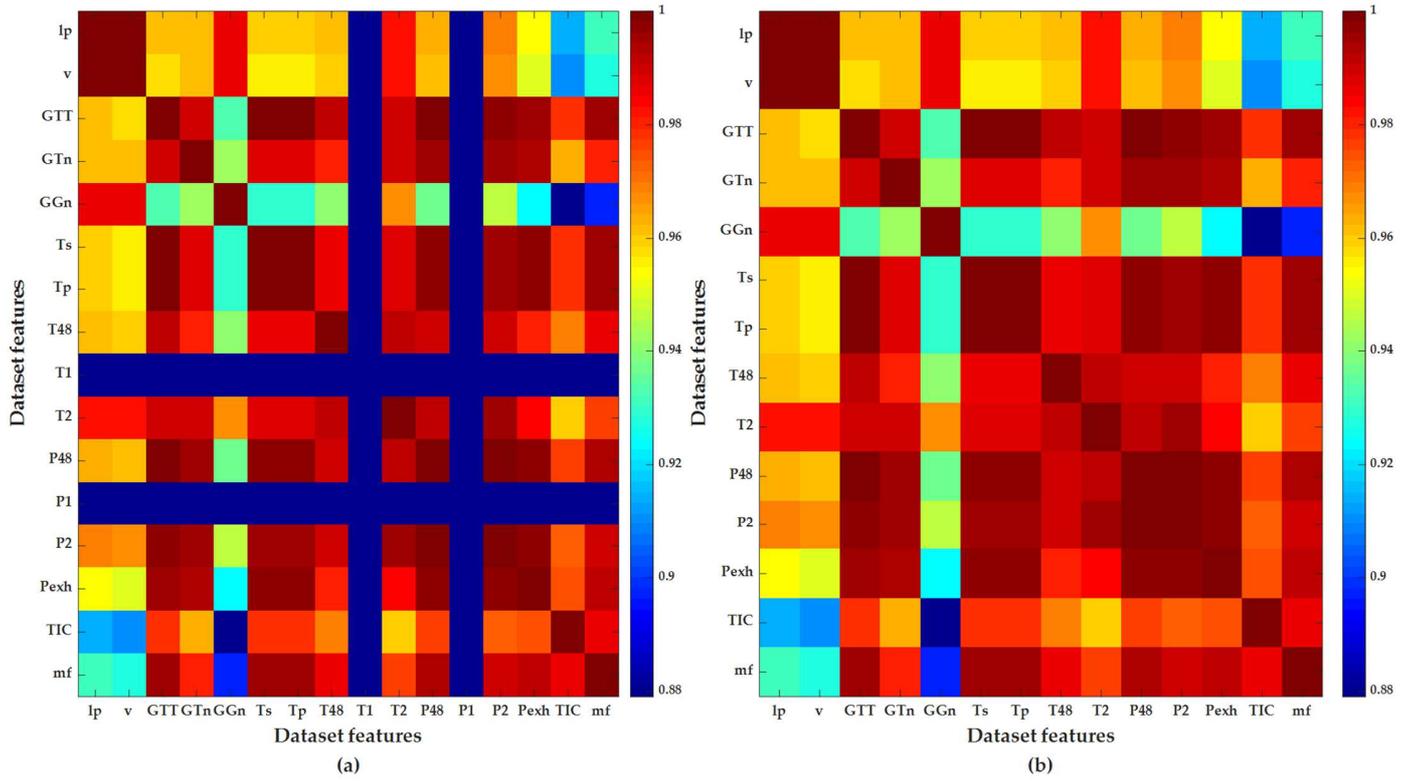


Figure 4. (a) Correlation density between dataset features before selection. (b) Correlation density between dataset features after selection.

### 3.2. Training algorithm

The new training rules of the proposed deep neural network are inspired from basic OS-ELM [32]. This is one of the ELM variants that have been developed attempting to address online learning as a feedback of environmental data accumulation of real world problems. OS-ELM with the single hidden layer architecture is presented in Figure 5. It can be trained for any dataset of  $n$  mini-batches  $\{X_k, T_k\}_{k=1}^n$  with the size of  $\{S_k\}_{k=1}^n$  based on ELM theories in two main phases, which are: the initialization phase and the recursive phase, where  $\{X_k, T_k\}$  represent the training inputs and outputs, respectively.

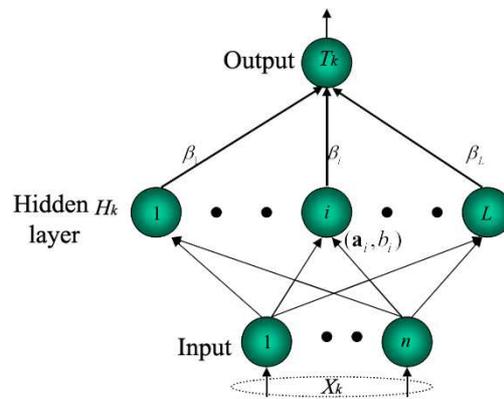


Figure 5. Single hidden layer based ELM notations [25].

In the initialization phase, hidden nodes parameters, namely weights and biases ( $a, b$ ), are randomly and independently generated from the training data and must be scaled within the range  $[-1, 1]$ . After that, the hidden layer  $H$  can be mapped and activated according to any chosen activation function ( $G$ ) as shown in equation (4).

$$H_k = G(aX_k + b). \quad (4)$$

The initial training weights  $\beta_1$  of the output layer can be analytically determined using the Moore-Penrose pseudo-inverse of the matrix as shown in equation (5), using the covariance matrix  $M_1$  in (6).

$$\beta_1 = M_1 H_1 T_1, \quad (5)$$

$$M_1 = (H_1^T H)^{-1}. \quad (6)$$

In the recursive phase, the output weights  $\beta$  are updated using formula (9) depending on the updated covariance matrix  $M_{k+1}$  and estimation error  $e_{k+1}$  in equations (7) and (8), where  $I$  denotes the identity matrix.

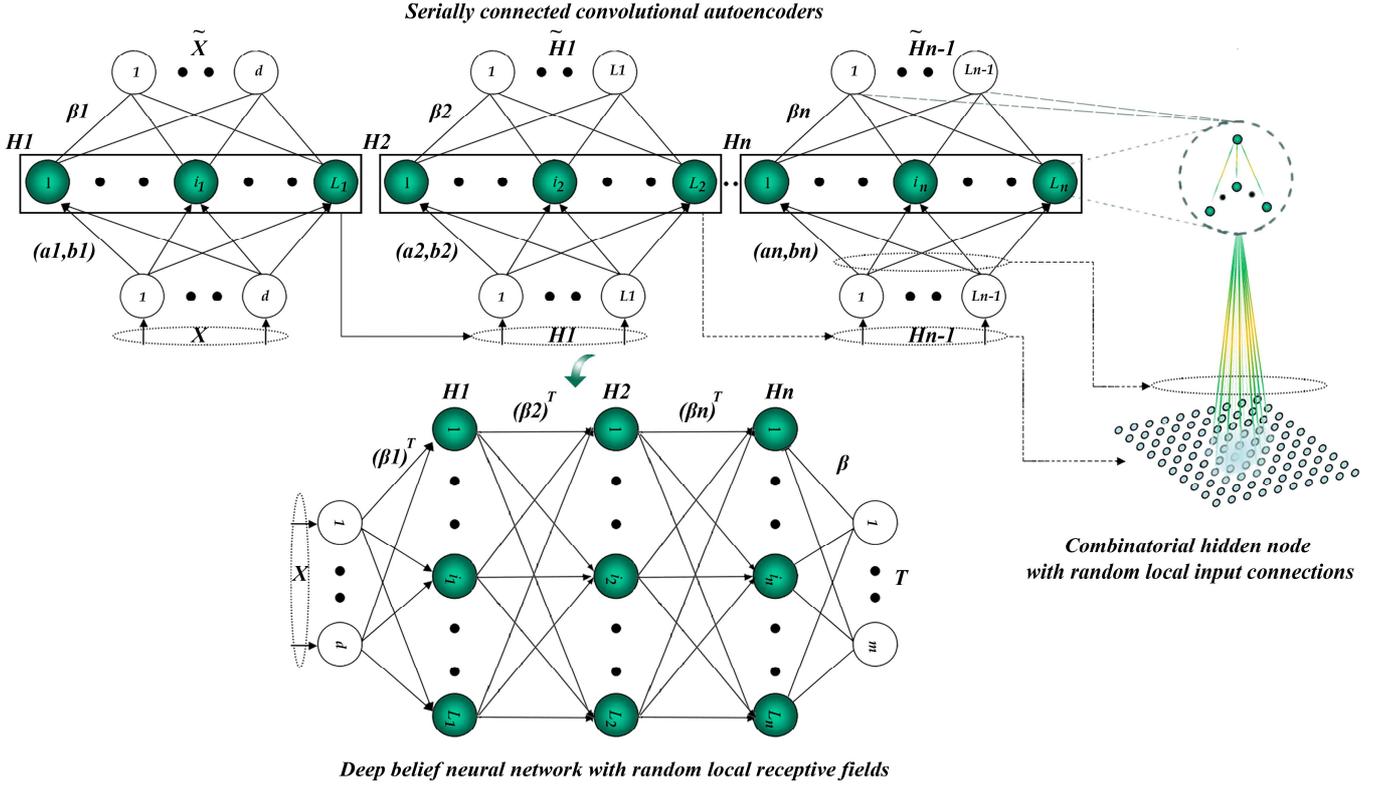
$$M_{k+1} = M_k - \frac{M_k H_{k+1} H_{k+1}^T M_k}{I + H_{k+1}^T M_k H_{k+1}}, \quad (7)$$

$$e_{k+1} = (T_{k+1} - H_{k+1}^T \beta_k), \quad (8)$$

$$\beta_{k+1} = \beta_k + M_{k+1} e_{k+1}. \quad (9)$$

The proposed algorithm given by **Figure 6** is a result of a combination of two deep learning models: the local receptive fields (LRF) and deep belief neural networks. The deep network is trained for unsupervised learning with  $L$  number of serially connected autoencoders and then fine-tuned for supervised learning using OS-ELM theories. The hidden nodes in each autoencoder could be locally connected to sub-networks that address both convolutional mapping and pooling in a simple mapping theories based on ELM-LRF [30].

Each sub-network of the hidden layer is either fully connected or locally connected. The decision regarding the types of hidden nodes is chosen randomly by following ELM theories of randomness in parameters selection. The locally connected hidden nodes are designed to analyze local fields of random size from each sequence of training data. Inside each combinatorial node, the resulting feature maps are followed by a pooling process involving a fully connected layer with a single output and random weights to fill in the gaps in full rank mapping and to satisfy the learning constraints of the least squares method.



**Figure 6. Architecture of the proposed combinatorial DBN algorithm.**

In deep belief neural networks, feature mapping resulted in unsupervised reconstruction of inputs and the next stack of hidden layers, can help the multilayer neural network to get the best approximation during the fine tuning process [21]. Therefore, in the designed network, the feature mapping of the training inputs, which is resulted by convolutional filtering of each autoencoder, will be the input of the next autoencoder. The transpose matrix of the tuned weights will be used as the encoding key for the new coming samples. After the unsupervised learning, tuned weights of the encoders will be used to the multilayer neural network for supervised learning. Thus, a process of fine-tuning at the last output layer is necessary. The convolutional filtering under deep architecture will allow scanning multiple possibilities of representations of the features space, which will allow the neural network to be aware of any variation of data and to guarantee its adequate approximation.

According to regularized ELM (R-ELM) theories [34] and forgetting mechanism of OS-ELM [35], regularization and forgetting parameters  $\{C, \lambda\}$  can be added to reduce the amount of structural risk of OS-ELM training as well as to adapt the training weights towards data variations. As a result, equations (5) and (7) must be changed into (10) and (11) to fit the new training process. Besides, the target in equations (5) and (8) needs to be equal to the input in order to satisfy the unsupervised training process.

$$\beta_1 = \left( \frac{I}{C} + H_1^T H_1 \right) H_1^T T_1, \quad (10)$$

$$M_{k+1} = \frac{M_k}{\lambda} - \frac{M_k H_{k+1} H_{k+1}^T M_k}{\lambda (I + H_{k+1}^T M_k H_{k+1})}. \quad (11)$$

Where  $\lambda \in [0, 1]$  and  $C \in [0, +\infty)$ .

According to ELM learning theories of autoencoders as in[36],there is no need after the initial phase to use the random feature mapping with local receptive fields. Instead of that, the transpose of the determined output weights will be used for the mapping process as shown in equation (12).

$$a = \begin{cases} a, k = 1 \\ \beta_{k+1}^T, k > 1 \end{cases} \quad (12)$$

The proposed C-DBN approach is summarized in **Algorithm 1**.

<b>Algorithm1. C-DBN-based OS-ELM</b>	
Inputs: $X_k, T_k, S_k, C, \lambda, L, n, N$ .	
Outputs: Fully trained C-DBN.	
Initial phase	<ol style="list-style-type: none"> <li>1 <b>For</b> <math>j = 1: L</math></li> <li>2 Load initial mini-batches;</li> <li>3 <b>If</b> <math>j = 1</math></li> <li>4 <math>X_k = X_k</math>;</li> <li>5 <b>Else</b></li> <li>6 <math>X_k = H_k</math>;</li> <li>8 <b>End</b></li> <li>9 Generate input weights and biases (a, b);</li> <li>10 Map the inputs to the sub-networks via randomly connected receptive fields;</li> <li>11 Activate the feature mapping as an initial hidden layer using equation (4);</li> <li>12 Calculate the initial covariance matrix based on (6);</li> <li>13 Determine the initial output weights of the <b>autoencoders</b> using (10) by putting the target same as the input ;</li> <li>14 <b>If</b> <math>j = L</math></li> <li>15 Tune initial output weights of the <b>supervised</b> learning using equation (10);</li> <li>16 <b>End</b></li> </ol>
Sequential learning	<ol style="list-style-type: none"> <li>17 <b>For</b> <math>k = 2: n</math></li> <li>18 Load new mini-batches;</li> <li>19 <b>If</b> <math>j = 1</math></li> <li>20 <math>X_k = X_k</math>;</li> <li>21 <b>Else</b></li> <li>22 <math>X_k = H_k</math>;</li> <li>23 <b>End</b></li> <li>24 Calculate the hidden layer by <b>replacing</b> the input weights as mentioned in equation (12);</li> <li>25 Update the output weights of the <b>autoencoders</b> using (8), (9) and (11) by putting the target same as the input ;</li> <li>26 <b>If</b> <math>j = L</math></li> <li>27 Update the output weights for <b>supervised</b> learning, equation (8), (9) and (11);</li> <li>28 <b>End</b></li> <li>29 <b>End</b></li> <li>30 <b>End</b></li> </ol>

### 3.3. Model validation

The training algorithms in this case are evaluated using the error rate formula presented in equation (13), which is proposed by Coraddu *et al.* [5] for studying this benchmark.

$$l = \frac{1}{n} \sum_{i=1}^n \left| \frac{\tilde{T}_i - T_i}{T_i} \right|. \quad (13)$$

Where,  $n$  denotes the number of training or testing samples,  $\tilde{T}$  represents the estimated target,  $T$  describes the desired target and  $l$  refers to the error metric function.

#### 4. Experiments results and discussions

In current experiments, two main targets are considered, which are GTC and GT decay coefficients. Therefore, two training paradigms have been studied in the case of a single output and multiple outputs for trained neural networks. In each experiment, we use different manually tuned hyperparameters based on a simple grid search mechanism to enhance the accuracy of results. Experimental dataset processing has been performed by running the proposed algorithms in a personal computer with an Intel (R) Core (TM) i5-3427U CPU @ 1.80 GHz 2.80 GHz with 4.00 GB of RAM.

Since ELM depends on Moore Penrose pseudo-inverse method, which in general suffers from the interpolation problem caused by the least squares method [37], the best suggestion of results adjustment is rescaling according to initial condition from the targets values. In the current experiment, we have adjusted the targets using min-max normalization depending on the original values of the decay coefficients, to fit into a range of meanings and ensure that their statistical inferences make sense relatively to the original ones.

**Tables 2** and **3** give the results of the comparative study by considering all of the three cases of predictions (GTC, GT, GTC & GT). The training ratio was tuned according to [5] in order to avoid complicating validation problem and to keep the originality of the prediction problem. In the proposed approach of Coraddu *et al. in* [5], the authors used different sub-sampling approaches (uniform distribution, bimodal distribution and monomodal distribution) with different prediction cases (single output and multiple outputs). Different training sets of different sizes were made in each case. In our training scheme, we followed a simple random sampling process because it is easier to implement and leads to less algorithmic complexity. However, the dataset partitioning is defined the same way by taking only the largest training set in each case, which is shown in Tables 2 and 3.

**Table 2. Comparison results in case of random sampling (training ratio = 0.98, approximately 200 testing samples).**

<b>Decay estimation of GTC in case of multiple outputs,</b>				
	Training time(s)	Testing time(s)	Training error	Testing error
<b>C-DBN</b>	<b>2.2932</b>	<b>0.0312</b>	<b>0.0016</b>	<b>5.4639×10<sup>-04</sup></b>
<b>OS-ELM</b>	2.5584	0.0312	0.0143	0.0136
<b>ELM</b>	3.1044	0.0624	0.0138	0.0132
<b>Decay estimation of GT in case of multiple outputs, training ratio: 0.98</b>				
	Training time(s)	Testing time(s)	Training error	Testing error
<b>C-DBN</b>	<b>2.2932</b>	<b>0.0312</b>	<b>0.0022</b>	<b>4.8814×10<sup>-04</sup></b>
<b>OS-ELM</b>	2.5584	0.0312	0.0070	0.0067
<b>ELM</b>	3.1044	0.0624	0.0068	0.0066

**Table 3. Comparison results in case of random sampling (training ratio = 0.95, approximately 500 testing samples).**

<b>Decay estimation of GTC in case of multi-outputs</b>				
	Training time(s)	Testing time(s)	Training error	Testing error
<b>C-DBN</b>	2.1060	0.0312	1.3279×10 <sup>-04</sup>	2.8132×10 <sup>-05</sup>
<b>OS-ELM</b>	0.8736	0.0312	8.5208×10 <sup>-05</sup>	7.4560×10 <sup>-05</sup>
<b>ELM</b>	1.1388	0	5.1762×10 <sup>-05</sup>	1.7372×10 <sup>-05</sup>
<b>Decay estimation of GT in case of multi- outputs, training ratio: 0.95</b>				
	Training time(s)	Testing time(s)	Training error	Testing error
<b>C-DBN</b>	0.9204	<b>0.0312</b>	<b>4.3501×10<sup>-04</sup></b>	<b>2.0366×10<sup>-04</sup></b>
<b>OS-ELM</b>	0.9048	0.0312	0.0104	0.0066
<b>ELM</b>	2.0748	0.0312	0.0073	0.0090

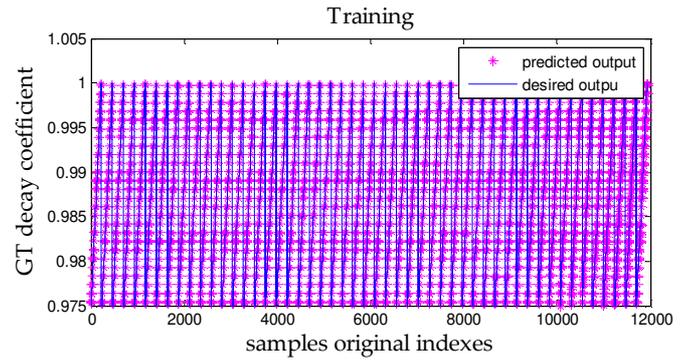
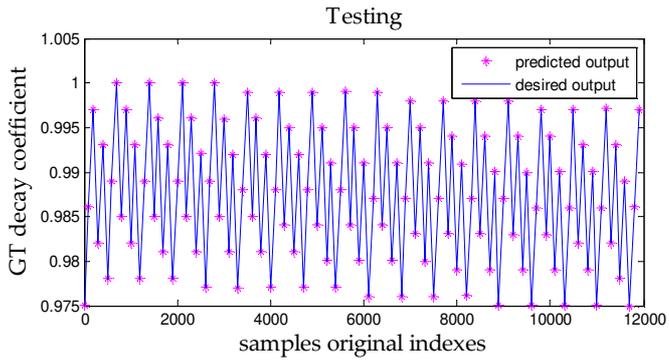
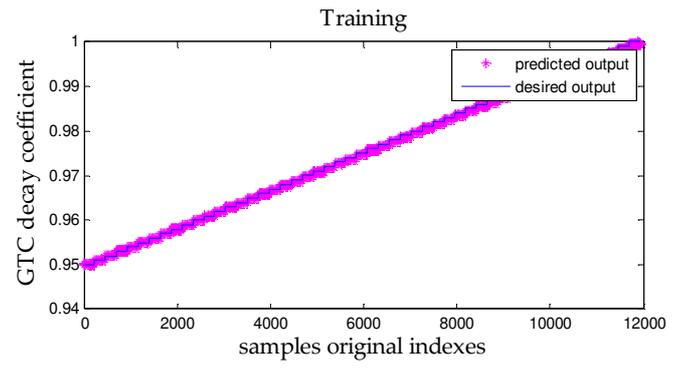
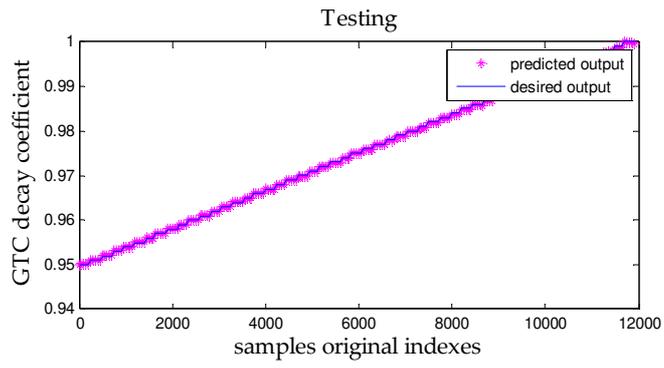
On one hand, it can be seen from **Tables 2 and 3** that the proposed algorithm outperforms offline training paradigms. Improvements in the prediction accuracy of the online learning attitude are the result of the dynamic adaptation with a forgetting mechanism. Unlike the basic ELM which is based on the offline least squares method, the sequence learning forgetting factor of OS-ELM acts as a temporal memory that allows the learning weights to decide which samples are much important to remember. In addition, the setting of new multi-feature maps based on the feature space resulting from both random convolutional filtering and pooling techniques which are integrated inside the hidden units as sub-networks, can push the hidden layers towards robust feature extraction and allow expressing more meaningful representations. Moreover, These tables also shows that ELM variants can handle single output better than multiple outputs under the current decay prediction problem. Besides, even in the current deep complex architecture of the algorithm, the computational time during training stage is less than 3 seconds. This is almost a negligible amount of time compared to the computational burden of the ordinary gradient descent DBNs and other old deep learning architectures. The current learning speed is the result of simplified learning rules driven by both non-iterative adjustments and fewer hyperparameters that allow less human intervention. On the other hand, by sub-sampling the training and test sets randomly from the original dataset with the proviso that the test samples encompass samples of sequences similar to the training set, the results of curve fitting can be viewed as an indicator of performances that visually explains the magnitude of variation in predicted samples.

**Figures 7 and 8** illustrate the proposed network behavior in case of a single target. Results of the curve fitting in training and testing are much closer to the original samples than those in **Figure 8**, which slightly deviated. This therefore leads to the same conclusion that recursive least squares methods can handle a single output better than multi-outputs.

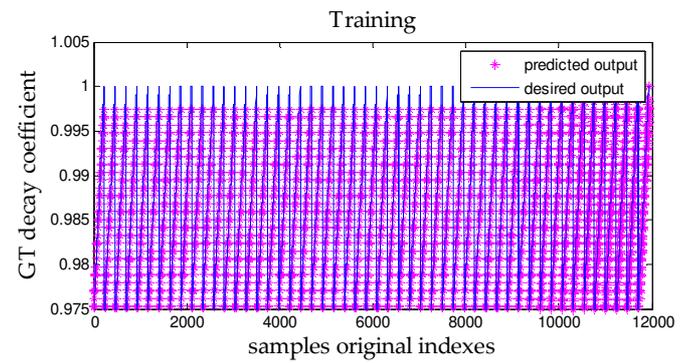
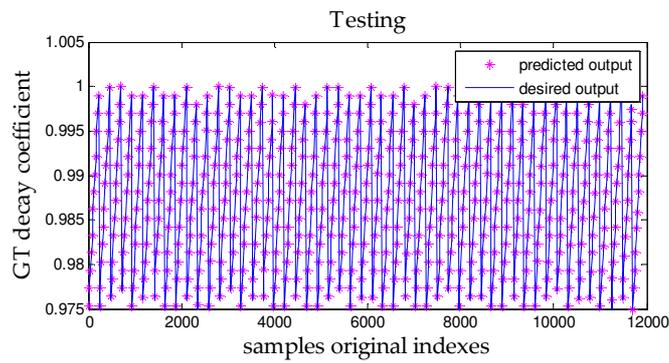
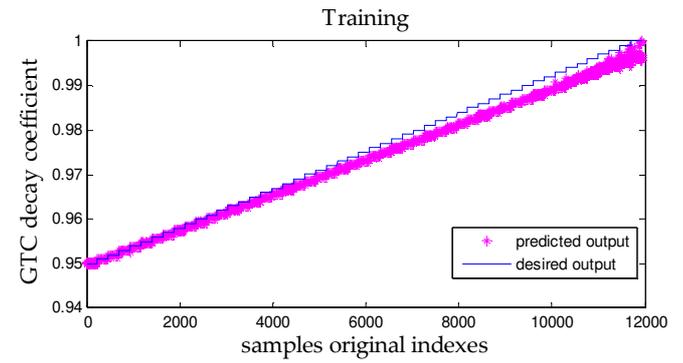
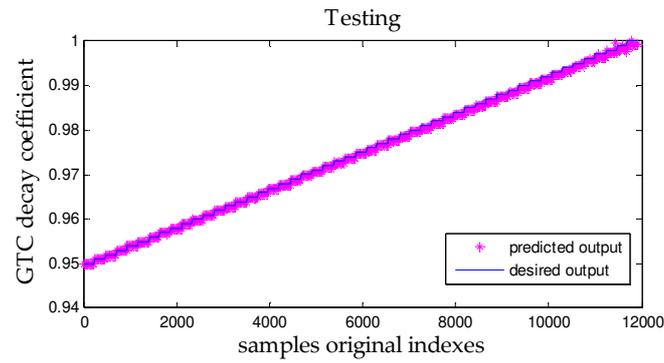
It is known that in ELM theories, more nodes in the hidden layer leads to more accurate tuning of the output weights by the Pseudo-inverse method. On this basis, a new comparative study is carried out in an attempt to prove the stability of the deep architecture, even under random feature mapping. By increasing the number of hidden nodes iteratively, the main goal is to ensure that the network always achieves the best results over the others.

**Figures 9 and 10** describe the studied networks error behavior during increasing the number of hidden nodes. Both figures highlight the network stability responses during model validation phases by showing less fluctuations and dynamic disturbances in the error rate behavior. In addition, under different incremental learning circumstances, C-DBN is still stable and has higher performance accuracy than ELM and OS-ELM.

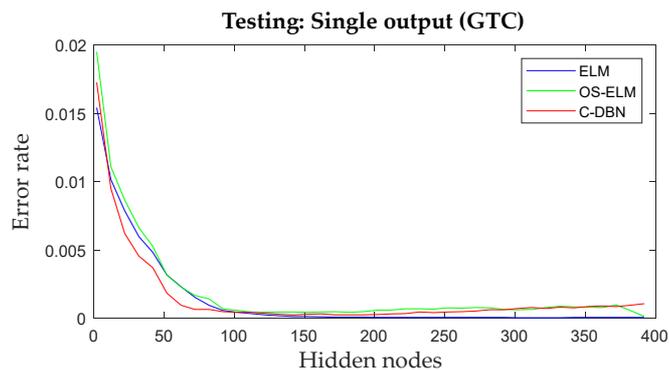
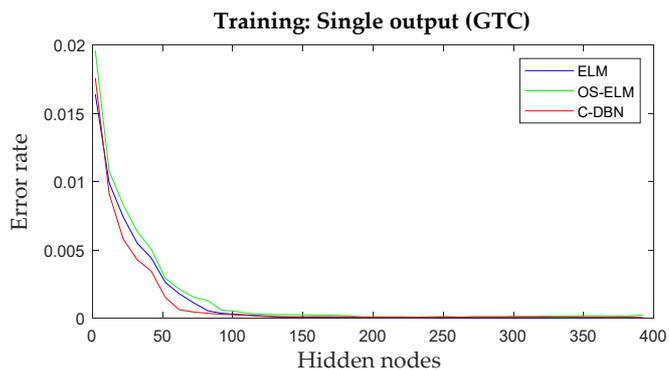
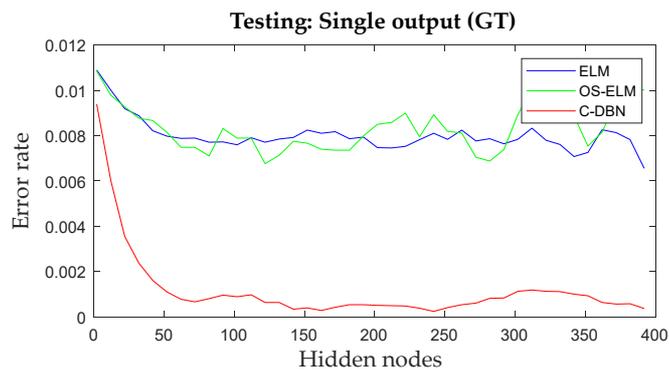
The new training scheme seems clearly more resistant against empirical risk. Indeed, as shown by **Figure 9** (testing with single output (GTC)), it can achieve better results than the other ones under less than 100 neurons. This leads to the conclusion that, the new network allows hidden layer in deep architecture to be more extended to give better a representation leading to an increased accuracy without suffering from overfitting compared to old paradigms.



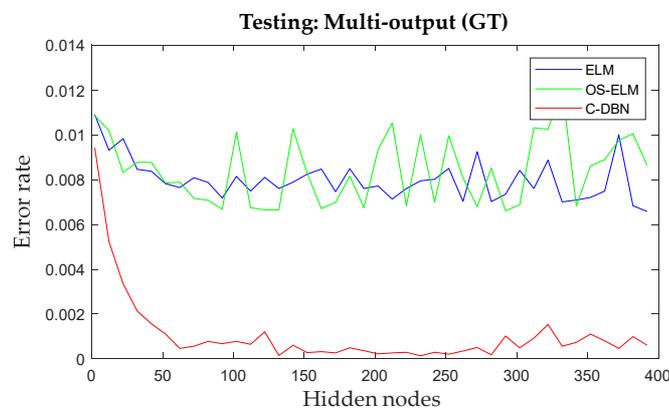
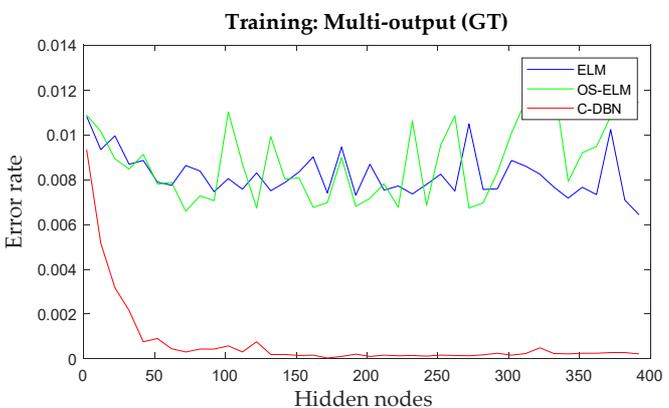
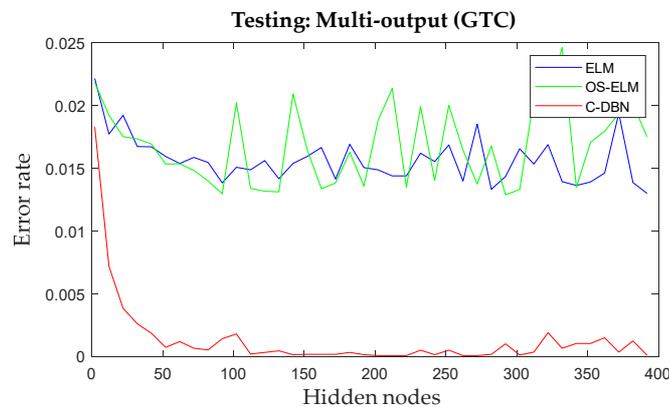
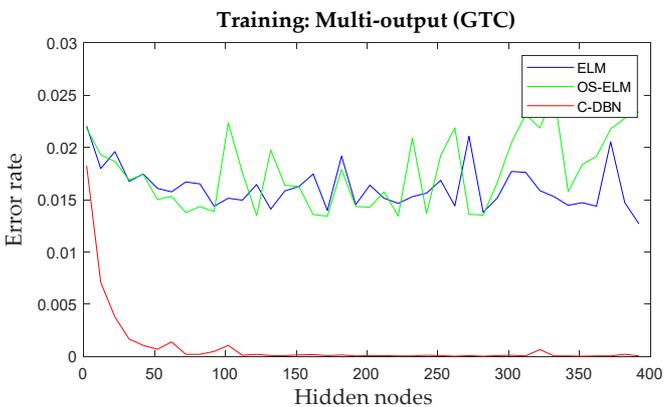
**Figure 7. Behavior of the proposed neural network in case of a single output.**



**Figure 8. Behavior of the proposed neural network in case of multi-outputs.**



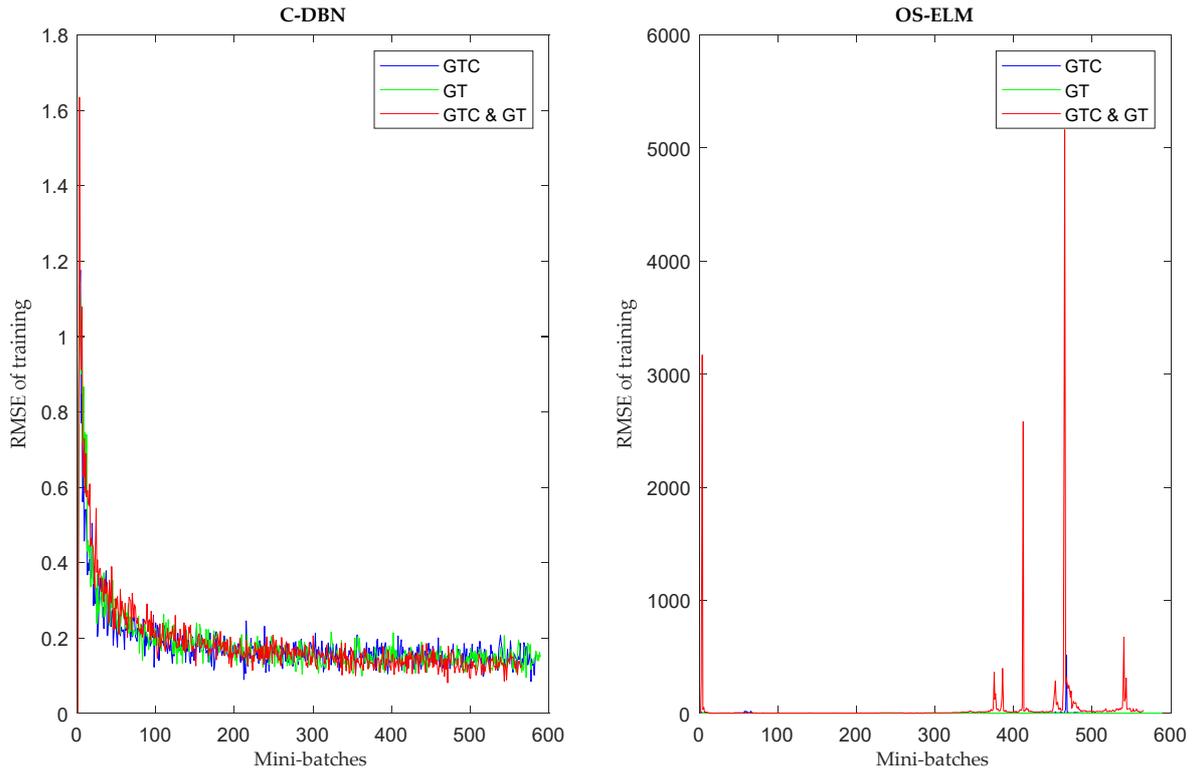
**Figure 9. Error rate behavior in case of a single output.**



**Figure 10. Error rate behavior in case of multi-outputs.**

When feeding each mini-batch entering the training model, a certain correlation between the last training model and the new driven data must be observed because this will prove that the model is generalized on the two sequences; The current and the last ones. Therefore, in order to prove the effectiveness of the adaptive learning of the C-DBN on the generalization process towards the newly arrived samples, additional experiments were investigated during the training process. The root-mean squared error (RMSE) of the prediction error (Equation (8)) using the latest output weights is recorded in each single iteration and the results are shown in Figure 11. In these experiments, the training data were divided into small mini-batches (20 samples) to study the behavior of the RMSE during sequential training. These results prove that the uniform gradual degradation of RMSE linked to C-DBN in the three cases (GTC, GT, GTC & GT) is more stable and show the generalization ability of the proposed approach. Indeed, unlike C-DBN, ordinary OS-ELM shows less generalization ability and learning stability to new samples by showing more fluctuations towards multiple sequences. These results show the effectiveness of the combination between convolutional mapping and deep learning under the recursive learning of OS-ELM.

In basic algorithms that have been previously discussed, such as SVM [5], SVM with isolation forest [6], and one-class SVM combined with global K-nearest neighbor [1], training algorithms have been developed attempting to solve the aforementioned condition-based maintenance problem by studying either a regression or a classification. Compared to the proposed algorithm, SVM is an offline learning method that is not able to address dynamic programming during adaptive learning with time varying data. Additionally, hybridization with SVM will produce more algorithmic complexity, which will ultimately result in higher computational costs. Moreover, unlike the OS-ELM, iterative gradient descent algorithms suffer from convergence problems when trying to find an optimal local minimum of the loss function. However, and due to the need for least squares methods of OS-ELM for full rank feature mapping, the flow of the mapping process to hidden layers in a deeper hierarchical architecture will cause important descriptive patterns to disappear. Therefore, careful selection of architecture is of a paramount importance.



**Figure 11. RMSE behavior during training process.**

## 5. Conclusion

This paper dealt with a new data-driven approach for predicting the decay of a combined Diesel electric and gas propulsion plant for naval propulsion systems. The proposed approach is a type of deep belief neural network (DBN) based on online sequential extreme learning machine (OS-ELM) rules, which has the capacity to perform convolutional mapping as well as the pooling in each single sub-network from its hidden layers according to ELM with local receptive fields theories. The introduced framework has been evaluated under *time varying* data, obtained from the system numerical model and compared with its original variants (ELM, OS-ELM). The results have shown that combinatorial DBN (C-DBN) is more efficient in terms of prediction capability, especially under single output. This makes it very promising for real world implementation of scheduled maintenance operations. This efficiency lies on the dynamic adaptation with the forgetting mechanism and regularization paradigm of the designed approach. In addition, extracting more significant features representation based on the designed filtering process and deep reconstruction proves that it plays an important role in accurate approximation as well as generalization.

This comparative study was carried out using random sampling and limited conditions. Consequently, further investigations need to be carried out to study such dataset using additional probabilistic distributions for sub-sampling and for more activation functions under cross-validation. Moreover, it should be interesting to investigate the use of random search algorithms for hyperparameters tuning to reach higher levels of accuracy and generalization.

## References

- [1] Cipollini, F., Oneto, L., Coraddu, A., Murphy, A. J., Anguita, D. (2018), Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback, *Reliability Engineering and System Safety*, 177: 12–23, 10.1016/j.ress.2018.04.015.
- [2] Tan, Y., Niu, C., Tian, H., Hou, L., Zhang, J. (2019), A one-class SVM based approach for condition-based maintenance of a naval propulsion plant with limited labeled data, *Ocean Engineering*, 193 (October): 106592, 10.1016/j.oceaneng.2019.106592.
- [3] Cipollini, F., Oneto, L., Coraddu, A., Murphy, A. J., Anguita, D. (2018), Condition-Based Maintenance of Naval Propulsion Systems with supervised Data Analysis, *Ocean Engineering*, 149: 268–278, 10.1016/j.oceaneng.2017.12.002.
- [4] Xiang, S., Qin, Y., Zhu, C., Wang, Y., Chen, H. (2020), Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction, *Engineering Applications of Artificial Intelligence*, 91 (February): 103587, 10.1016/j.engappai.2020.103587.
- [5] Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D., Figari, M. (2014), Machine learning approaches for improving condition-based maintenance of naval propulsion plants, 10.1177/1475090214540874.
- [6] Tan, Y., Niu, C., Tian, H., Lin, Y., Zhang, J. (2020), Decay detection of a marine gas turbine with contaminated data based on isolation forest approach, 5302, 10.1080/17445302.2020.1747750.
- [7] Chen, Z., Mauricio, A., Li, W., Gryllias, K. (2020), A deep learning method for bearing fault diagnosis based on Cyclic Spectral Coherence and Convolutional Neural Networks, *Mechanical Systems and Signal Processing*, 140:106683, 10.1016/j.ymsp.2020.106683.
- [8] Xiong, W., *et al.* (2018), Seismic fault detection with convolutional neural network, *Geophysics*, 83 (5):O97–O103, 10.1190/geo2017-0666.1.
- [9] Wen, L., Dong, Y., Gao, L. (2019), A new ensemble residual convolutional neural network for remaining useful life estimation, *Mathematical Biosciences and Engineering*, 16 (2):862–880, 10.3934/mbe.2019040.
- [10] Helbing, G., Ritter, M. (2018), Deep Learning for fault detection in wind turbines, *Renewable and Sustainable Energy Reviews*, 98 (January):189–198, 10.1016/j.rser.2018.09.012.
- [11] Wang, Y., Pan, Z., Yuan, X., Yang, C., Gui, W. (2020), A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network, *ISA Transactions*, 96: 457–467, 10.1016/j.isatra.2019.07.001.
- [12] Kong, Z. (2019), Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics, *Applied Sciences*, 2019, 9(19), 4156, 10.3390/app9194156.
- [13] García Nieto, P. J., García-Gonzalo, E., Sánchez Lasheras, F., De Cos Juez, F. J. (2015), Hybrid PSO-SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability, *Reliability Engineering and System Safety*, 138:219–231, 10.1016/j.ress.2015.02.001.
- [14] Dragomir, O. E., Gourtveau, R., Zerhouni, N., Dragomir, F. (2007), Framework for a distributed and hybrid prognostic system, *IFAC Proceedings Volumes*, 40 (18): 431–436, 10.3182/20070927-4-RO-3905.00072.
- [15] Lecun, Y., Bengio, Y., Hinton, G. (2015), Deep learning, *Nature*, 521 (7553): 436–444, 10.1038/nature14539.
- [16] Yin, B., Dridi, M., Moudni, A. El. (2019), Recursive least-squares temporal difference learning for adaptive traffic signal control at intersection, *Neural Computing and Applications*, 31: 1013–1028, 10.1007/s00521-017-3066-9.
- [17] Hinton, G. E. (2012), A practical guide to training restricted Boltzmann machines, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTU: 599–619, 10.1007/978-3-642-35289-8-32.
- [18] Deutsch, J., He, D. (2016), Using deep learning based approaches for bearing remaining useful life prediction, *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 2016-October: 1–7.
- [19] Gollapudi, S. (2019), “Deep Learning for Computer Vision,” in *Learn Computer Vision Using OpenCV*, Berkeley, CA: Apress, 2019, :51–69.

- [20] Zhu, J., Chen, N., Peng, W. (2019), Estimation of bearing remaining useful life based on multiscale convolutional neural network, *IEEE Transactions on Industrial Electronics*, 66 (4): 3208–3216, 10.1109/TIE.2018.2844856.
- [21] Feng, D., Xiao, M., Liu, Y., Song, H., Yang, Z., Hu, Z. (2016), Finite-sensor fault-diagnosis simulation study of gas turbine engine using information entropy and deep belief networks, *Frontiers of Information Technology & Electronic Engineering*, 17 (12): 1287–1304, 10.1631/FITEE.1601365.
- [22] Heimes, F. O. (2008), Recurrent neural networks for remaining useful life estimation, 2008 International Conference on Prognostics and Health Management:1–6, 10.1109/PHM.2008.4711422.
- [23] Chen, Z., Li, W. (2017), Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network, *IEEE Transactions on Instrumentation and Measurement*, 66 (7): 1693–1702, 10.1109/TIM.2017.2669947.
- [24] Liao, L., Jin, W., Pavel, R. (2016), Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment, *IEEE Transactions on Industrial Electronics*, 63 (11): 7076–7083, 10.1109/TIE.2016.2586442.
- [25] Huang, G. Bin. (2015), What are extreme learning machines? Filling the gap between Frank Rosenblatt’s dream and John von Neumann’s puzzle, *Cognitive Computation*, 7 (3): 263–278, 10.1007/s12559-015-9333-0.
- [26] Huang, G. Bin, Zhu, Q. Y., Siew, C. K. (2004), Extreme learning machine: A new learning scheme of feedforward neural networks, in *IEEE International Conference on Neural Networks - Conference Proceedings*, 2004, 2 (August): 985–990, , 10.1109/IJCNN.2004.1380068.
- [27] Huang, G. B., Chen, L., Siew, C. K. (2006), Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Transactions on Neural Networks*, 17 (4): 879–892, 10.1109/TNN.2006.875977.
- [28] Zhang, B., Li, Y., Bai, Y., Cao, Y. (2020), Aeroengines remaining useful life prediction based on improved C-loss ELM, *IEEE Access*, 8: 49752–49764, 10.1109/ACCESS.2020.2979747.
- [29] Berghout, T., Mouss, L. H., Kadri, O., Saïdi, L., Benbouzid, M. E. H. (2020), Aircraft engines remaining useful life prediction with an adaptive denoising online sequential Extreme Learning Machine, *Engineering Applications of Artificial Intelligence*, 96: 103936, 10.1016/j.engappai.2020.103936.
- [30] Huang, G. B., Bai, Z., Kasun, L. L. C., Vong, C. M. (2015), Local receptive fields based extreme learning machine, *IEEE Computational Intelligence Magazine*, 10 (2): 18–29, 10.1109/MCI.2015.2405316.
- [31] Zhu, W., Miao, J., Qing, L., Huang, G. Bin. (2015), Hierarchical extreme learning machine for unsupervised representation learning, *Proceedings of the International Joint Conference on Neural Networks*, 2015-Septe, 10.1109/IJCNN.2015.7280669.
- [32] Huang, G., Liang, N., Rong, H., Saratchandran, P., Sundararajan, N. (2005), On-line sequential extreme learning machine review of extreme learning machine (ELM), in *International Conference on Computational Intelligence*, 2005, (Ci).
- [33] “UCI Machine Learning Repository: Condition Based Maintenance of Naval Propulsion Plants Data Set.”.
- [34] Deng, W., Zheng, Q., Chen, L. (2009), Regularized extreme learning machine, 2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 - Proceedings, (60825202): 389–395, 10.1109/CIDM.2009.4938676.
- [35] Guo, W., Xu, T., Tang, K., Yu, J., Chen, S. (2018), Online sequential extreme learning machine with generalized regularization and adaptive forgetting factor for time-varying system prediction, *Mathematical Problems in Engineering*, 2018, 10.1155/2018/6195387.
- [36] Zhou, H., Huang, G. B., Lin, Z., Wang, H., Soh, Y. C. (2015), Stacked extreme learning machines, *IEEE Transactions on Cybernetics*, 45 (9): 2013–2025, 10.1109/TCYB.2014.2363492.
- [37] Miche, Y., van Heeswijk, M., Bas, P., Simula, O., Lendasse, A. (2011), TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization, *Neurocomputing*, 74 (16): 2413–2421, 10.1016/j.neucom.2010.12.042.