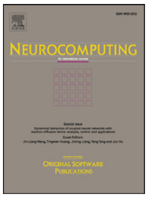
Neurocomputing 000 (2017) 1–18

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Ensemble of optimized echo state networks for remaining useful life prediction

Marco Rigamonti[a], Piero Baraldi[a,*], Enrico Zio[a,b], Indranil Roychoudhury[c], Kai Goebel[d], Scott Poll[d]

[a] Energy Department, Politecnico di Milano, Via Ponzio 34/3, Milan 20133, Italy
[b] Chair System Science and the Energy Challenge, Fondation Electricité de France (EDF), CentraleSupélec, Université Paris Saclay, Grande Voie des Vignes, 92290 Chatenay-Malabry, France
[c] Stinger Ghaffarian Technologies, Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA
[d] NASA Ames Research Center, Moffett Field, CA 94035, USA

ARTICLE INFO

Article history:
Received 1 March 2017
Revised 9 November 2017
Accepted 29 November 2017
Available online xxx

Communicated by H.R. Karimi

Keywords:
Echo state networks
Recurrent neural networks
Ensembles
Prediction uncertainty
Prediction Intervals
Differential Evolution

ABSTRACT

The use of Echo State Networks (ESNs) for the prediction of the Remaining Useful Life (RUL) of industrial components, i.e. the time left before the equipment will stop fulfilling its functions, is attractive because of their capability of handling the system dynamic behavior, the measurement noise, and the stochasticity of the degradation process. In particular, in this paper we originally resort to an ensemble of ESNs, for enhancing the performances of individual ESNs and providing also an estimation of the uncertainty affecting the RUL prediction. The main methodological novelties in our use of ESNs for RUL prediction are: i) the use of the individual ESN memory capacity within the dynamic procedure for aggregating of the ESNs outcomes; ii) the use of an additional ESN for estimating the RUL uncertainty, within the Mean Variance Estimation (MVE) approach. With these novelties, the developed approach outperforms a static ensemble and a standard MVE approach for uncertainty estimation in tests performed on a synthetic and two industrial datasets.

1. Introduction

Prognostics is the engineering discipline concerned with the prediction of the time left before the equipment will no longer perform its intended function, i.e., its Remaining Useful Life (RUL). Such prediction is typically performed starting from the current health state of the equipment and taking into account its past history and future operation. The capability of RUL prediction enables the possibility of identifying equipment problems at an early stage and timely performing maintenance to anticipate failures [41,65,68]. In this view, prognostics represents an important opportunity for industry, in terms of efficient and agile maintenance management, in principle providing the right part to the right place at the right time, with the necessary resources [23].

Prognostics require the availability of models capable of providing accurate RUL predictions and the associated uncertainty [46,88]. Prognostics models should take into account the different sources of uncertainty affecting RUL predictions [9,87]: (*i*) randomness in the equipment future degradation path, due to the intrinsic stochasticity of the degradation process and the unknown future operation and environmental conditions; (*ii*) inaccuracy of the prognostic model; (*iii*) measurement noise; and (*iv*) imperfect knowledge of the degradation initiation time.

Prognostic methods are typically classified as model-based, data-driven and hybrid [16]. Model-based methods use an explicit mathematical model of the degradation process to predict the future evolution of the degradation state and, thus, the RUL of the system [55]. In practice, even when the model of the degradation process is known, the RUL estimate may be difficult to obtain, since the degradation state of the system may not be directly observable and/or the measurements may be affected by noise and disturbances. In these cases, model-based estimation methods aim at inferring the dynamic degradation state and provide a reliable quantification of the estimation uncertainty on the basis of the sequence of available noisy measurements [27,28,44].

On the other side, data-driven methods are used when an explicit model of the degradation process is not available, but sufficient historical data have been collected. These methods are

[*] Corresponding author.
E-mail addresses: marcomichael.rigamonti@polimi.it (M. Rigamonti), piero.baraldi@polimi.it (P. Baraldi), enrico.zio@polimi.it, enrico.zio@ecp.fr (E. Zio), indranil.roychoudhury@nasa.gov (I. Roychoudhury), kai.goebel@nasa.gov (K. Goebel), scott.poll@nasa.gov (S. Poll).

https://doi.org/10.1016/j.neucom.2017.11.062
0925-2312/© 2017 Elsevier B.V. All rights reserved.

based on statistical models that ‘learn’ trends from the data [80]. In this respect, Artificial Neural Networks (ANNs) are often used [16,33,66]; other examples are Autoregressive Moving Average techniques [74], Relevance Vector Machines [25,33,74], fuzzy similarity-based methods [98]. Finally, hybrid approaches combine physics-based models of the degradation process with the use of historical data collected from degrading components [80].

Among these data-driven approaches, ANNs have often been used for time series forecasting, due to their capability of approximating non-linear complex functions [33]. Feedforward ANNs have been used in prognostics for the prediction of rotating machineries [56] and Lithium-ion batteries [77] RUL. However, the use of feedforward ANNs in prognostics is limited by the fact that they are direct models characterized by oriented connections among neurons, without feedback and loop connections. Therefore, since the output of any layer does not affect the same layer, feedforward ANNs are not able to catch the system dynamic behaviors. An attempt to provide the system dynamics in input to feedforward ANNs has been proposed in [93], where the models receive in input the current and past signal values collected in a time window. The main limitations of this approach are the difficulty in identifying the proper lengths of the time window and the largely increased number of model inputs.

Local field NNs [37,92], whose dynamics is based on the local field states of their neurons, have also been exploited for time series predictions. Their local associative memory properties have been applied with success to the problem of predicting groundwater levels [96], but, at the best of the authors knowledge, they have not been applied to prognostic problems. Alternatively, Spiking NNs, based on the use of spiking neurons characterized by internal states which change with time, are attractive for inherent dynamic problems such as those typical of prognostics [57]. General shortcomings of SNNs are the computational burden and the sensitivity of gradient descent-based learning algorithms to the SNN initial state [57]. Although significant advances have recently been made, these issues have not been fully resolved [32].

An alternative solution to the problem of learning the system dynamic using ANNs is given by Recurrent Neural Networks (RNNs). Since the RNNs internal states are characterized by cyclic connections and feedbacks among neurons, they are capable of encapsulating into their neurons a nonlinear transformation of the input history [18,21,45,54]. This provides memory properties to RNNs, enabling them to handle sequential tasks, such as time series prediction [79]. RNNs have been applied to different prognostic problems, such as the prediction of machine deterioration evolution using vibration data [86] and of helicopter drivetrain system gearbox [75] and turbofan engines [35] RULs. However, the application of traditional RNNs in time series forecasting problems is limited by the difficulty of optimizing their numerous internal parameters and the significant computational effort of the training process [53]. To overcome this problem, we exploit the use of Echo State Networks (ESNs), a relatively new type of RNNs. An ESN is an RNN trained by using a Reservoir Computing (RC) method based on the random generation of a RNN, called reservoir, which remains unchanged during the training phase and is passively excited by the input patterns. Since the only weights of the ESN to be optimized are those of the connections among the reservoir internal states and the output, RNN training is computationally more efficient [53].

Although ESNs have been shown to provide good generalization capabilities [47], few applications of ESNs for RUL prediction have been proposed. In [58], an ESN-based approach for the prediction of the RUL of industrial Fuel Cells has been developed. In [29], a hybrid approach combining ESN and Conditional Restricted Boltzmann Machines (CRBM) for predicting the occurrence of railway operation disruptions has been proposed. A possible reason for the limited use of ESNs in prognostics is the difficulty of setting of the ESN architecture parameters, which heavily influence the ESN modeling capability and typically requires a high level of expertise.

Ensemble of models has been used in many application fields for prediction accuracy improvement and uncertainty quantification [7,14,22,42,52,69]. The basic idea is that the diverse models in the ensemble complement each other by leveraging their strengths and overcoming their drawbacks. Thus, the combination of the outcomes of the individual models in the ensemble improves the accuracy of the predictions compared to the performance of a single model [10,14,17,39]. Also, the distribution of the individual model outcomes provides information on the ensemble modeling error [42,97]. Different methods, such as ANN [11], Support Vector Machine (SVM) [51] and kernel learning [52], have been used with success to build the individual models. For example, it has been shown that the use of an ensemble of just-in-time kernel learning models has allowed reducing the root mean square error of the prediction of the crystal size distribution in crystallization processes by 26% with respect to an individual model [52].

With respect to improving accuracy in prognostics, an ensemble of feedforward Artificial Neural Network (ANN) has been embedded into a Particle Filter (PF) for the prediction of crack length evolution [9] and an ensemble of data-driven regression models has been exploited for the RUL prediction of lithium-ion batteries [91].

As far as the authors know, the only application of ensembles of RNNs in prognostics has been presented in [39], where the RUL predicted by various data-driven models, including a RNN, were aggregated considering three different methods, and the developed procedure was applied to RUL prediction of turbofan engines, power transformers, and cooling fans. RNN-based ensembles have also been used for time series forecasting in [6,83] and in [94], where the considered RNN models were ESN.

In this work, we develop an ensemble of ESNs, whose model architecture is optimized by Multi-Objective Differential Evolution (DE) [73]. Differently from [73], where the ESN architecture corresponding to the DE chromosome giving the most accurate RUL predictions is selected, a novelty of this work is the use of the ensemble of ESNs whose architectures are given by the chromosomes of the Pareto front reached at DE convergence. This allows increasing prediction accuracy and estimating prediction uncertainty [14].

Once the individual models of the ensemble have been generated, it is also necessary to define a strategy for the aggregation of their outcomes. Aggregation methods are typically classified into static or local [14]: a static ensemble assigns the same weight to each model, regardless of the input pattern under test, whereas a local ensemble assigns a dynamic weight to each model according to its local performance measured considering input patterns similar to that under test, such as the nearest neighbors in a validation set. The accuracy of local ensembles has been reported to be more satisfactory than that of static ensembles in several applications [14,52,53,63,85]. The challenge for the application of local aggregation methods to ensembles of recurrent models, such as ESNs, is the evaluation of the local performances of the individual models. Since the ESN output does not depend only on the current input pattern but also from the previous input pattern history, due to its memory property, the identification of similar patterns is not straightforward. For example, if an ESN has large memory, i.e., the current output depends on a large input history, the input time window to be considered for the identification of the test pattern nearest neighbors should be long. On the contrary, if an ESN has low memory, i.e., the current output depends on a short input history, the input time window to be considered should be short. This problem has been here overtaken by proposing a novel local aggregation method which associates to each ESN a specific input time window whose width is directly related to its memory property, computed according to the method proposed in [48]. According to the author's best knowledge, the memory property has never been used for the aggregation of the ensemble individual model outcomes.

The second challenge addressed in this work is the quantification of the uncertainty affecting the RUL predictions. This is typically accomplished by providing Prediction Intervals (PIs), i.e. intervals of values within which the actual RUL is expected to lie with a predetermined probability [1,4,42,43]. In [36] and [97], a method to quantify the ensemble prediction uncertainty has been proposed, which distinguishes between the contributions due to the modeling error, estimated considering the distribution of the individual model outcomes, and to the other sources of uncertainty, estimated by a properly developed neural model. In [97] the Mean Variance Estimation (MVE) method has been proposed, which directly estimates the output uncertainty using a feedforward ANN. Practical applications of PI estimations can be found in [81], where the uncertainty of the estimation of safety parameters is quantified using bootstrapped ANNs; in [71], where the Lower Upper Bound Estimation method [43] is applied to electricity load prediction for separately estimating the lower and the upper bound of the prediction interval; in [8], where the PIs of the predicted turbine blade creep growth are estimated; in [1], where ANN is trained to provide the PIs of scale deposition rate in oil and gas equipment; in [38], where an ANN ensemble procedure embedding the Lower Upper Bound Estimation approach and Genetic Algorithm is used to improve the quality of PIs by optimizing the aggregation weights; in [2], where an ANN is used for the uncertainty quantification of short-term wind speed prediction. For a comprehensive review of ANN-based PIs, the interested reader can refer to [42].

The main difficulty to be tackled for quantifying the uncertainty affecting the RUL prediction provided by an ensemble of ESNs is the definition of the input of the model dedicated to the PI estimate, which should take into account the previous input history. The novelty of this work with respect to the estimation of the uncertainty associated to the RUL prediction is the use of an ESN model for the PIs estimation which, thanks to its memory property, allows automatically taking into account the memory of the ensemble. According to the authors best knowledge, RNNs have never been used within ensemble local aggregation methods nor for PIs estimation.

The proposed approach is verified with respect to synthetic, experimental and industrial case studies. The experimental case study concerns the prediction of the RUL of a fleet of turbofan engines working under variable operating conditions, whose data are taken from the NASA Ames Prognostics CoE Data Repository [76]. The industrial case study concerns the RUL prediction of a set of industrial knives used for packaging applications.

The results obtained on the considered case studies in terms of prediction accuracy and uncertainty estimation are compared to those obtained by using a static ensemble and the feedforward ANN-MVE approach, respectively.

The remaining part of the paper is organized as follows: Section 2 illustrates the work objectives and states the problem. Section 3 illustrates the developed method for RUL prediction and PI estimation, providing also a basic background on ESN and MVE. Section 4 shows the application of the proposed methodology to the two considered case studies and discusses the obtained results. Finally, in Section 5, some conclusions and remarks are drawn.

2. Notation and work objectives

We assume the availability of R run-to-failure trajectories describing the degradation of R similar components. The generic rth run-to-failure trajectory, $r=1,..,R$, consists of the time series of L signals collected through sensors from the degradation onset until the component failure time t_f^r. For the rth trajectory, the measurement of the L signals at the generic time t after the onset of the degradation process is indicated by:

$$\vec{x}_t^r = [x_t^{r,1} \quad x_t^{r,2} \quad .... \quad x_t^{r,L}] \quad t = 1, .., t_f^r \tag{1}$$

The objective of this work is the development of a prognostic model for the prediction of the RUL of a degrading industrial component [64], i.e., time span between the current time t and the time at which the component will not be able to perform its intended function. Therefore, the model receives an input $\vec{x}_{1:t}^r$, i.e. the measurements of the signals collected from the degradation onset until the present time t and provides in output an estimation $R\hat{U}L_t$of the component groundtruth RUL RUL_t^{GT}. Furthermore, this work also aims at estimating the uncertainty of the RUL prediction in the form of a PI $[L_\alpha; U_\alpha]$, i.e. an interval of values within which the groundtruth RUL, RUL_t^{GT}, is expected to lie with a predetermined probability $(1-\alpha)$.

3. Method

We have developed a method for RUL prediction and PI estimation based on an ensemble of ESNs. Section 3.1 describes the individual ensemble models and how diversity among the models is obtained; Section 3.2 illustrates the procedure used to aggregate the individual model outcomes and Section 3.3 presents the method for the estimation of the RUL prediction uncertainty.

3.1. Individual models

The individual models considered in this work are ESNs. The choice of this modeling technique is due to the capability of encapsulating the dynamic temporal behavior and preserving information about the input time history; this is obtained by using feedback connections between the neurons of a layer and those of the preceding layers [60]. Thanks to this, ESNs have intrinsic memory properties, i.e., the system output depends on the observed input time history [47], which is a desirable characteristic for prognostic models of dynamic degradation processes.

ESNs are a relatively new type of RNNs [47]. The difference from the traditional RNNs lies in the conceptual separation between the reservoir, a randomly created RNN used as nonlinear temporal expansion function, and a linear recurrence-free readout for synthesizing the expansion and producing the desired output (Fig. 1); notice that this latter is the only part of the ESN to be trained, which brings significant computational savings [53]. The critical part of the ESN is the *reservoir*, which is suggested to be generated of large dimension N but with sparse connections among the internal neurons in order to produce a rich set of dynamics [47,49].

In this work, we consider a discrete-time ESN with L input units receiving at time t the current signal measurements $\vec{x}_t = [x_t^1 \quad x_t^2 .... \quad x_t^L \ ]$. The reservoir is characterized by N internal network units whose internal states are represented by the vector $\vec{u}_t = [u_t^1 \quad u_t^2 ... \quad u_t^N \ ]$ and one output unit producing the output signals $y_t = RUL_t$. The activation of internal units u_t at time t is obtained using:

$$\vec{u}_t = \bar{f}(W^{in}\vec{x}_t + W\vec{u}_{t-1} + W^{back}y_{t-1}) \tag{2}$$

where $\bar{f} \ =(f_1 ... \ f_N)$ are the internal units activation functions, which are typically sigmoidal, $W^{in} = (w_{ij}^{in})$ is the $N \times L$ input

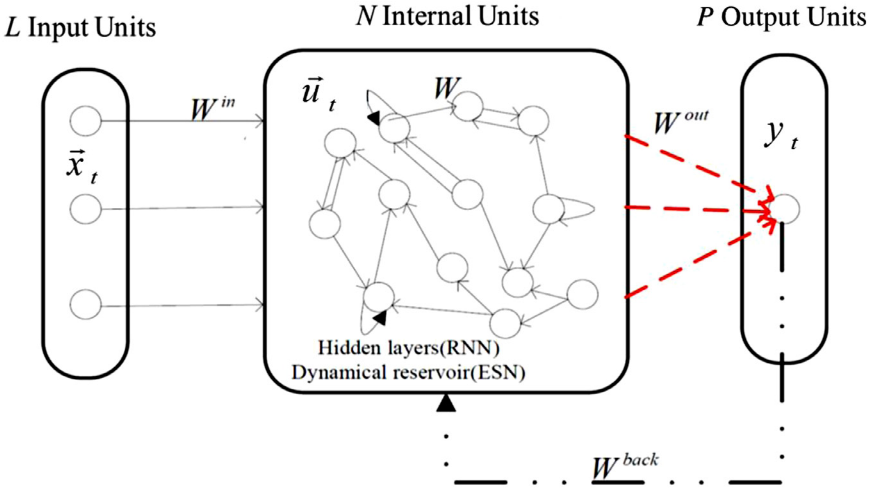


Fig. 1. Basic architecture of the ESN [67].

weights matrix, $W=(w_{ij})$ is the $N\times N$ internal weights matrix, and $W^{back}=(w_{ij}^{back})$ is the $N\times P$ output feedback weights matrix, where P represents the output channels (in this case $P=1$, i.e. the component RUL). The input weights W^{in} and the output feedback weights W^{back} are randomly generated from a uniform distribution. In order to deal with a specific task, both W^{in} and W^{back} can be scaled: the scaling of W^{in} (*IS*) and shifting of the input (*IF*) depend on how much nonlinearity of the processing unit the task needs. If the inputs are close to 0, the sigmoidal neurons tend to operate with activations close to 0, where they are essentially linear, while inputs far from 0 tend to drive them more toward saturation, where they exhibit more nonlinearity; the same idea drives the choice of the output scaling (*OS*) and shifting (*OF*), whose values affect the range of the trained W^{out} and might lead to an unstable condition. Finally, the scaling of W^{back} (*OFB*) is, in practice, limited by a threshold at which the ESN starts to exhibit an unstable behavior, i.e., the output feedback loop starts to amplify the output entering into a diverging generative mode [48]. The output provided by the ESN is:

$$y_t = \bar{f}_{out}\left(W^{out}(\vec{x}_t, \vec{u}_t, y_{t-1})\right) \tag{3}$$

where $\bar{f}_{out} = (f_{out}^1 \cdots f_{out}^P)$ are the output units activation functions, which are typically linear, and $W^{out}=(w_{ij}^{out})$ is the $P\times(L+N+P)$ output weights matrix. The ESN training aims at finding optimal values for W^{out} and is performed through a Least Squares linear regression step to minimize the error between the network output and a target signal on a set of training data. Once the ESN has been trained, it can be used to predict the output $y_t=RUL_t$, applying Eqs. (2) and (3) to the input $\vec{x}_t$.

The ESNs are characterized by the echo state property [47], which states that the effect of a previous state $\vec{u}_t$ and a previous input $\vec{x}_t$on a future state $\vec{u}_{t+j}$ should vanish gradually as time passes, and not persist or even get amplified. This property is practically assured if the reservoir weight matrix W is scaled so that its spectral radius (*SR*) $\rho(W)$ (i.e., the largest absolute eigenvalue of W) satisfies $\rho(W)<1$. For a detailed description of the ESN theory and application, the interested reader can refer to [47,49].

3.1.1. Multi-objective differential evolution for ESN architecture optimization

One main difficulty for developing an ESN model is the setting of the architecture parameters (Table 1); according to [49], this task requires a great level of expertise.

We adopt a Multi-Objective Differential Evolution (MO-DE)-based approach for the optimization of the ESN architecture. MO-DE is a parallel, direct, genetic-algorithm-based search method which manipulates a population of *NP* vectors of decision variable values $v_{i,G}$, $i=1, 2, . . ., NP$, called *chromosomes*, through an iterative search for solutions optimal with respect to some objective functions [84]. The initial population of chromosomes is sampled randomly from uniform probability distributions covering the decision variables range of values. Then, the DE approach is based on a three-step procedure: (*i*) *mutation*, which generates new chromosomes by adding the weighted difference between two chromosomes to a third chromosome, where each of these three chromosomes has been randomly selected; (*ii*) *crossover,* which mixes the mutated chromosome values with those of another predetermined chromosome, the target chromosome, to yield the so-called trial chromosome; and (*iii*) *selection,* which evaluates the objective functions of the trial chromosome and, if their values are better than those obtained with the target chromosome, keeps the trial chromosome in the population for the new generation in replacement of the target one. Details on DE can be found in Storn et al. [84].

Table 1
ESN architecture parameters.

ESN architecture parameters
Size of the reservoir (S)
Spectral radius (SR)
Connectivity (C)
Input scaling (IS) and shifting (IF)
Output scaling (OS) and shifting (OF)
Output feedback (OFB)

As objective functions to be maximized, in this work we consider the Cumulative Relative Accuracy (CRA) and Alpha-Lambda ($\alpha-\lambda$) accuracy prognostic metrics [78]. The CRA provides an average estimation of the RUL prediction relative error and, being a relative measure, tends to enlarge errors made at the end of the system life; on the other side, the $\alpha\ -\lambda$ accuracy indicates how many times, on average, the RUL prediction falls within two relative confidence bounds, without providing any information about how much this prediction is close to the target value. Since these two metrics estimate the RUL accuracy from different points of view, they have been chosen due to their complementarity. A detailed description of the considered metrics can be found in Appendix A.

3.1.2. Generation of an ensemble of diverse ESNs

According to [14], the accuracy of the overall ensemble model can be improved by enhancing the diversity among the individual models of the ensemble. In this work, model diversity is obtained by using (*i*) ESNs characterized by different architectures and (*ii*) ESNs trained using different datasets. With respect to (*i*), we build the models of the ensemble with the architecture parameters encoded in the chromosomes of the Pareto-optimal solutions found by the MO-DE optimization. Indeed, as the optimization is done with respect to two objectives, the optimal solutions are characterized by different trade-offs among them, and this leads to different Pareto-optimal solutions. Our approach is similar to that employed in [50], where an ensemble of ANNs has been developed by using the models contained in the last population of an Evolutionary Algorithm search, and in [83], where the RNN models of the ensemble have been chosen among those belonging to the Pareto-optimal solutions found by a multi-objective Evolutionary Algorithm. In our work, the diversity of the models is further enhanced by training each model on a different training set, randomly generated using the bagging algorithm [15].

3.2. *A local approach for the aggregation of the individual ESNs outcomes*

Static and local aggregations are the two most used procedures for the combination of the individual model outcomes into the ensemble aggregated outcome [14,69]. Static aggregation assigns a fixed weight to each model of the ensemble, independently from the input pattern. The weights can be equal for all the models or can be proportional to a measure of the individual model performance properly estimated on a set of input–output data. On the contrary, local aggregation dynamically assigns a weight to each model according to its local performance typically evaluated considering a fraction of the available historical input–output patterns characterized by input signal values similar to that of the test pattern [14,52]. The idea behind local aggregation is that the individual model performance is typically different in the different parts of the input domain.

In prognostics, local aggregation requires the computation of the local performance of the individual models on a set of R_{Test} run-to-failure degradation trajectories not previously used for training. In Section 3.2.1 the standard local aggregation procedure for non-recursive models is described, whereas in Section 3.2.2 we describe the new procedure purposely developed in this work for recursive models.

3.2.1. *The local aggregation procedure*

Considering an ensemble constituted by M ESNs that provide the RUL predictions $R\hat{U}L_t^m$, $m = 1,..,M$, in correspondence of the received input pattern $\vec{x}_t$, the local aggregation is based on the following steps:

1. Identify the input pattern $\vec{x}_{Nearest}^{r_{Test}}$ of each one of the R_{Test} trajectories with the minimum Euclidean distance from $\vec{x}_t$:

$$\vec{x}_{Nearest}^{r_{Test}} = \min_{j=1,..,t_f^{r_{Test}}} \left\{ d\left(\vec{x}_t, \vec{x}_j^{r_{Test}}\right) \right\},\ j = 1, .., t_f^{r_{Test}};\ r_{Test} = 1, .., R_{Test}; \tag{4}$$

2. Among the R_{Test} trajectories, identify the K trajectories $r_{Test,\ k}$, $k = 1,..,K$, with the corresponding $\vec{x}_{Nearest}^{r_{Test,k}}$ minimum Euclidean distances from $\vec{x}_t$.
3. Evaluate the absolute error, LE_m^k, $k = 1,..,K$, $m = 1,..,M$, of each of the M models on the set of K identified input vectors $\vec{x}_{Nearest}^{r_{Test,k}}$.

$$LE_m^k = \left| RUL^{GT}\left(\vec{x}_{Nearest}^{r_{Test},k}\right) - RUL^m\left(\vec{x}_{Nearest}^{r_{Test},k}\right) \right| \tag{5}$$

where $RUL^{GT}(\vec{x}_{Nearest}^{r_{Test},k})$ is the groundtruth RUL in correspondence of the kth identified nearest $\vec{x}_{Nearest}^{r_{Test,k}}$.

4. Compute the Local Error of each model (LE_m) as the sum of the K local errors:

$$LE_m = \sum_{k=1}^{K} LE_m^k \tag{6}$$

5. Assign to each model a weight inversely proportional to its local error:

$$\omega_m = \frac{\frac{1}{LE_m}}{\sum_{j=1}^{M} \frac{1}{LE_j}} \tag{7}$$

6. With the previously computed weights (Eq. (7)), compute the aggregated ensemble output $R\hat{U}L_t$ for the input pattern $\vec{x}_t$as the weighed sum of each model output $R\hat{U}L_t^m$:

$$R\hat{U}L_t = \sum_{m=1}^{M} \omega_m \cdot R\hat{U}L_t^m \tag{8}$$

With respect to steps 4 and 5 of the procedure, notice that they are analogous to that proposed in [14] and [52].

3.2.2. *Local aggregation: procedure for ensemble of recurrent models*

The application of a local approach to an ensemble of recurrent models requires the definition of a proper distance measure between the observed input patterns and those of the test set (Step 1 of the procedure of Section 3.2.1). Since our ensemble is formed by recurrent models, for which the model output depends not only from the present input pattern but also from previous input patterns [48], the distance cannot be computed by just taking into account the present input pattern as it is done when ensemble of feedforward ANNs are used. Therefore, we propose to compute the similarities considering signal time windows $\vec{I}_t^{Win} = [\vec{x}_{t-W_{IWin}+1}, \vec{x}_{t-W_{IWin}+2}, .., \vec{x}_t]$ of the input signals $\vec{x}_t$, whose width, W_{IWin}, depends from the memory property of the recurrent model. To this aim, we fix W_{IWin}equal to the model Memory Capacity (MC) measure [48], which quantifies the memory span of the ESN by measuring its capability of encapsulating a certain input span within its internal states, being able to *“remember”* and recall it. Details about the Memory Capacity computation can be found in Appendix B.

Considering the procedure in Section 3.2.1, the first step requires the identification of the input window of the generic r_{Test}th test trajectory, $r_{Test} = 1,..,R_{Test}$, characterized by the minimum distance to the present input time window $\vec{I}_t^{Win}$ (dashed box in Fig. 2). This requires the computation of the Euclidean distances between $\vec{I}_t^{Win}$ and all the time windows $\vec{I}_{t_{r_{Test}}}^{Win} = [\vec{x}_{t_{r_{Test}}-MC+1}, \vec{x}_{t_{r_{Test}}-MC+2}, .., \vec{x}_{t_{r_{Test}}}]$ with $t_{r_{Test}} = MC,\ldots,t_f^{r_{Test}}$, that can be extracted from the r_{Test}th test trajectory (boxes delimited by the continuous lines in Fig. 2). Since different ESN individual models are characterized by different memory capacities, the width of the similarity time windows associated to the different models will be different.

3.3. *Ensemble uncertainty estimation*

In this Section we present the method proposed for the estimation of the RUL PI, i.e. an interval defined by a lower and an upper bound $[L_\alpha(\vec{x}_t), U_\alpha(\vec{x}_t)]$, in which the unknown value of the groundtruth RUL at time *t*, RUL_t^{GT}, is expected to lie with a predetermined confidence level $(1-\alpha)$ [31,42,43]:

$$\Pr\left(L_\alpha(\vec{x}_t) < RUL_t^{GT} < U_\alpha(\vec{x}_t)\right) = 1 - \alpha \tag{9}$$

Several methods, such as the Gaussian Process Regression (GPR) [72], the Delta method [40], the Bootstrap method [36],

Please cite this article as: M. Rigamonti et al., Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.11.062

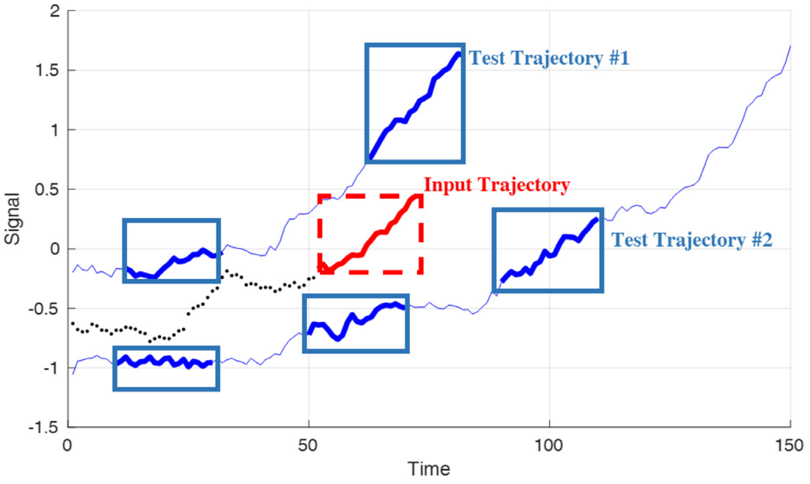


Fig. 2. Example of input time window (dashed) and time windows of the test trajectories used for the similarity comparison (solid lines).

the Lower Upper Bound Estimation method [43] and the MVE method [61], have been presented in literature for PI estimation [70,82,89,90,95].

In particular, GPR [72] is a probabilistic technique for non-linear, non-parametric regression that estimates the probability distribution of the output quantity based on Bayesian inference. It has been used in prognostics for the estimation of component future degradation, from which the component RUL can be estimated by referring to a failure threshold. In [12], GPR has been used for the estimation of the PIs of the predicted RUL of a condenser filter subject to clogging, without the need of any a priori hypothesis on the RUL distribution. The difficulty of this approach, however, is that it requires the setting of a failure threshold, which is often not feasible in practical prognostic problems.

Delta, MVE and Bootstrap methods assume that the prediction error, i.e. $\varepsilon_t = RUL_t^{GT} - R\hat{U}L_t$, which will be referred to as prediction residual, is an uncertain variable distributed according to a zero-mean Gaussian distribution whose variance $\sigma^2_{\varepsilon_t}(\vec{x}_t)$ has to be estimated. Since Delta method assumes that the variance of the prediction error is constant for all the input patterns, it is not apt for prognostic application where the prediction is expected to be more uncertain at the beginning of the component life than that at the end.

In this work we consider the MVE method, because it can be easily embedded within a local ensemble framework. The fundamental assumption of this method is the dependence of the target variance $\sigma^2_{\varepsilon_t}$ from the input pattern $\vec{x}_t$, i.e., the distribution of the prediction residuals depends on the region of the input domain to which the input $\vec{x}_t$ belongs [42]. Within the MVE approach, the target residual variance is estimated by developing a dedicated feedforward ANN with an exponential activation function that provides a strictly positive variance estimation. Notice that, since the true residual variance is not a priori known, it is not possible to uniquely estimate the residual associated to a given input pattern, but only the variance of the associated Gaussian probability distribution. Thus, the feedforward ANN dedicated to uncertainty estimation cannot be trained using error-based minimization techniques; rather, it is trained to maximize the likelihood function [42,61]:

$$P\left(RUL_i^{GT}|\vec{x}_i\right) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2_{\varepsilon_i}}} e^{-\frac{\left(RUL_i^{GT} - R\hat{U}L_i\right)^2}{2\hat{\sigma}^2_{\varepsilon_i}}} \qquad (10)$$

where, given the input vector $\vec{x}_i$, $\hat{\sigma}^2_{\varepsilon_i}$ is the estimated residual variance (the feedforward ANN output), $(RUL_i^{GT} - R\hat{U}L_i)^2$represents the squared RUL prediction residuals, and RUL_i^{GT} is the prediction target value (i.e., the groundtruth RUL of the component).

Considering the hypothesis that the residuals are normally distributed, the PI with a confidence level equal to $(1 - \alpha)$ is given by:

$$\left[R\hat{U}L_t - k_\alpha\hat{\sigma}_{\varepsilon_t}, R\hat{U}L_t + k_\alpha\hat{\sigma}_{\varepsilon_t}\right] \qquad (11)$$

where $\hat{\sigma}_{\varepsilon_t}$ is the residual distribution variance estimated in correspondence of the input pattern $\vec{x}_t$ and k_α is the parameter representing the $(\alpha/2)$ quantile of a Student t-distribution with a degree of freedom equal to the number of ensemble models M [36,42].

Notice the assumption that the prediction residual is locally distributed according to a zero-mean Gaussian distribution may not fit some practical applications, in which case the prediction intervals obtained could be affected by errors. In the case of the empirical distribution of the prediction error, the MVE method can be adapted by properly modifying the likelihood function in Eq. (10) and the k in Eq. (11). For the case of exponential distributions, the interested reader can refer to [89,90].

The main issue to tackle the computation of the PI provided by an ensemble of recursive models is that the input pattern $\vec{x}_t$ influencing the RUL prediction $R\hat{U}L_t$and, thus, the corresponding residual ε_t is not clearly defined. In practice, at each time step t the model receives in input the current signal measurements $\vec{x}_t$ which, thanks to the memory properties of the recurrent models, will continue to influence the future output predictions $R\hat{U}L_\tau$, $\tau > t$. Therefore, for obtaining accurate estimations of the prediction uncertainty, it is necessary to consider the significant previous input history, which results in the need of identifying the input vector of the feedforward ANN-MVE. Since the ensemble is formed by several models, each one characterized by a different Memory Capacity, the problem cannot be tackled by considering a time window, as in the previous section.

In order to overcome this problem, another novelty of this work is the use of another ESN for the estimate of the residuals variance, which will be referred to as ESN-MVE. Thanks to its memory property, the ESN-MVE will automatically take into account the significant previous input history, providing more reliable and accurate local PIs. Notice that the ESN-MVE output weights W^{out} are optimized to maximize the likelihood objective function in Eq. (10).

Please cite this article as: M. Rigamonti et al., Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.11.062

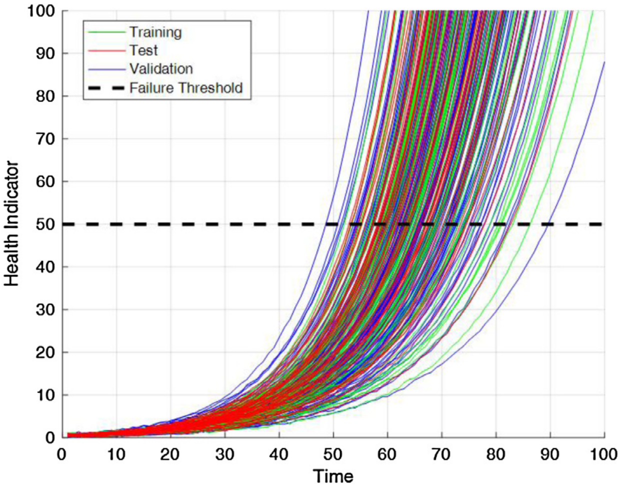


Fig. 3. Evolution of the health indicator during the 250 simulated degradation trajectories.

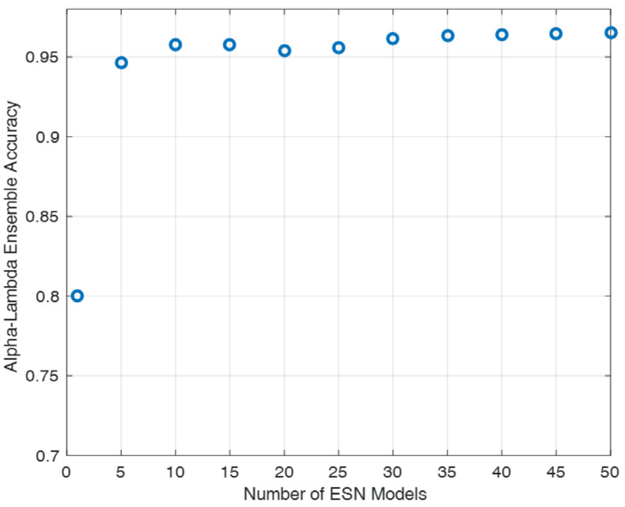


Fig. 4. Influence of the number of ESN models on the ensemble $\overline{\alpha-\lambda}$ accuracy.

4. Case study

The developed prognostic approach for RUL prediction and uncertainty estimation has been verified considering both a synthetic and an experimental case study.

4.1. Synthetic case study

The synthetic case study mimics the degradation of a fleet of 250 similar components from pristine conditions until failure. The values of a health indicator, x^H, are measured with time interval $\Delta t = 1$ in arbitrary units, during each one of the 250 run-to-failure trajectories. The data have been simulated by using a 10 time steps Auto Regressive (AR) process [3]:

$$x_t^H = \vec{a} \cdot \vec{x}^H_{t-10:t-1} + \nu \tag{12}$$

where x_t^H indicates the value of the health indicator at time t, $\vec{a}$ is a vector of 10 parameters randomly sampled from a uniform distribution in [0.12: 0.17] for each degradation trajectory, $\vec{x}^H_{t-10:t-1}$ represents the vector containing the values of the health indicator between *(t – 10)* and *(t – 1)*, and ν represents a normally distributed zero-mean process noise, whose standard deviation has been set equal to $\sigma = 0.2$ and which is sampled at each time step. Notice that the sampled vector $\vec{a}$ is kept fixed for a whole run-to-failure trajectory to represent the individual characteristics of the degrading component, whereas the random noise has been introduced to represent the stochasticity of the degradation process. Fig. 3 shows the degradation indicator time evolution for the 250 simulated degradation trajectories.

Notice that although the different degradation trajectories are characterized by similar functional behaviors, they show very different growth rates, leading to quite spread values of the corresponding failure times (i.e., between $t = 46$ and $t = 90$). Since the degradation evolution at time t described by Eq. (12) depends not only from the degradation value at the previous time *(t –1)*, but also from the past degradation history until time *(t – 10)*, it is not possible to properly predict the future evolution and the failure time relying only on the last observed value x_t^H, but it is necessary to take into account also the previous history of x^H.

The simulated dataset has been divided into the following subsets:

- **Training Set**: 60 trajectories used to train the ESN models.
- **Test Set**: 50 trajectories used for: *i*) optimizing the ESN architecture and the ensemble parameters, such as the number of ensemble models M and the number of trajectories in the bagging training set, *ii*) computing trajectory similarities within the local aggregation procedure and *iii*) training the ESN-MVE for PI estimation.
- **Validation Set:** 140 trajectories used to evaluate the proposed method performance.

4.1.1. Ensemble RUL prediction

According to Section 3.1.1, the ESN parameters have been automatically optimized by using a MO-DE algorithm with CRA (Eq. (A.1)) and the $\overline{\alpha-\lambda}$ accuracy (Eq. (A.3)) [77] as objective functions to be maximized. The Differential Amplification Factor (F) and Crossover Rate (CR) used by the DE search have been set equal to 0.75 and 0.5, respectively, according to the guidelines in [13,34] and [5] and trial and error experiments using the Test Set data. The DE population has been taken of 200 chromosomes and has evolved for 50 generations.

According to the local aggregation procedure described in Section 3.2.1, we have selected $M = 25$ ESNs corresponding to all the 7 solutions of the Pareto optimal front, all the 12 solutions of the second layer of the Pareto front and 6 solutions randomly chosen among the 16 solutions belonging to the third layer of the Pareto front. The diversity among the ensemble individual models has been further increased through the use of the bagging procedure [15], which resulted in training each selected ESN with 5 randomly chosen training trajectories. The number of models M and the number of trajectories in the training set have been fixed by adopting a trial and error procedure on the test set trajectories. Fig. 4 shows the influence of the number M of ESN models on the ensemble $\overline{\alpha-\lambda}$ accuracy. Notice that although the trend of the performance metric is monotonously increasing with the number of ensemble individual models, the performance improvement becomes small when we use more than 10 models. Therefore, to obtain a good compromise between ensemble prediction accuracy and computational time of the ESN ensemble, we have built an ensemble made by $M = 25$ ENS models.

To apply the proposed local aggregation approach described in Section 3.2.1, we have computed the Memory Capacity measures of the 25 generated ESNs and used them for defining the corresponding time window width $W^m_{|Win}$, $m = 1,..,25$, employed by the local aggregation technique according to the procedure described in Section 3.2.2. Table 2 compares the prognostic performance provided by the proposed local ensemble, provided by an individual ESN model, selected among the optimized models by

Table 2
Comparison of the prognostic performance provided by the individual ESN, the static and the proposed local ESN ensemble.

RUL prediction	Individual ESN model	Static ensemble	Proposed local ensemble
Cumulative relative accuracy (CRA)	0.913	0.914	0.941
Alpha–Lambda accuracy ($\alpha = 20\%$)	0.939	0.943	0.981

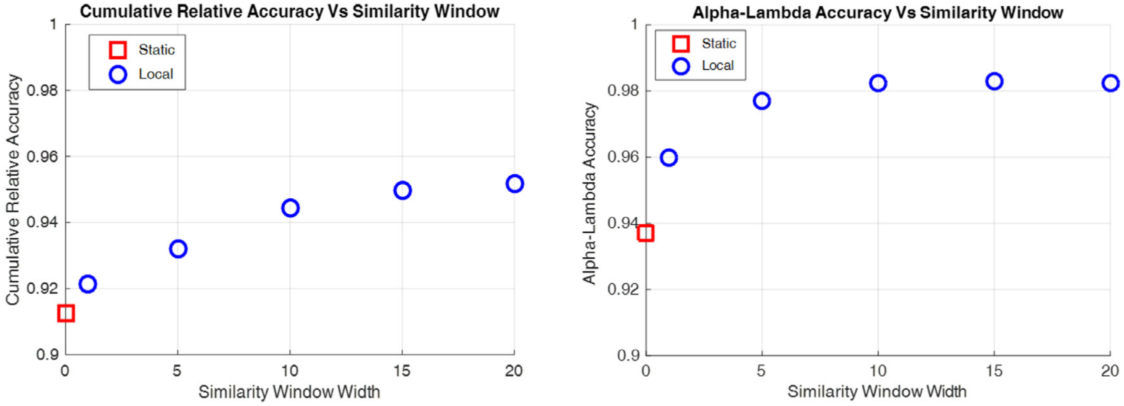


Fig. 5. Ensemble performance in terms of CRA (left) and $\overline{\alpha - \lambda}$ accuracy (right) as a function of the similarity time window width.

applying the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [20,62], and that provided by a static ensemble which assigns the same weight to each model, independently from the location of the training pattern.

Notice that the proposed local aggregation procedure allows obtaining an improvement of 3% of the CRA metric and of 4% of the $\overline{\alpha - \lambda}$ accuracy with respect to both the individual ESN and the static ensemble. The influence of the similarity time window width on the ensemble performance has been further investigated by considering a new ensemble of ESNs all with the same architecture corresponding to that of the TOPSIS solution and, therefore, by the same Memory Capacity of 9.6.

Fig. 5 shows the performance of the ensemble obtained by changing the similarity time window width. The $\overline{\alpha - \lambda}$ accuracy (right) significantly increases until the similarity time window width reaches 10, i.e., the closest integer to the ESN Memory Capacity value. With respect to the CRA, for similarity time windows width larger than 10 the performance continues to slightly increase showing the beginning of an asymptotic behavior. Notice, however, that increasing the width of the similarity time window over the corresponding ESN's Memory Capacity does not allow significantly improving the ensemble predictive performance, while it requires to wait a larger time interval before providing an RUL prediction. Therefore, the obtained results confirm that the choice of a similarity time window width equal to the memory capacity of the ESN allows properly representing the recurrence of the ESN within the local aggregation procedure.

4.1.2. RUL prediction uncertainty: prediction intervals estimation

In this Section, we compare the approach proposed in Section 3.3 to properly estimate the RUL prediction uncertainty with a literature approach based on the use of a feedforward ANN within the MVE method. To this aim, we assess the quality of the estimated PIs by considering the coverage probability and the prediction interval width [59]. The coverage probability is evaluated by the Prediction Interval Coverage Probability (PICP) metric, which quantifies the probability that the estimated PI values contain the groundtruth RUL values, and is estimated by:

$$PICP = \frac{1}{N_{targ}} \sum_{i=1}^{N_{targ}} c_i \quad \text{with} \quad \begin{cases} c_i = 1 & if\, RUL_i^{GT} \in [L_\alpha(\vec{x}_i), U_\alpha(\vec{x}_i)] \\ c_i = 0 & otherwise \end{cases} \tag{13}$$

The Normalized Mean PI Width (NMPIW) metric quantifies the average PI's width normalized with respect to the target value [42]:

$$NMPIW = \frac{1}{N_{targ}} \sum_{i=1}^{N_{targ}} \frac{(U_\alpha(\vec{x}_i) - L_\alpha(\vec{x}_i))}{RUL_i^{GT}} \tag{14}$$

Since the objectives are to have the smallest PIs with the largest possible probabilities of containing the true component RULs, one should therefore aim at simultaneously maximizing the PICP and minimizing the NMPIW [1].

The architecture of the ESN for RUL uncertainty estimation has been optimized using a DE algorithm with objective the maximization of the likelihood function defined in Eq. (10). To this aim, the Test Set has been divided into two subsets used for ESN training and estimation of the objective function, respectively. It is worth noting that the ESN model for RUL uncertainty estimation is fed by only the current values of the prognostic signals (i.e., in this case, the current health indicator value x_t^H). With respect to the development of the MVE approach using a traditional feedforward ANN, we have considered time windows $[x_t^H, x_{t-1}^H, ..., x_{t-n}^H]$ of different width $(n+1)$ as input to the feedforward ANN model developed for the RUL uncertainty estimation.

The results in Fig. 6 show that the 90% PI confidence provides satisfactory coverage values independently from the time window width, whereas the narrowest PI is obtained by using a time window width of 10 time units. This confirms that the significant part of trajectory to be considered in order to provide accurate estimation of the current RUL and of the associated uncertainty is equal to 10, which corresponds to the grade of the AR process used to simulate the data. Satisfactory coverage values are also obtained when different PI confidence values are used (e.g. 80%, 95% and 98%).

Please cite this article as: M. Rigamonti et al., Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.11.062

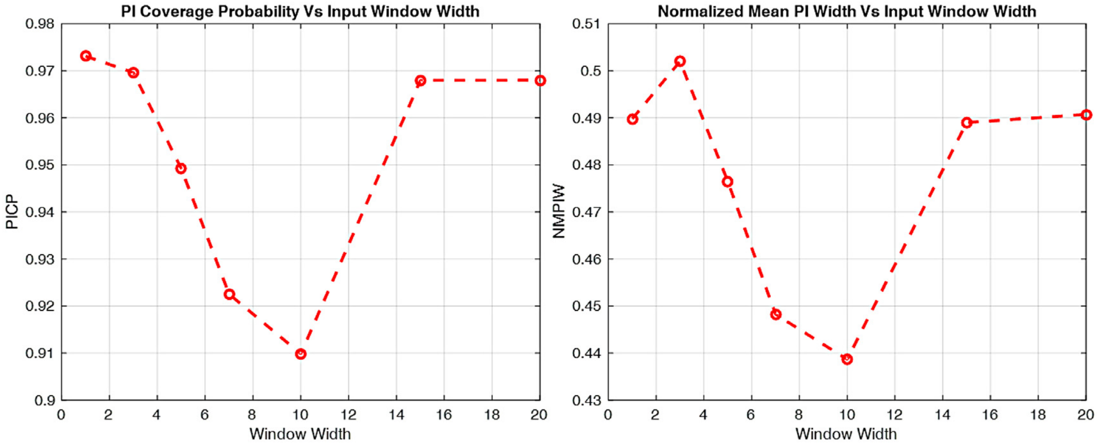


Fig. 6. PICP (left) and NMPIW (right) obtained by using feedforward ANN-MVE with different input window width.

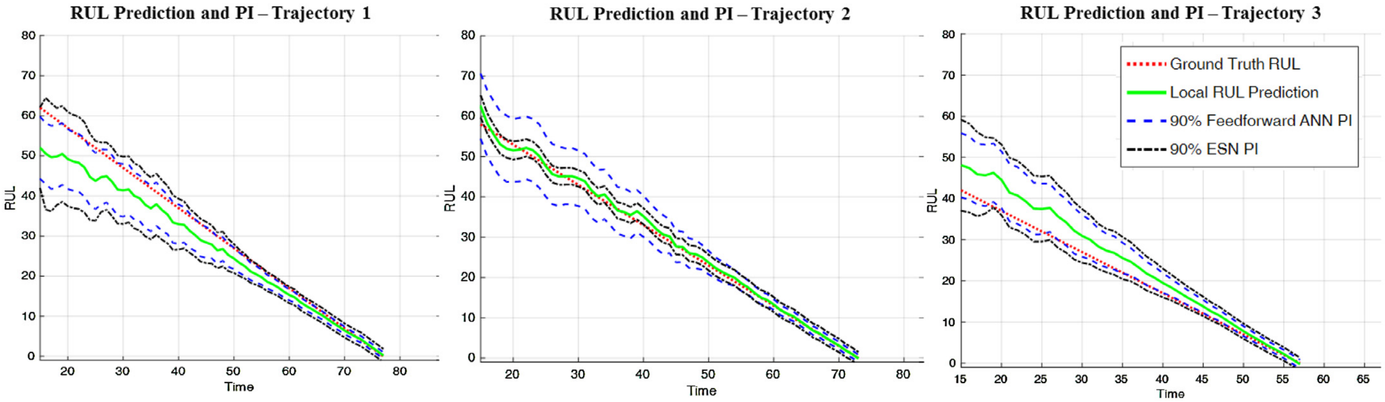


Fig. 7. Prediction of the RUL (solid line) and of the PIs provided by the feedforward ANN-MVE method (dashed line) and the ESN-MVE method (dot-dashed line) for 3 representative test trajectories.

Table 3
PIs performance comparison between the proposed ESN-MVE approach and the traditional feedforward ANN-MVE approach with the time window width of 10 time units.

Prediction interval 90% confidence	Feedforward ANN-MVE	ESN-MVE
PICP (average coverage)	0.91	0.93
NMIPW (average relative width)	0.44	0.42

With respect to the comparison of the PI performance provided by the feedforward ANN-MVE and proposed ESN-MVE methods, Table 3 reports that the proposed approach allows improving the PIs coverage probability (PICP is increasing of 2%) while reducing the PIs width (NMPIW is decreasing of 2%).

Fig. 7 shows 3 examples of RUL prediction provided by the proposed local ensemble and the corresponding PIs provided by both the feedforward ANN-MVE (dashed line) and the ESN-MVE (dot-dashed lines) approaches. Notice that when the difference between the predicted and the groundtruth RUL (dotted line) is relatively large (left and right Figures at the beginning of the component life), the ESN approach provides a PI larger than that provided by the feedforward ANN approach, whereas when the prediction residual is low (central Figure), the PI provided by the ESN is much narrower than that provided by the feedforward ANN.

To further investigate the capability of the ESN-MVE method of properly representing the local RUL prediction uncertainty, we have investigated the capability of the method of properly estimating the variance $\sigma^2_{\varepsilon_t}$ of the residual $\varepsilon_t = RUL_t - \hat{RUL}_t$ considering the three degradation trajectories reported in Fig. 8, characterized by very different end-of-life time instants, i.e., $t_f = 56$, $t_f = 73$ and $t_f = 77$, at 7 time instants between $t = 20$ and $t = 50$.

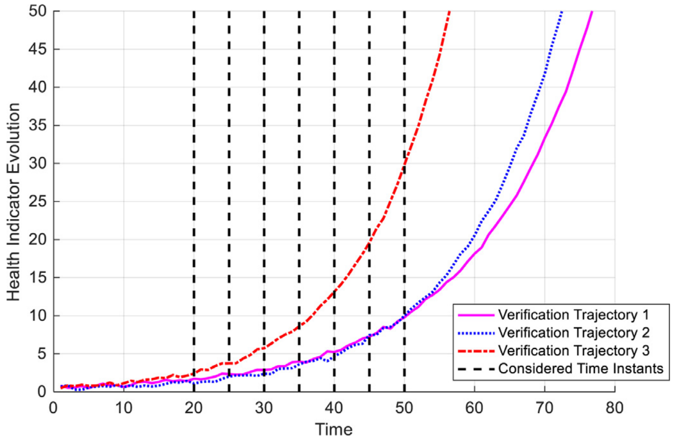


Fig. 8. Health Indicator evolution of the three verification trajectories.

In this synthetic case study, the (true) residual variance σ^2_{ε}can be estimated by simulating 1000 degradation trajectories using Eq. (12), all starting from the same degradation value and with the same $\vec{a}$ vector characterizing the degrading component. Then, the (true) residual variance $\sigma^2_{\varepsilon}$is approximated by the variance of the residuals between the RUL prediction provided by the ensemble and the 1000 RULs of the simulated trajectories.

Fig. 9 shows the comparison among the real RUL prediction residuals variance (circle-solid lines) computed trough the de-

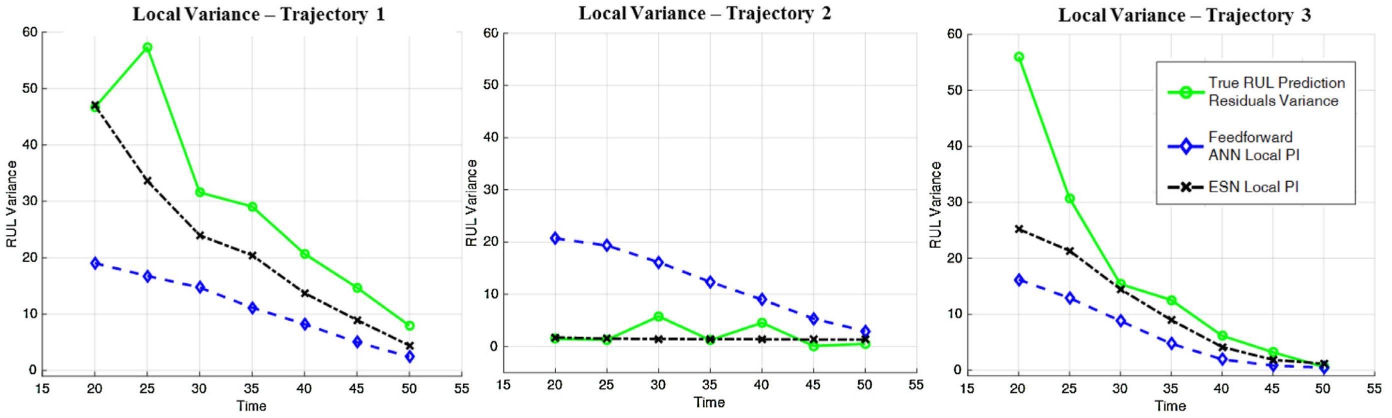


Fig. 9. True RUL prediction residuals variance (solid line), feedforward ANN estimated variance (dashed line), ESN estimated variance (dot-dashed line) for the 3 verification trajectories at times (20:10:70).

scribed Monte Carlo simulation (Eq. (12)), the variance estimated by the feedforward ANN-MVE approach (square-dashed lines) and that estimated by the proposed ESN-MVE approach (dot-dashed crossed lines).

The results confirm that the local variance estimated by the ESN-MVE is always closer to the true RUL prediction residuals variance than that estimated by the feedforward ANN-MVE.

4.2. CMAPPS case study

The proposed methods for RUL prediction and PI estimation have been further verified considering data describing the degradation of a fleet of turbofan engines working under variable operating conditions.

The dataset has been taken from the NASA Ames Prognostics CoE Data Repository [76], and consists of 218 run-to-failure trajectories. Each trajectory is a 24-dimensional time series of different length, formed by 21 signals describing the component operation and 3 signals referring to the turbofan engines operating conditions (Altitude, Mach number and Throttle Resolver Angle). These latter signals define six different operating conditions which significantly influence the values of the other 21 measured signals. The run-to-failure trajectories have been generated using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) model that receives as input an evolving health indicator (i.e., a parameter representing the degradation level of an engine component) and provides as output the values of the signals [30]. Process noise has been added and percolated throughout the different stages of the simulation model and random measurement noise added to the output signals. This multi-stage noise contamination process has produced complex noise characteristics similar to those often observed in real data, thus posing a realistic challenge to the RUL prediction. In this work, we have considered the data belonging to the training set of the 2008 PHM Challenge Dataset, which are characterized by the occurrence of a single failure mode caused by the degradation of the High Pressure Compressor (HPC) of the engine.

The dataset, containing 218 run-to-failure trajectories, has been partitioned into the following 3 subsets:

- **Training Set**: 60 trajectories used to train the ESN models.
- **Test Set**: 50 trajectories used for: (*i*) optimizing the ESN architecture and the ensemble parameters, (*ii*) computing trajectory similarities within the local aggregation procedure, and (*iii*) training the ESN-MVE for PI estimation.
- **Validation Set:** 108 trajectories used to evaluate the prognostic and the uncertainty estimation performances.

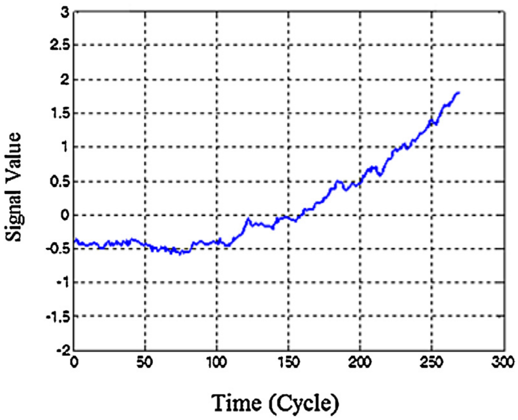


Fig. 10. Pre-processed and filtered behavior of Signal 11 of a run-to-failure trajectory.

A three-step preprocessing procedure, which includes (*i*) the normalization of the signals with respect to the operating conditions, (*ii*) the filtering of the normalized signals, and (*iii*) the identification of the most significant prognostic signals to be used as input of the prognostic model, has been applied to deal with the complexity of the data [73]. As an example, Fig. 10 shows the time evolution of pre-processed and filtered signal 11 of a run-to-failure trajectory.

With step (*iii*), we have identified the 6 most significant prognostic signals to be used as input to the prognostic models. Finally, we have considered that only the accelerated degradation phase of the run-to-failure trajectories has a strict physical relationship with the component RUL; therefore, we have applied the Z-Test [24] to identify the *elbow point* of the considered trajectory, i.e. the time instant at which the accelerated degradation phase begins, and have considered only the data subsequent to the detected elbow point time instant. More details on the 3 steps preprocessing procedure and elbow point identification can be found in [73].

4.2.1. RUL prediction

An ensemble of 50 models has been developed by applying the procedure of Section 3. The MO-DE search has produced 200 Pareto solutions, each one corresponding to an ESN architecture characterized by a different trade-off between the two objectives (CRA and $\overline{\alpha - \lambda}$ accuracy) and different classes of dominance. The MO-DE search has required 50 h of computation using an Intel Core i7 4700HQ CPU @ 2.40 GHz Personal Computer. Among the 200 identified solutions, we have selected the $M = 50$ ESN

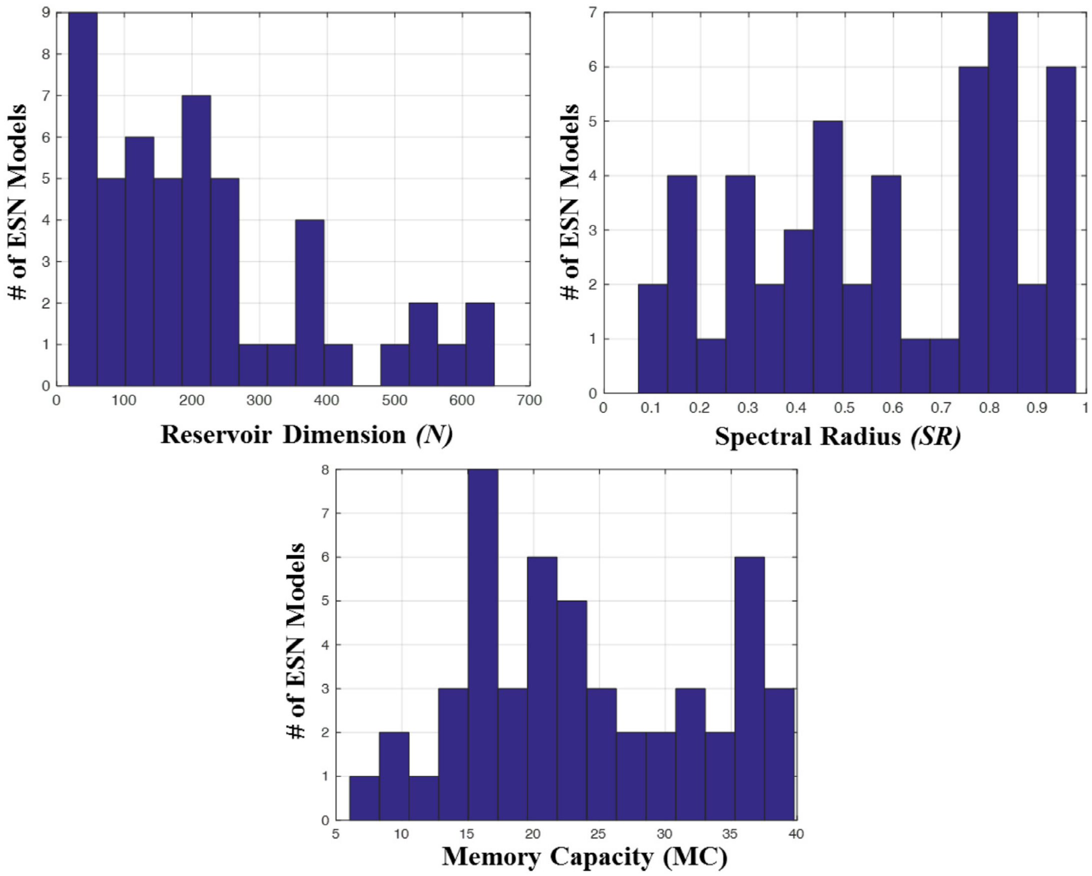


Fig. 11. Histogram of the reservoir dimensions (upper-left), spectral radii (upper-right) and memory capacities (bottom) of the ensemble individual models.

architectures with the most satisfactory trade-off between the objectives, i.e. the solutions of the first layers of the Pareto front (16 solutions of the Pareto optimal front, 23 belonging to the second layer of the Pareto front and 11 solutions randomly chosen among those belonging to the third layer of the Pareto front). Notice that the first source of diversity within the ensemble individual models is given by the different ESN architectures provided by the MO-DE search. Fig. 11 shows the diversity of the ensemble models in terms of the spread among the reservoirs dimension N, the spectral radii SR and the obtained Memory Capacities.

Models diversity has been further enhanced by training each ESN model with 5 trajectories randomly sampled by the bagging procedure [15] from the 60 trajectories of the training set.

The developed local ensemble has been applied to the prediction of the RUL of the Validation Set trajectories, and the obtained performances have been compared to that provided by the best single ESN predictor (i.e., the TOPSIS solution), by a local ensemble of $M = 50$ ESNs all characterized by the TOPSIS architecture and trained using different, randomly chosen, degradation trajectories according to the bagging procedure (i.e., the TOPSIS local ensemble), and by the DE static ensemble, whose output is the average of the ensemble models outcomes. Being the ESN training procedure very fast, the computational time required for training and testing all the developed predictive models has been of only 200 s using an Intel Core i7 4700HQ CPU @ 2.40 GHz Personal Computer.

The obtained results are reported in Table 4.

Considering the large number of validation trajectories, $R_{Val} = 108$, we can conclude that the proposed ESN-DE local ensemble provides a statistically significant improvement of the predictive performance: CRA increases by 15% with respect to the best performing ESN model, by 11% with respect to the DE static ensemble and by 7% with respect to the TOPSIS local ensemble. Furthermore, the proposed ESN-DE local ensemble allows slightly increasing the alpha-lambda accuracy with respect to all the other considered approaches.

To further investigate the performance of the proposed ESN-DE ensemble, we quantify its models diversity by resorting to the Ambiguity Decomposition index proposed in [17] which distinguish between the accuracy and the diversity of the M ensemble individual models.

The Individual Error Term quantifies the accuracy of the ensemble individual models with respect to the prediction target:

$$\textbf{Individual error term}: \sum_{m=1}^{M} w_m \left(\widehat{RUL}^m - RUL^{GT}\right)^2 \tag{15}$$

whereas the Ambiguity Term measures how much the individual models predictions are spread around the predicted RUL:

$$\textbf{Ambiguity term}: \sum_{m=1}^{M} w_m \left(\widehat{RUL}^m - \widehat{RUL}\right)^2 \tag{16}$$

Table 5 compares the Individual Error and Ambiguity terms of the TOPSIS and of the DE-ESN ensembles.

The results show that the more satisfactory performance of the ESN-DE ensemble is due to the larger diversity of its individual models (Ambiguity Term) which is able to compensate the larger errors of the ensemble individual models (Individual Error Term). Thus, the use of an ensemble made by ESNs with different architectures (ESN-DE ensemble) allows obtaining better performances by improving the models diversity.

An example of the improvement brought by the proposed method with respect to the RUL prediction accuracy is shown in

Table 4
Comparison among the RUL prediction performance provided by the considered approaches.

RUL prediction	Best ESN	TOPSIS local ensemble	ESN-DE static ensemble	ESN-DE local ensemble
CRA	0.49	0.59	0.55	0.66
$\overline{\alpha - \lambda}$ accuracy ($\alpha = 20\%$)	0.38	0.42	0.41	0.43

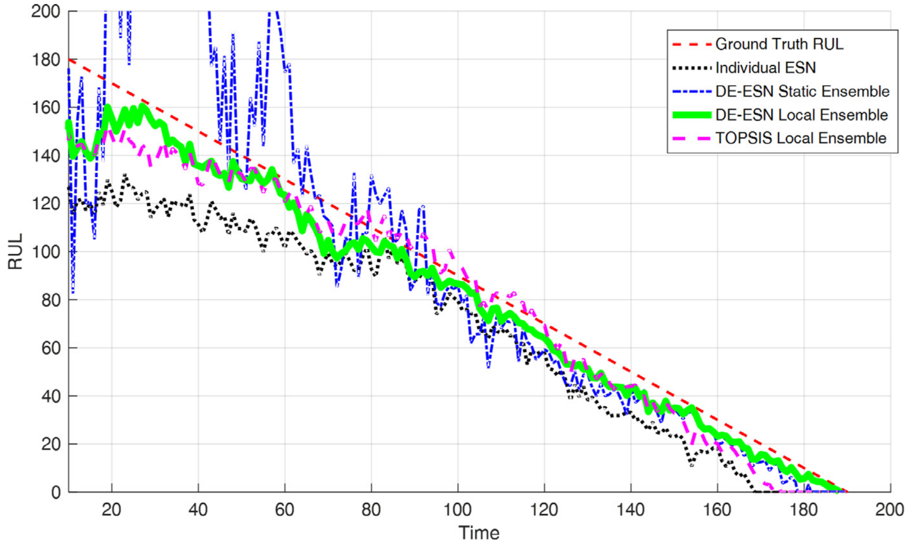


Fig. 12. Comparison between the RUL prediction provided by the best ESN individual model (dotted line), the DE-ESN static ensemble (dash-dotted line), the TOPSIS local ensemble (dashed line) and the proposed DE-ESN local ensemble (solid line).

Table 5
Ambiguity decomposition.

	TOPSIS local ensemble	ESN-DE local ensemble
Individual error term	843	1234
Ambiguity term	347	815

Table 6
Comparison among the PI estimation performance provided by MVE and ESN-based approach.

Prediction interval 90% confidence	Feedforward ANN-MVE	ESN-MVE
PICP	0.70	0.67
NMIPW	1.13	0.89

Fig. 12 for a given validation trajectory, where the local RUL prediction (solid line) is compared to that provided by both the best ESN model (dotted line) and the static ensemble (dash-dotted line).

4.2.2. *RUL prediction uncertainty: prediction interval estimation*

The ESN-MVE approach has been applied by dividing the Test Set into two subsets: the first, formed by 30 trajectories, has been used for training the ESN model; the second, formed by the remaining 20 trajectories, has been used for evaluating the ESN performance within a DE-optimization scheme aiming at maximizing the likelihood function of Eq. (10). The performance of the ESN-MVE approach has been compared to that of a feedforward ANN-MVE approach employing an optimal (by trial and error) time window width of 20. Table 6 reports the obtained PI performance in terms of PICP and NMPIW.

The feedforward ANN-MVE approach is seen to provide a PICP that is 3% larger than that provided by the proposed ESN-MVE approach, and its PIs are on average 24% larger than those provided by our method (NMIPW = 0.89). Similar coverage values are obtained when different PI confidence values are used (e.g. 80%, 95% and 98%). Fig. 13 shows the DE-ESN based local ensemble RUL prediction for one validation trajectory and the associated 90% PI confidence bounds provided by both the feedforward ANN-MVE and the ESN-MVE approaches.

Notice that the ESN-MVE PI width shows the expected behavior of being wider at the beginning of the component life, when there is more uncertainty on the failure time, and narrower at the end of life. With respect to the discrepancy between the expected coverage and that provided by the two methods (90% against 67% for ANN-MVE and 70% for ESN-MVE), notice that one of the drawback of the MVE method is that it assumes that the ensemble precisely estimates the true expected value of the RUL, i.e. the expected value of the RUL over all the possible degradation trajectories characterized by the current input vector observations. This assumption can be violated in practice because of a number of reasons, including the existence of a bias in the ensemble individual model predictions due to omission of important input quantities affecting the component RUL [42]. In these cases, it has been reported that the MVE PI estimation tends to be narrower than the true one, and, thus, the obtained coverage values are low [26]. Another possible cause of the low coverage value is the fact that the RUL prediction residuals may not be distributed according to a Gaussian distribution, as shown in the examples reported in Fig. 14. In such case, the estimation of the RUL percentiles obtained from the residual variance provided by the MVE approach may not be precise.

4.3. *Industrial case study*

We have considered an industrial case study regarding the prediction of the RUL of cutting knives used in the packaging industry. The time evolution of a physical quantity indirectly related to the knife degradation has been periodically measured with knife in operation. Such signal measurements have been taken only a limited number of times during each single run-to-failure degradation trajectory (10 trajectories in total). To deal with this type of data, we applied a pre-processing procedure followed by a feature extraction step, which has allowed the identification of two features

Please cite this article as: M. Rigamonti et al., Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.11.062

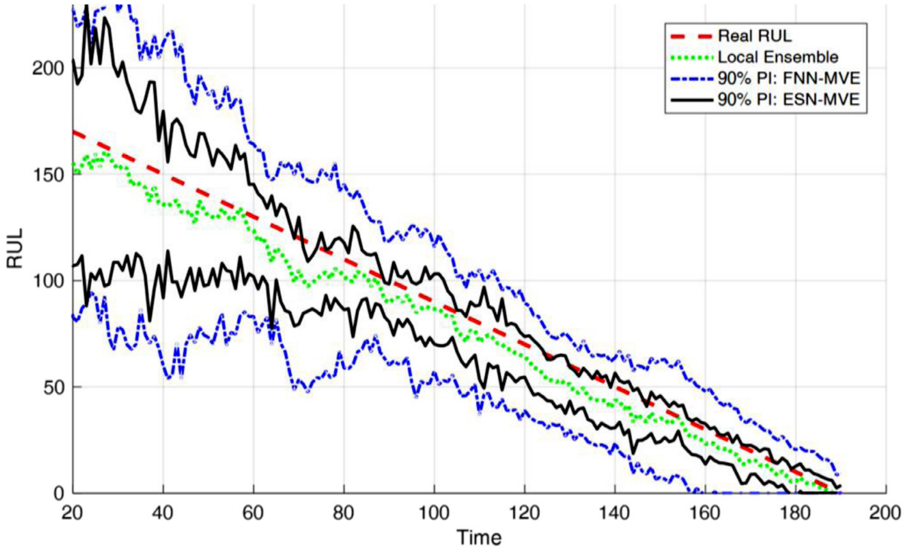


Fig. 13. Local Ensemble RUL prediction with the corresponding MVE and ESN 90% confidence bounds.

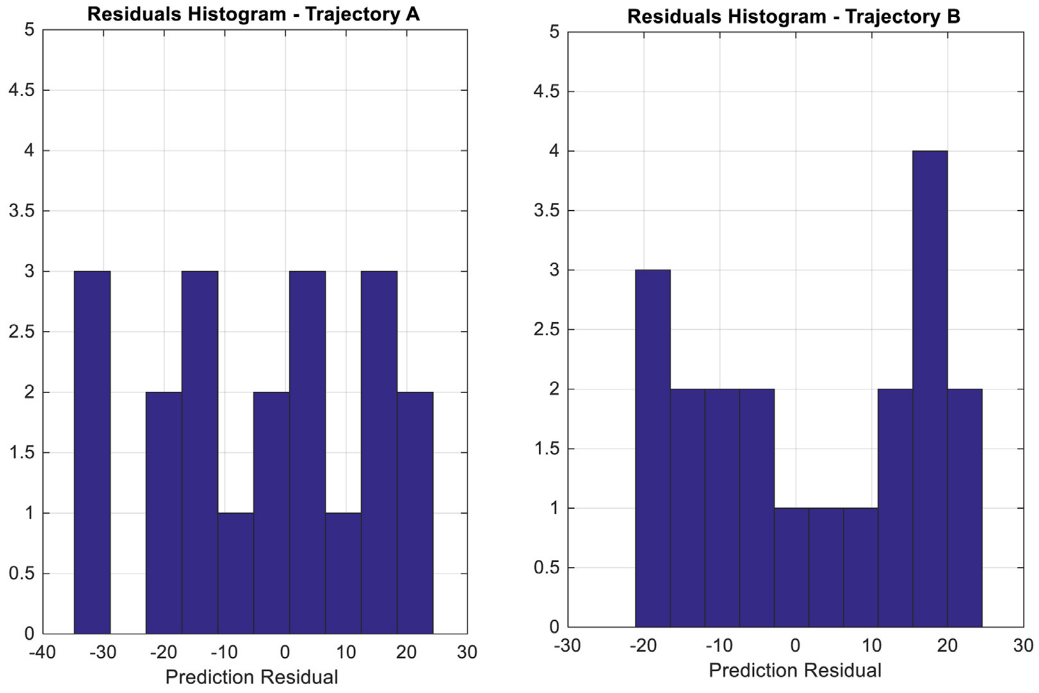


Fig. 14. Distribution of the local residuals in two different test trajectories.

Table 7
Comparison of the RUL prediction performances provided by the considered approaches.

RUL prediction	Best ESN	ESN-DE static ensemble	ESN-DE local ensemble
$\overline{\text{CRA}}$	0.58	0.60	0.70
$\overline{\alpha-\lambda}$ accuracy ($\alpha=20\%$)	0.41	0.32	0.44

significant for the prognostic task. More details on the dataset can be found in [19]; Fig. 15 shows the evolution of the two extracted features in the 10 available run-to-failure degradation trajectories.

A local ensemble of 50 ESNs, whose architectures have been optimized using the MO-DE approach, has been developed. Given the limited number, $R=10$, of run-to-failure trajectories, we have resorted to a leave-one-out procedure for evaluating the performance of the proposed method. Table 7 compares the performance of the proposed approach with those of the TOPSIS best ESN and of a static ensemble; the values reported in the Table are the averages over the 10 leave-one-out iterations of the method, each of them performed considering 5 training trajectories, 4 test trajectories and 1 validation trajectory. Notice that the proposed approach outperforms the others in terms of both $\overline{\text{CRA}}$ and $\overline{\alpha-\lambda}$ accuracy. Fig. 16 shows an example of obtained RUL prediction.

With respect to the RUL PI estimation, notice that the considered case study is characterized by the availability of a very limited number of degradation trajectories, each one formed by very few

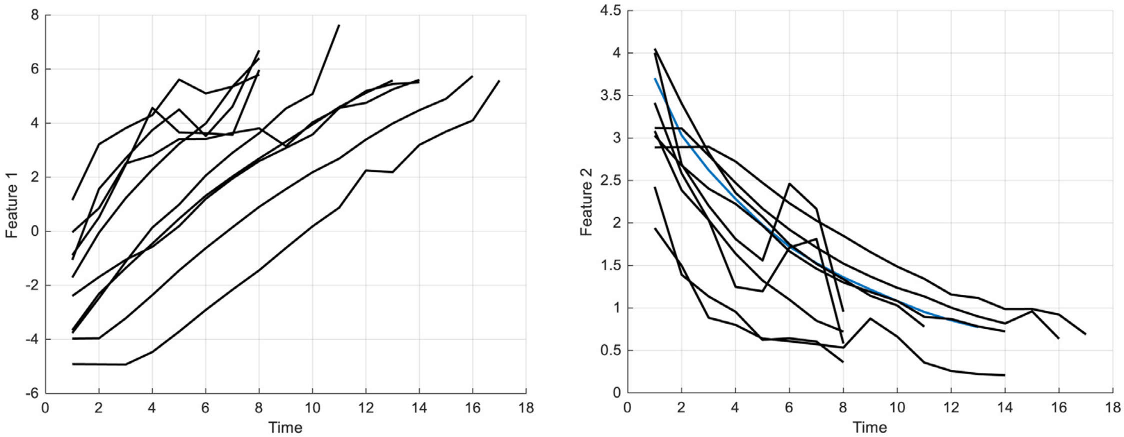


Fig. 15. Evolution of the two identified degradation features in the 10 available run-to-failure degradation trajectories (left— Feature 1; right—Feature 2).

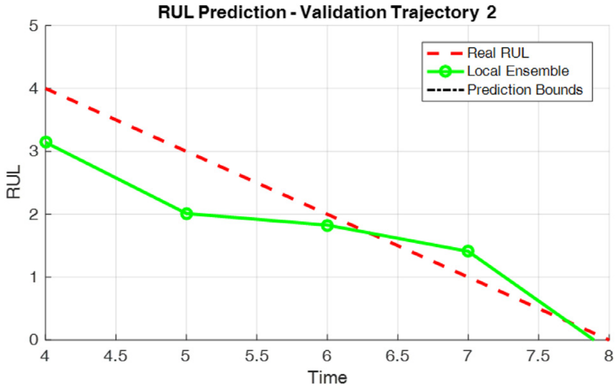


Fig. 16. Local ensemble RUL prediction for the validation trajectory number 2.

patterns (between 8 and 17). Therefore, the available 4 test trajectories do not allow developing an accurate ESN for the estimation of the prediction residuals.

5. Conclusion

In this work we have developed a local ensemble of ESNs for predicting the RUL of an industrial component and estimating the associated uncertainty. ESNs have been chosen due to their capability of catching the system dynamic behavior, combined with a very fast training procedure.

From the methodological point of view, the main novelties of the proposed approach are: (*i*) a new local aggregation procedure for combining the individual models outputs, which takes into account the individual ESNs memory capacities; (*ii*) the use of an ESN within the MVE approach for estimating the RUL prediction uncertainty.

From the practical point of view, a further novelty of the work is the application of an ensemble of ESNs, whose architectures have been optimized by using a MO-DE algorithm, which allows enhancing the diversity between the individual models of the prognostic ensemble.

The proposed approach has been verified with reference to a synthetic case study, an experimental case study concerning the RUL prediction of turbofan engines, taken from the NASA Ames Prognostics CoE Data Repository, and an industrial case study regarding the RUL prediction of industrial knives used in the packaging industry. With respect to the RUL prediction accuracy, the proposed method has outperformed an individual ESN predictor model and a classical static ensemble; with respect to the estimation of the RUL prediction uncertainty, the performance of the proposed approach has outperformed that of a feedforward ANN-MVE approach in the synthetic and in the turbofan engine cases, which are characterized by the availability of a large number of run-to-failure trajectories, whereas the performance is not satisfactory in the considered industrial case, which is characterized by the availability of too few data for training the dedicated ESN.

The satisfactory results obtained and the generality of the proposed approaches encourage the application to real industrial cases, aiming at improving maintenance for increasing the safety and the availability of the monitored systems while reducing the maintenance costs.

Appendix A. Prognostic metrics

The ***Cumulative Relative Accuracy*** is the normalized weighted sum of Relative Accuracy (RA) values, computed at specific t_λ time instances.

$$CRA = \frac{1}{|p_\lambda|}\sum_{i\in p_\lambda} RA_\lambda \tag{A.1}$$

where p_λ is the set of all time instants at which an RUL prediction is made for a degradation trajectory (in this work, all the available prediction points), $|p_\lambda|$ is the cardinality of the set, and RA_λ is defined as:

$$RA_\lambda = 1 - \frac{\left|\hat{RUL}_\lambda - RUL_\lambda\right|}{RUL_\lambda}, \tag{A.2}$$

where $\hat{RUL}_\lambda$ is the predicted RUL at time t_λ and RUL_λ is the groundtruth for RUL at time $t_{\lambda.}$ Values of RA_λ close to 1 indicate more accurate predictions.

α–λ Accuracy: The $\overline{\alpha - \lambda}$ accuracy evaluates the average fraction of predictions which falls within specified α-bounds, which are expressed as percentage of the actual RUL_λ at t_λ. It is defined as:

$$\overline{\alpha - \lambda} = \frac{1}{|p_\lambda|}\sum_{\lambda=1}^{|p_\lambda|} \alpha - \lambda_{t_\lambda} \tag{A.3}$$

where $\alpha - \lambda_{t_\lambda}$ is a binary value defined as:

$$\alpha - \lambda_{t_\lambda} = \begin{cases} 1 & \text{if } (1-\alpha)\cdot RUL_\lambda \le \hat{RUL}_\lambda \le (1+\alpha)\cdot RUL_\lambda \\ 0 & \text{Otherwise} \end{cases} \tag{A.4}$$

In the latter equation, λ refers to the $t_\lambda \in p_\lambda$ instant at which the prediction is performed and α is the percentage value defining

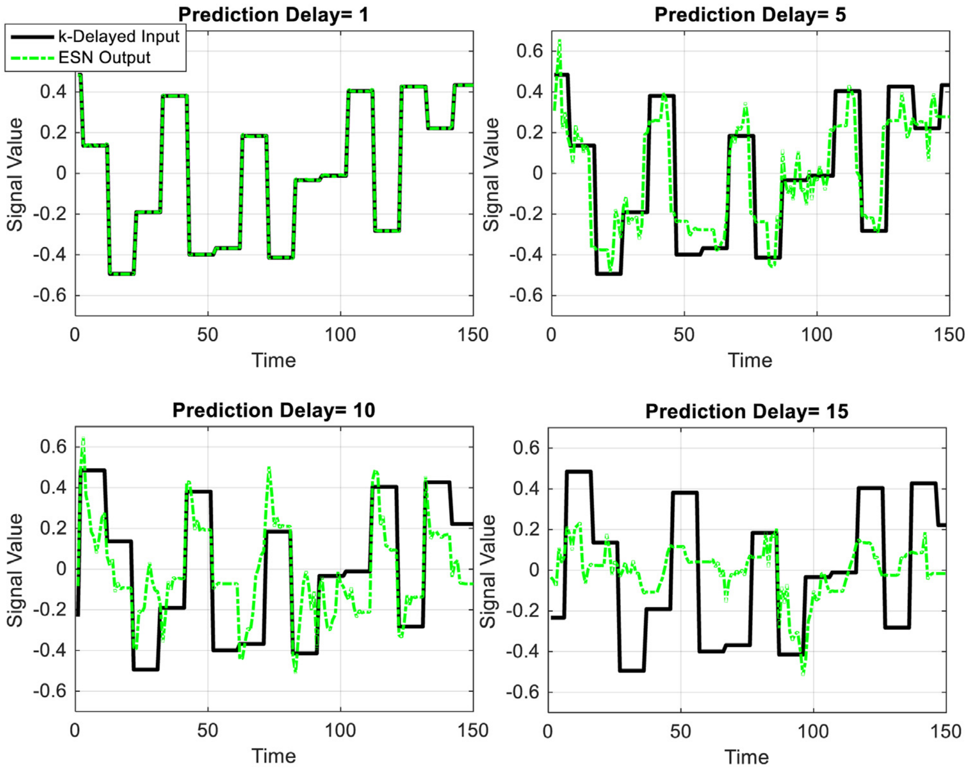


Fig. B.1. Example of STM computation for an ESN.

the acceptance confidence bounds, which in this work has been set equal to 20%.

Appendix B. ESN memory capacity computation

The Memory Capacity (MC) metric aims at quantifying the limit of the memory span and the precision of the prediction at which an ESN is able to generate as its output the delayed version *ν(t−k)* of a single-channel input *ν(t)*. In particular, in order to only focus on the memory property on the ESN, the single-channel input *ν(t)* has to be a sequence of i.i.d. values: by doing this, since there is no mathematical relation between the input and the delayed output, the only possibility for the network to provide as output the delayed input is that the network has been able to encapsulate the input into its internal states, i.e., it can remember it.

According to this, with respect to a certain delay *k*, [48] define the *k*-delay determination coefficient, i.e., MC_k as:

$$MC_k = \frac{\mathrm{cov}^2(\nu(t-k), y_k(t))}{\sigma^2(\nu(t))\sigma^2(y_k(t))} \tag{B.1}$$

where *ν(t−k)* represents the input delayed by *k* time steps and $y_k(t)$ represents the provided output, trained to be equal to the input delayed by *k* time steps. Basically, the determination coefficient is the squared correlation coefficient of two signals. It ranges between 0 and 1 and represents the fraction of variance explainable in one signal by the other [48]. In order to have an indication about the overall memory of the ESN, [48] defined also the overall STM capacity, which is a sum of the determination coefficients for all the possible time delay *k*:

$$MC = \sum_{k=1}^{\infty} MC_k = \sum_{k=1}^{\infty} \frac{\mathrm{cov}^2(\nu(t-k), y_k(t))}{\sigma^2(\nu(t))\sigma^2(y_k(t))} \tag{B.2}$$

For ease of comprehension, in Fig. B.1 we reported the ESN realigned outputs with respect to the same random single-channel input for different *k* values of the considered delay. It is possible to observe that the considered ESN is able to perfectly reproduce the input with a delay $k=1$; when the delay is $k=5$, the network is able to reproduce the behavior of the input, but the prediction accuracy starts to decrease; when the delay is $k=10$, the ESN is able to partially remember the general trend of the input, but the prediction accuracy is very bad; finally, when the delay is $k=15$, it is possible to observe that the network is not able to reproduce the input anymore.

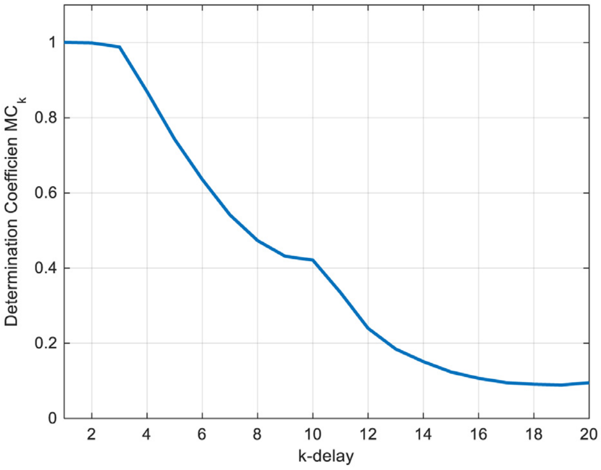


Fig. B.2. k-delay determination coefficient versus k-delay.

In Fig. B.2 the behavior of the $\mathrm{MC_k}$ is reported with respect to the *k* delay of the input to be recalled: it is possible to observe that the $\mathrm{MC_k}$ value is equal to 1 when the delay is $k=1$, i.e. it reflects that the delayed input is reproduced perfectly; on the contrary, when the delay is $k=15$, i.e. when the network is not able anymore to reproduce the delayed input, the value of the corresponding MC_k falls below 0.1, thus indicating very bad performance for that specific delay value.

According to Section 3.2.1, and in particular with respect to the local aggregation procedure for the ESN-based ensemble, since the Memory Capacity represents a direct measure of the memory property of a certain ESN, we consider that this measure can be indicative of the input time span on which the actual ESN output depends on. In particular, we assumed that the Memory Capacity of an ESN is representative of how much the output of the network depends on the previous input history, and thus we decided to consider a similarity time input window whose width (in time steps) was equal to the computed Memory Capacity. By doing this, if the ensemble is constituted by the same model which has been trained with different subsets of the training set, i.e., the typical bagging approach, [15], it is possible to consider the same similarity time window width for every model; on the contrary, if the ensemble is constituted by different ESNs with different memory properties, for each model we should take into account the proper width for the similarity time window. For example, for the ESN referring to the results shown in Fig. B.2, i.e. the TOPSIS solution of the synthetic case study of Section 4.1, the overall Memory Capacity is equal to 9.6: thus, according to our assumptions, the optimal width of the similarity time window to be considered is expected to be equal to 10.

References

[1] R. Ak, Y. Li, V. Vitelli, E. Zio, E.L. Droguett, C.M.C. Jacinto, NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment, Expert Syst. Appl. 40 (4) (2013) 1205–1212.
[2] R. Ak, V. Vitelli, E. Zio, An interval-valued neural network approach for uncertainty quantification in short-term wind speed prediction, IEEE Trans. Neural Netw. Learn. Syst. 26 (11) (2015) 2787–2800.
[3] H. Akaike, Fitting autoregressive models for prediction, Ann. Inst. Stat. Math. 21 (1) (1969) 243–247.
[4] A. Akusok, A. Gritsenko, Y. Miche, K.M. Björk, R. Nian, P. Lauren, A. Lendasse, Adding reliability to ELM forecasts by confidence intervals, Neurocomputing 219 (2017) 232–241.
[5] J.B. Ali, L. Saidi, A. Mouelhi, B. Chebel-Morello, F. Fnaiech, Linear feature selection and classification using PNN and SFAM neural networks for a nearly online diagnosis of bearing naturally progressing degradations, Eng. Appl. Artif. Intell. 42 (2015) 67–81.
[6] M. Assaad, R. Boné, H. Cardot, A new boosting algorithm for improved time-series forecasting with recurrent neural networks, Inf. Fusion 9 (1) (2008) 41–55.
[7] P. Baraldi, R. Razavi-Far, E. Zio, Classifier-ensemble incremental-learning procedure for nuclear transient identification at different operational conditions, Reliab. Eng. Syst. Saf. 96 (4) (2011) 480–488.
[8] P. Baraldi, F. Mangili, E. Zio, Ensemble of bootstrapped models for the prediction of the remaining useful life of a creeping turbine blade, in: 2012 IEEE Conference on Prognostics and Health Management (PHM), IEEE, 2012, pp. 1–8.
[9] P. Baraldi, F. Mangili, E. Zio, Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data, Reliab. Eng. Syst. Saf. 112 (2013) 94–108.
[10] P. Baraldi, F. Cadini, F. Mangili, E. Zio, Model-based and data-driven prognostics under different available information, Probab. Eng. Mech. 32 (2013) 66–79.
[11] P. Baraldi, M. Compare, S. Sauco, E. Zio, Ensemble neural network-based particle filtering for prognostics, Mech. Syst. Signal Process. 41 (1) (2013) 288–300.
[12] P. Baraldi, F. Mangili, E. Zio, A belief function theory based approach to combining different representation of uncertainty in prognostics, Inf. Sci. 303 (2015) 134–149.
[13] P. Baraldi, F. Cannarile, F. Di Maio, E. Zio, Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions, Eng. Appl. Artif. Intell. 56 (2016) 1–13.
[14] P.P. Bonissone, F. Xue, R. Subbu, Fast meta-models for local fusion of multiple predictive models, Appl. Soft Comput. J. 11 (2) (2011) 1529–1539.
[15] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[16] T. Brotherton, G. Jahns, J. Jacobs, D. Wroblewski, Prognosis of faults in gas turbine engines, in: Aerospace Conference Proceedings, 2000 IEEE, 6, IEEE, 2000, pp. 163–171.
[17] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Inf. Fusion 6 (1) (2005) 5–20.
[18] F. Cadini, E. Zio, N. Pedroni, Recurrent neural networks for dynamic reliability analysis, J. Polish Saf. Reliab. Assoc. 1 (2007).
[19] F. Cannarile, P. Baraldi, M. Compare, D. Borghi, L. Capelli, M. Cocconcelli, A. Lahrac, E. Zio, An unsupervised clustering method for assessing the degradation state of cutting tools in the packaging industry. Safety and reliability: theory and application, in: Proceedings of the European Safety and Reliability Conference, ESREL, 2017.
[20] S.J., Chen, C.L., Hwang, (1992). Fuzzy multiple attribute decision making: methods and applications. Springer- Verlag, Berlin.
[21] J. Cheng, S. Zhong, Q. Zhong, H. Zhu, Y. Du, Finite-time boundedness of state estimation for neural networks with time-varying delays, Neurocomputing 129 (2014) 257–264.
[22] L.H. Chiang, E. Russel, R. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag, London, 2001.
[23] M. Compare, L. Bellani, E. Zio, Availability model of a PHM-equipped component, IEEE Trans. Reliab. 66 (2) (2017) 487–501.
[24] M.J., Daigle, I., Roychoudhury, G., Biswas, X.D., Koutsoukos, A., Patterson-Hine, S., Poll, (2010). A comprehensive diagnosis methodology for complex hybrid systems: A case
[25] F. Di Maio, J. Hu, P. Tse, M. Pecht, K. Tsui, E. Zio, Ensemble-approaches for clustering health status of oil sand pumps, Expert Syst. Appl. 39 (5) (2012) 4847–4859.
[26] R. Dybowski, S. Roberts, Confidence intervals and prediction intervals for feed–forward neural networks, in: R. Dybowski, V. Gant (Eds.), Clinical Applications of Artificial Neural Networks, Cambridge Univ. Press, Cambridge, UK, 2000.
[27] A., Doucet, (1998). On sequential simulation-based methods for Bayesian filtering.
[28] A. Doucet, J.F.G De Freitas, N.J Gordon, Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.
[29] O. Fink, E. Zio, U. Weidmann, Predicting time series of railway speed restrictions with time-dependent machine learning techniques, Expert Syst. Appl. 40 (15) (2013) 6033–6040.
[30] D. Frederick, J. DeCastro, J. Litt, User's Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS), NASA/ARL, 2007 Technical Manual TM 2007-215026.
[31] S. Geisser, Predictive Inference, 55, CRC press, 1993.
[32] S. Ghosh-Dastidar, H. Adeli, Spiking neural networks, Int. J. Neural Syst. 19 (4) (2009) 295–308.
[33] K. Goebel, B. Saha, A. Saxena, A comparison of three data-driven techniques for prognostics, in: Sixty-Second Meeting of the Society for Machinery Failure Prevention Technology (mfpt), 2008, May, pp. 119–131.
[34] W. Gong, Z. Cai, Y. Wang, Repairing the crossover rate in adaptive differential evolution, Appl. Soft Comput. 15 (2014) 149–168.
[35] F.O. Heimes, Recurrent neural networks for remaining useful life estimation, in: 2008 International Conference on Prognostics and Health Management, PHM, 2008 art. n. 4711422.
[36] T. Heskes, Practical confidence and prediction, in: Proc. of the 1996 Conf. Advances in Neural Information Processing Systems, 9, 1997, p. 176.
[37] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, in: Proceedings of the USA National Academy of Sciences, 79, 1982, pp. 2554–2558.
[38] M.A. Hosen, A. Khosravi, S. Nahavandi, D. Creighton, Improving the quality of prediction intervals through optimal aggregation, IEEE Trans. Ind. Electron. 62 (7) (2015) 4420–4429.
[39] C. Hu, B.D. Youn, P. Wang, J.T. Yoon, Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life, Reliab. Eng. Syst. Saf. 103 (2012) 120–135.
[40] J.G. Hwang, A.A. Ding, Prediction intervals for artificial neural networks, J. Am. Stat. Assoc. 92 (438) (1997) 748–757.
[41] M.S. Kan, A.C.C. Tan, J. Mathew, A review on prognostic techniques for non-stationary and non-linear rotating systems, Mech. Syst. Signal Process. 62 (2015) 1–20.
[42] A. Khosravi, S. Nahavandi, D. Creighton, A.F. Atiya, Comprehensive review of neural network-based prediction intervals and new advances, IEEE Trans. Neural Netw. 22 (2011) 1341–1356.
[43] A. Khosravi, S. Nahavandi, D. Creighton, A.F. Atiya, Lower upper bound estimation method for construction of neural network-based prediction intervals, IEEE Trans. Neural Netw. 22 (2011) 337–346.
[44] G. Kitagawa, Non-Gaussian state—space modeling of nonstationary time series, J. Am. Stat. Assoc. 82 (400) (1987) 1032–1041.
[45] E.B. Kosmatopoulos, M.M. Polycarpou, M.A. Christodoulou, P.A. Ioannou, High-order neural network structures for identification of dynamical systems, IEEE Trans. Neural Netw. 6 (2) (1995) 422–431.
[46] O. Kramer, F. Gieseke, B. Satzger, Wind energy prediction and monitoring with neural computation, Neurocomputing 109 (2013) 84–93.
[47] H. Jaeger, The echo state approach to analyzing and training recurrent neural networks, German National Research Center for Information Technology, 2001 Technical Report GMD Report 148.
[48] H. Jaeger, Short term memory in echo state networks, GMD-Forschungszentrum Informationstechnik (2001).
[49] H. Jaeger, A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the Echo state network approach, German National Research Center for Information Technology, 2002 Technical Report GMD Report 159.
[50] V.M. Landassuri-Moreno, J.A. Bullinaria, Neural network ensembles for time series forecasting, in: Proceedings of the Eleventh Annual Conference on Genetic and Evolutionary Computation, ACM, 2009, July, pp. 1235–1242.
[51] Y. Liu, A. An, X. Huang, Boosting prediction accuracy on imbalanced datasets with SVM ensembles, in: PAKDD, 6, 2006, April, pp. 107–118.
[52] Y. Liu, Z. Zhang, J. Chen, Ensemble local kernel learning for online prediction of distributed product outputs in chemical processes, Chem. Eng. Sci. 137 (2015) 140–151.

[53] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. 3 (3) (2009) 127–149.
[54] S.X. Lun, X.S. Yao, H.Y. Qi, H.F. Hu, A novel model of leaky integrator echo state network for time-series prediction, Neurocomputing 159 (2015) 58–66.
[55] J. Luo, K.R. Pattipati, L. Qiao, S. Chigusa, Model-based prognostic techniques applied to a suspension system, IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans 38 (5) (2008) 1156–1168.
[56] A.K. Mahamad, S. Saon, T. Hiyama, Predicting remaining useful life of rotating machinery based artificial neural network, Comput. Math. Appl. 60 (4) (2010) 1078–1087.
[57] W. Maass, Networks of spiking neurons: the third generation of spiking neural network models, Neural Netw. 10 (9) (1997) 1659–1671.
[58] S. Morando, S. Jemei, R. Gouriveau, N. Zerhouni, D. Hissel, Fuel cells prognostics using echo state network, in: IECON Proceedings (Industrial Electronics Conference), 2013, pp. 1632–1637. 6699377.
[59] M.C. Moura, I.D. Lins, R.J. Ferreira, E.L. Droguett, C.M.C. Jacinto, Predictive maintenance policy for oil equipment in case of scaling through support vector machines, in: Proceedings of Advances in Safety, Reliability and Risk Management (Esrel 2011), Troyes, France, 2011, pp. 503–507.
[60] A.I. Moustapha, R.R. Selmic, Wireless sensor network modeling using modified recurrent neural networks: application to fault detection, IEEE Trans. Instrum. Meas. 57 (5) (2008) 981–988.
[61] D.A. Nix, A.S. Weigend, Estimating the mean and variance of the target probability distribution, in: 1994 IEEE International Conference on Neural Networks, 1994. IEEE World Congress on Computational Intelligence.„ Vol.1, 1994, June, pp. 55–60.
[62] S. Opricovic, G.-H. Tzeng, Compromise solution by MCDM methods: a comparative analysis of VIKOR and TOPSIS, Eur. J. Oper. Res. 156 (2) (2004) 445–455.
[63] E. Ott, B.R. Hunt, I. Szunyogh, A.V. Zimin, E.J. Kostelich, M. Corazza, J.A. Yorke, A local ensemble Kalman filter for atmospheric data assimilation, Tellus A 56 (5) (2004) 415–428.
[64] D. Pan, J.B. Liu, J. Cao, Remaining useful life estimation using an inverse Gaussian degradation model, Neurocomputing 185 (2016) 64–72.
[65] M. Pecht, Prognostics and Health Management of Electronics, John Wiley & Sons, Ltd, 2008.
[66] L. Peel, Data driven prognostics using a Kalman filter ensemble of neural network models, in: International Conference on Prognostics and Health Management, 2008. PHM 2008, IEEE, 2008, October, pp. 1–6.
[67] Y. Peng, H. Wang, J. Wang, D. Liu, X. Peng, A modified echo state network based remaining useful life estimation approach, in: PHM 2012—2012 IEEE Int. Conf. on Prognostics and Health Management: Enhancing Safety, Efficiency, Availability, and Effectiveness of Systems Through PHM Technology and Application, Conference Program, 2012 art. no. 6299524.
[68] K. Pipe, Practical prognostics for condition based maintenance, in: Proc. 2008 Int. Conf. Prognostics Health Manage., 2008, 2008, pp. 1–10.
[69] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[70] H. Quan, D. Srinivasan, A. Khosravi, Particle swarm optimization for construction of neural network-based prediction intervals, Neurocomputing 127 (2014) 172–180.
[71] M. Rana, I. Koprinska, A. Khosravi, V.G. Agelidis, Prediction intervals for electricity load forecasting using neural networks, in: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, 2013, pp. 1–8.
[72] C.E. Rasmussen, C.K. Williams, Gaussian Processes for Machine Learning, 1, MIT press, Cambridge, 2006.
[73] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, S. Poll, Echo state network for the remaining useful life prediction of a turbofan engine, in: PHME 2016 Conference Proceedings, 5, Bilbao, Spain, 2016 July 5th-8th2016.
[74] B. Saha, K. Goebel, J. Christophersen, Comparison of prognostic algorithms for estimating remaining useful life of batteries, Trans. Inst. Meas. Control 31 (3-4) (2009) 293–308.
[75] B. Samanta, K. Al-Balushi, Artificial neural network based fault diagnostics of rolling element bearings using time-domain features, Mech. Syst. Signal Process. 17 (2) (2003) 317–328.
[76] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 International Conference on Prognostics and Health Management, PHM 2008, 2008 art. no. 4711414.
[77] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, Evaluating algorithm performance metrics tailored for prognostics, in: Aerospace conference, 2009 IEEE, IEEE, 2009, March, pp. 1–13.
[78] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, Metrics for offline evaluation of prognostic performance, Int. J. Prognostics Health Manage. 1 (1) (2010) 4–23.
[79] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[80] M. Schwabacher, A survey of data-driven prognostics, in: Proceedings of the AIAA Infotech@ Aerospace Conference, 2005, September, pp. 1–5.
[81] P. Secchi, E. Zio, F. Di Maio, Quantifying uncertainties in the estimation of safety parameters by using bootstrapped artificial neural networks, Ann. Nucl. Energy 35 (12) (2008) 2338–2350.
[82] N.A. Shrivastava, B.K. Panigrahi, Point and prediction interval estimation for electricity markets with machine learning techniques and wavelet transforms, Neurocomputing 118 (2013) 301–310.
[83] C. Smith, Y. Jin, Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction, Neurocomputing 143 (2014) 302–311.
[84] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[85] Y. Su, S. Shan, X. Chen, W. Gao, Hierarchical ensemble of global and local classifiers for face recognition, IEEE Trans. Image Process. 18 (8) (2009) 1885–1896.
[86] P.W. Tse, D.P. Atherton, Prediction of machine deterioration using vibration based fault trends and recurrent neural networks, J. Vib. Acoust. Trans. ASME 121 (3) (1999) 355–362.
[87] R.M. Vandawaker, D.R. Jacques, J.K. Freels, Impact of prognostic uncertainty in system health monitoring, Int. J. Prognostics Health Manage. 6 (2015).
[88] W. Wang, D.Z. Li, J. Vrbanek, An evolving neuro-fuzzy technique for system state forecasting, Neurocomputing 87 (2012) 111–119.
[89] Y. Wei, J. Qiu, H.R. Karimi, M. Wang, Filtering design for two-dimensional Markovian jump systems with state-delays and deficient mode information, Inf. Sci. 269 (2014) 316–331.
[90] Y. Wei, J. Qiu, H.R. Karimi, M. Wang, Model approximation for two-dimensional Markovian jump systems with state-delays and imperfect mode information, Multidimens. Syst. Signal Process. 26 (3) (2015) 575–597.
[91] Y. Xing, E.W. Ma, K.L. Tsui, M. Pecht, An ensemble model for predicting the remaining useful performance of lithium-ion batteries, Microelectron. Reliab. 53 (6) (2013) 811–820.
[92] Z.B. Xu, H. Qiao, J. Peng, B. Zhang, A comparative study of two modeling approaches in neural networks, Neural Netw. 17 (1) (2004) 73–85.
[93] Z. Yang, P. Baraldi, E. Zio, A comparison between extreme learning machine and artificial neural network for remaining useful life prediction, in: Prognostics and System Health Management Conference (PHM-Chengdu), 2016, 2016, October, pp. 1–7.
[94] W. Yao, Z. Zeng, C. Lian, H. Tang, Ensembles of echo state networks for time series prediction, in: 2013 Sixth International Conference on Advanced Computational Intelligence (ICACI)„ IEEE, 2013, pp. 299–304.
[95] C. Zhang, H. Wei, L. Xie, Y. Shen, K. Zhang, Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework, Neurocomputing 205 (2016) 53–63.
[96] Q., Zhou, H., Chen, H., Zhao, G., Zhang, J., Yong, & J., Shen, (2016). A local field correlated and Monte Carlo based shallow neural network model for nonlinear time series prediction.
[97] E. Zio, A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes, IEEE Trans. Nucl. Sci. 53 (3) (2006) 1460–1478.
[98] E. Zio, F. Di Maio, A fuzzy similarity-based method for failure detection and recovery time estimation, Int. J. Perform. Eng. 6 (5) (2010) 407–424.

Marco Rigamonti (Ph.D. in Nuclear Engineering, Politecnico di Milano, May 2017) during his Ph.D. has been working on the development of data-driven methods for failure diagnostics and machine learning methods for failure prediction. He is co-author of 4 works accepted for publications on international journals.

Piero Baraldi (Ph.D. in nuclear engineering, Politecnico di Milano, 2006) is professor of Nuclear Engineering at the Department of Energy at the Politecnico di Milano. He has been Technical Committee Co-chair of the European Safety and Reliability Conference, ESREL2014, and Technical Programme Chair of the 2013 Prognostics and System Health Management Conference (PHM-2013). He is co-author of 2 books and more than 100 papers on international journals and proceedings of international conferences.

Enrico Zio (Nuclear Engineer Politecnico di Milano (1991); MSc in mechanical engineering, University of California, Los Angeles, UCLA (1995); Ph.D. in nuclear engineering, Politecnico di Milano (1995); Ph.D. in Probabilistic Risk Assessment, Massachusetts Institute of Technology, MIT (1998); Full professor, Politecnico di Milano (2005-); Director of the Chair on Complex Systems and the Energy Challenge at Ecole Centrale Paris and Supelec, Fondation Europeenne pour l'Energie Nouvelle—EdF (2010-present); Chairman of the European Safety and Reliability Association-ESRA (2010-present). He is co-author of seven books and more than 300 papers on international journals.

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems. Dr. Roychoudhury is a Senior Member of the IEEE and a member of the Prognostics and Health Management Society.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universitat Munchen, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is currently the Technical Area Lead of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence and is the Technical Lead for Real-Time Safety Modeling with NASA's SMART-NAS project. Prior to joining NASA in 2006, he was a Senior Research Scientist at General Electric Corporate Research and Development Center since 1997. He was also an Adjunct Professor of the Computer Science Department at Rensselaer Polytechnic Institute, Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has carried out applied research in the areas of real time monitoring, diagnostics, and prognostics and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems. Dr. Goebel holds 18 patents and has co-authored more than 300 technical papers in the field of Prognostics Health Management. He is currently member of the board of directors of the Prognostics and Health Management Society and Associate Editor of the International Journal of Prognostics and Health Management.

Scott Poll is the Deputy Lead of the Discovery and Systems Health (DaSH) Technical Area in the Intelligent Systems Division at NASA Ames Research Center. He also leads the Diagnostics and Prognostics Group within DaSH. He has conducted research to benchmark diagnostic algorithms in detecting and isolating faults in an experimental testbed. He has also conducted research in detection, isolation, accommodation, and situational awareness of aircraft flight control system failures. Prior to that, he was a researcher and assistant project director for a multi-phase wind tunnel test program of a cargo transport aircraft. He received the BSE degree in Aerospace Engineering from the University of Michigan, and the MS degree in Aeronautical Engineering from the California Institute of Technology.

Please cite this article as: M. Rigamonti et al., Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.11.062