Ensemble empirical mode decomposition and long short-term memory neural network for multi-step predictions of time series signals in nuclear power plants

Hoang-Phuong Nguyen, Piero Baraldi, Enrico Zio

Version of Record: https://www.sciencedirect.com/science/article/pii/S0306261920317281
Manuscript_088771175ae1cfe8ecd93b798d3f8878

Ensemble Empirical Mode Decomposition and Long Short-Term Memory Neural Network for Multi-Step Predictions of Time Series Signals in Nuclear Power Plants

Hoang-Phuong Nguyen[1,a], Piero Baraldi[2,b], Enrico Zio[3,b,c,d*]

Affiliation:

[a] Chair on System Science and the Energetic Challenge, CentraleSupélec, Université Paris-Saclay, France

[b] Department of Energy, Politecnico di Milano, Milano, Italy

[c] MINES ParisTech / PSL Université Paris, Centre de Recherche sur les Risques et les Crises (CRC), France

[d] Eminent Scholar, Department of Nuclear Engineering, Kyung Hee University, South Korea

Address:

[a] 9 rue Joliot-Curie, 91192 Gif-sur-Yvette, France

[b] Via La Masa 34, 20156 Milano, Italy

[c] 1 rue Claude Daunesse, 06904 Sophia Antipolis, France

[d] 26 Kyungheedae-ro, Hoegi-dong, Dongdaemun-gu, Seoul, South Korea

Email:

[1] hoang-phuong.nguyen@centralesupelec.fr

[2] piero.baraldi@polimi.it

[3] enrico.zio@mines-paristech.fr

Corresponding author:

[*] Enrico Zio, MINES ParisTech / PSL Université Paris, Centre de Recherche sur les Risques et les Crises (CRC), Sophia Antipolis, France

Email: enrico.zio@mines-paristech.fr

Abstract

We address the problem of multi-step ahead time series signal prediction in the energy industry, with the aim of improving maintenance planning and minimizing unexpected shutdowns. For this, we develop a novel method based on the combined use of Ensemble Empirical Mode Decomposition and Long Short-Term Memory neural network. Ensemble Empirical Mode Decomposition decomposes the time series into a set of Intrinsic Mode Function components which facilitate the prediction task by effectively describing the system dynamics. Then, Long Short-Term Memory neural network models perform the multi-step ahead prediction of the individual Ensemble Empirical Mode Decomposition components and the obtained predictions are aggregated to reconstruct the time series. A Tree-structured Parzen Estimator algorithm is employed for the optimization of the hyperparameters of the Long Short-Term Memory neural network. The proposed method is validated by considering various long-term prediction horizons of real time series data acquired from Reactor Coolant Pumps of Nuclear Power Plants. The results show the superior performance of the proposed method with respect to alternative state of the art methods.

Keywords

predictive maintenance, prognostics, multi-step ahead prediction, ensemble empirical mode decomposition, long short-term memory recurrent neural network, reactor coolant pump.

Nomenclature

Abbreviations

AM-FM	Amplitude-Modulated-Frequency-Modulated
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
DNBR	Departure from Nucleate Boiling Ratio
EEMD	Ensemble Empirical Mode Decomposition
EI	Expected Improvement
EMD	Empirical Mode Decomposition
ES	Exponential Smoothing
ESN	Echo State Network
EU	European Union
FNN	False Nearest Neighbor
IMF	Intrinsic Mode Function
IRENA	International Renewable Energy Agency
LCOE	Levelized Cost Of Electricity
LOCA	Loss-Of-Coolant Accident
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MIMO	Multi-Input Multi-Output
MSE	Mean Square Error
NPP	Nuclear Power Plant
O&M	Operation and Maintenance
PHM	Prognostics and Health Management
PWR	Pressurized Water Reactor
RCP	Reactor Coolant Pump
RCS	Reactor Coolant System
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SMBO	Sequential Model-based Bayesian Optimization
SVR	Support Vector Regression
TPE	Tree-structured Parzen Estimator
WPD	Wavelet Packet Decomposition

Symbols	
$c_j(t)$	EEMD component at the *j*th sifting iteration
C_t	output of the LSTM cell state at time *t*
$\tilde{C}_t$	new values of the LSTM cell state at time *t*
d	embedding dimension
f	prediction model
f_t	output of the LSTM forget gate at time *t*
h_t	output of the LSTM memory block at time *t*
$G(\theta,\alpha)$	optimization function of the LSTM network
h	prediction horizon
i_t	output of the LSTM input gate at time *t*
$IMF_i(t)$	*i*th decomposed IMF at time *t*
$\overline{IMF}_i(t)$	*i*th ensemble IMF at time *t*
J	number of the EEMD noise realizations
$L_j(t)$	lower envelope of the decomposed component at the *j*th sifting iteration
$m_j(t)$	envelope mean of the decomposed component at the *j*th sifting iteration
N	number of samples
N_c	number of IMF components
N_{epoch}	number of LSTM training epochs
N_{max_epoch}	maximum number of the LSTM training epochs
$N_{patience}$	number of epochs with no improvement after which training will be stopped
N_{init}	number of TPE startup iterations
N_{opt}	number of TPE iterations
o_t	output of the LSTM output gate at time *t*
$\Pr_B(\theta)$	probability that the hyperparameter set θ belongs to the bad group
$\Pr_G(\theta)$	probability that the hyperparameter set θ belongs to the good group
$r_i(t)$	*i*th decomposed residue at time *t*
$SD(j)$	stopping criterion value at the *j*th sifting iteration
t	time instance
$U_j(t)$	upper envelope of the decomposed component at the *j*th sifting iteration
w_t^j	*j*th realization of white Gaussian noise

(W_C, b_C)	weight and bias of the LSTM cell state, respectively
(W_f, b_f)	weight and bias of the LSTM forget gate, respectively
(W_i, b_i)	weight and bias of the LSTM input gate, respectively
(W_o, b_o)	weight and bias of the LSTM output gate, respectively
x_t	actual value at time *t*
$\hat{x}_t$	predicted value at time *t*
X_t	time series collected up to time *t*
$y^{(i)}$	fitness score of the *i*th hyperparameter set in the search space
y^*	fitness score threshold for classifying hyperparameter groups
α	learning rate of the LSTM network
ε	stopping criterion threshold of the EMD sifting process
θ	hyperparameter set
θ^*	optimal hyperparameter set
$\phi(x)$	activation function of the LSTM network
σ	sigmoidal layer function used in the LSTM repeating memory modules
σ_N	noise standard deviation used in EEMD

1 Introduction

Since the early 1950s, maintenance engineering has played a fundamental role for maintaining the reliability, availability and safety of energy production plants components and systems, and reducing their life cycle costs [1]. Nowadays, the rapid growth of information technologies, along with the massive increase in information and data availability, has enabled the development and application of Prognostics and Health Management (PHM). PHM is a field of research and application, which utilizes past and present information to detect at an early stage the degradation of industrial components and systems, diagnose the fault root causes and predict the future evolution of the degradation and the Remaining Useful Life (RUL) [2]. Accurate and reliable predictions provided by PHM allow planning maintenance actions at the most convenient and inexpensive time, thus reducing the operation and maintenance (O&M) cost and energy production loss from unplanned downtime [3]. In 2017, the International Renewable Energy Agency (IRENA) reported that the O&M cost in

Germany and United Kingdom accounted for 20-25% of the levelized cost of electricity (LCOE) [4]. The importance of PHM in the reduction of the O&M cost is witnessed by the estimation of the investments in software platforms in support of predictive maintenance within the European Union (EU) energy industry, which is estimated to reach 0.2 billion euros in 2030 [5].

Several factors need to be accounted for when developing an effective PHM, such as the specific requirements of the application, the knowledge and data available on the components and systems degradation and failure, and the prediction horizon, i.e. how far into the future the model should predict and with what accuracy [6]. In safety-critical applications, such as those typically encountered in the nuclear industry, components and systems are designed to guarantee very high reliability levels given the potentially catastrophic consequences of their failures. Therefore, given the long-term horizons of the degradation processes, prognostics is called to accurately predict components and systems behaviors multi-step ahead. This is of paramount importance in the nuclear industry where maintenance interventions of some critical components should be planned well in advance given the impossibility of performing some of them during plant operation. Also, long-term predictions of the components degradation are needed to decide whether a component can safely operate until the next planned plant outage, which can involve predictions over time horizons of months [7]. Despite its importance, multi-step ahead prediction remains a difficult task of PHM because prediction uncertainty tends to exponentially increase with the time horizon of the prediction. This is mainly caused by the intrinsic stochasticity of the degradation process, the accumulation of the prognostic model errors and the difficulty of predicting the component operating conditions, which can largely influence the degradation process [6]. Large prediction uncertainty has limited prognostics in nuclear applications to one-step ahead predictions of the departure from nucleate boiling ratio (DNBR) distribution in a hot fuel rod [8], the leak flow rate in loss-of-coolant accidents (LOCAs) [9], the water level in steam generators [10] and pressurizer [11] and Nuclear Power Plants (NPPs) parameters in abnormal conditions [7]. In this context, this work develops a prognostic method specifically designed to deal with multi-step ahead predictions for practical O&M applications in NPPs to the benefit of energy production and economy.

In general, multi-step ahead prediction models can be classified as statistical or machine learning approaches [12]. Statistical approaches, such as Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ES), attempt to model the data autocorrelation structure and make predictions assuming a linear dependence between future and past data [13]. Because of this assumption, statistical approaches are not the appropriate choice for complex real-world systems, such as nuclear power plants which typically exhibit nonlinear and nonstationary behaviors. Alternatively, machine learning approaches have been shown able to automatically learn arbitrary complex mappings between inputs and outputs directly from historical data and achieve accurate predictions without the need of prespecifying the model form [14]. The most widely used machine learning approaches for multi-step ahead predictions are Support Vector Regression (SVR) [15], Artificial Neural Network (ANN) [16], Neuro-Fuzzy [17] and Recurrent Neural Network (RNN) [18]. Recently, the use of Long Short-Term Memory (LSTM) has been proposed to improve the performance of conventional RNN in dealing with long-term predictions [19]. An LSTM is based on a series of memory cells recurrently connected through layers to capture and retain the data long-term dependencies, thus enhancing the network capability in learning and predicting multi-step ahead into the future. Successful applications of LSTM for multi-step ahead prediction have been reported in many different fields, such as the forecasting of wind speed [20], solar energy [21], air quality [22], stock market [23], electricity and gas demand [24], and oil and petroleum production [25].

A problem typically encountered in the development of multi-step ahead prediction models is the data complexity, i.e. time series collected from real-world systems contains at the same time multiple and very different dynamic trends superposed on each other. Attempting to simultaneously capture various trends in the data can lead to unsatisfactory prediction performance when the time horizon of the prediction increases [26]. This issue has been recently addressed by using hybrid prediction models which take advantage of the strength of ensembles of different individual models. For example, Moshkbar-Bakhshayesh and Ghofrani [7] have presented a hybrid framework integrating ARIMA and ANN for separately dealing with linear and nonlinear components of the time series trends. Similarly, Buyuksahin and Ertekin [27] have presented a comparison among hybrid ARIMA-

ANN models and individual models considering different applications. Their experimental results show that hybrid models are much more accurate in capturing different data structures than individual models, and, thus, allow improving prediction performance. Li et al. [28] have developed a decomposition-based hybrid model, which combines wavelet packet decomposition (WPD) and ANN for the prediction of wind speed data over a 9-step ahead horizon. The basic idea behind decomposition-based hybrid models is to break down time-series data into several components, which are characterized by more linear and more stationary trends, and, therefore, are easier to be individually predicted. The work demonstrates the superior performance of the decomposition-based hybrid approach with respect to conventional models in long-term horizon predictions. A comprehensive analysis on hybrid approaches for the applications concerning multi-step ahead prediction can be found in [29].

In this work, a hybrid model combining Ensemble Empirical Mode Decomposition (EEMD) and LSTM networks with an automatic hyperparameter optimization is proposed for multi-step ahead time series prediction for application to the energy industry. EEMD is a self-adaptive decomposition technique specifically tailored for analyzing nonlinear and nonstationary data [30]. It is employed to increase the prediction performance by decomposing original time series into features representing separate spectral components, which are easier to predict. Then, multiple LSTM models are applied to the obtained features to predict their multi-step ahead behaviors. The obtained predictions are aggregated to reconstruct the multi-step ahead prediction of the original time series. A Multi-Input Multi-Output (MIMO) strategy is employed to avoid the error accumulation problem in long-term predictions. The problem of automatic hyperparameter optimization is addressed by integrating a Tree-structured Parzen Estimator (TPE) algorithm within the LSTM models.

In summary, the main methodological contributions of this work are:

(1) The novel multi-step ahead prediction method based on the combination of the EEMD decomposition algorithm and the LSTM neural networks.

(2) The integration of an automatic hyperparameter optimization based on a TPE optimization algorithm and a *k*-fold cross-validation technique within the LSTM models.

A case study based on real time-series datasets acquired from NPPs is carried out to validate the proposed modeling framework. To the authors' knowledge, this is the first study using a hybrid framework combining EEMD and LSTM for addressing the multi-step ahead prediction problem of NPP signals.

The rest of the paper is organized as follows. Section II introduces the EEMD decomposition technique, the LSTM neural network and the TPE hyperparameter optimization. Section III focuses on describing the proposed method for multi-step ahead prediction. The details of the practical case study are presented in Section IV and the obtained results are discussed in Section V. Finally, Section VI concludes the work.

2 Related methodologies

2.1 Signal decomposition methods

This Section presents methods for signal decomposition based on empirical mode decomposition (EMD). Section 2.1.1 and 2.1.2 are dedicated to the original EMD and the EEMD algorithms, respectively.

2.1.1 Empirical Mode Decomposition (EMD)

EMD was proposed by Huang et al. [31] as an adaptive signal processing method for decomposing nonlinear and nonstationary time-series into separate spectral modes called Intrinsic Mode Functions (IMFs). Specifically, IMFs are Amplitude-Modulated-Frequency-Modulated (AM-FM) signals representing certain frequency bands of the original time series from high-frequency (first IMF) to low-frequency bands (last IMF) [32]. Each IMF satisfies the following properties: 1) the number of zero-crossings and local extrema differ at most by one; 2) the mean value of the upper and lower envelopes of an IMF, identified by local maxima and minima, is zero at any time. The main advantage of EMD with respect to other decomposition methods such as WPD is that the time series is decomposed into a finite set of IMFs and a monotonic residue by an adaptive decomposition process (also known as the sifting process), without any need of predefining basic functions

(Algorithm 1) [33].

Algorithm 1. EMD decomposition pseudo code.

Input: Time series $X_t = \{x_1, x_2, ..., x_t\}$, threshold of the stopping criterion ε (typically set in the range [0.2; 0.3] [31]).

Output: A set of N_c IMFs $\{IMF_i(\tau)\}$ $(i = 1, 2, ..., N_c; \tau = 1, 2, ..., t)$ and a residue $r_{N_c}(t)$.

Decomposition process:

1. Initialize the index $i = 1$ and residue $r_0(t) = X_t$.
2. Extract $IMF_i(t)$:
   a. Assign the *i*th component equal to the previous residue: $c_j(t) = r_{i-1}(t)$, with the sifting iteration index *j* set equal to 1.
   b. Determine the local maxima and minima of $c_j(t)$ and use a cubic spline interpolation to compute their upper and lower envelopes, $U_j(t)$ and $L_j(t)$, respectively.
   c. Compute the envelope mean:
   $$m_j(t) = \left[U_j(t) + L_j(t)\right] / 2$$
   d. Generate the new component $c_{j+1}(t)$ of the next sifting iteration:
   $$c_{j+1}(t) = c_j(t) - m_j(t)$$
   e. Compute the squared difference between two consecutive siftings as follows:
   $$SD(j) = \sum_{l=1}^{t} \frac{\left|c_{j+1}(l) - c_j(l)\right|^2}{\left|c_j(l)\right|^2}$$
   f. If the stopping criterion $SD(j) < \varepsilon$ is verified, the new $IMF_i(t) = c_{j+1}(t)$ is defined and go to Step 3; otherwise, update $j = j + 1$ and repeat a sifting iteration by performing Steps 2.b) – 2.f).
3. Update the residue as follows:
   $$r_i(t) = r_i(t) - IMF_i(t)$$
4. If the number of extrema of $r_i(t)$ is less than 2 or $r_i(t)$ becomes monotonic, the decomposition process is terminated; otherwise, repeat Step 2 with $i = i + 1$.

The sifting process decomposes the original time series X_t into:

$$X_t = \sum_{i=1}^{N_c} IMF_i(t) + r_{N_c}(t) \tag{5}$$

2.1.2 Ensemble Empirical Mode Decomposition (EEMD)

Limitations of EMD are that different oscillation components may coexist in a single IMF and very similar oscillations may reside in different IMFs, which are called mode-mixing [34]. To address these problems, EEMD has been developed [30]. The key idea of EEMD is to use an ensemble of IMFs obtained by performing EMD over several different realizations of the original time series obtained by adding to it a white Gaussian noise. The effect of adding a white Gaussian noise reduces the mode-mixing problem by populating the whole time-frequency space and utilizing the dyadic filter bank behavior of EMD [32]. The EEMD algorithm is described in Algorithm 2.

Algorithm 2. EEMD decomposition pseudo code.

Input: Time series $X_t = \{x_1, x_2, ..., x_t\}$.

Output: A set of ensemble IMFs $\{\overline{IMF_i}(\tau)\}$ $(i = 1, 2, ..., N_c; \tau = 1, 2, ..., t)$.

Decomposition process:

1. Generate the noisy time series:

   $$X_t^j = X_t + w_t^j, \ j = 1, 2, ..., J$$

   where w_t^j are realizations of white Gaussian noise and *J* is the predefined number of noise realizations.

2. Apply Algorithm 1 to each time series X_t^j and obtain the corresponding $\{IMF_i^j(t)\}$, $i = 1, 2, ..., N_c$, $j = 1, 2, ..., J$.

3. Compute $\overline{IMF_i}(t)$ by averaging the $IMF_i^j(t)$:

   $$\overline{IMF_i}(t) = \frac{1}{J}\sum_{j=1}^{J} IMF_i^j(t)$$

The EEMD decomposes the original time series X_t into N_c IMFs and a residue:

$$X_t = \sum_{i=1}^{N_c} \overline{IMF}_i(t) + r_{N_c}(t) \tag{8}$$

2.2 Long Short-Term Memory (LSTM)

LSTM is a type of RNN which has been developed to address the problems of the vanishing or exploding gradient that are typically encountered when training traditional RNNs in case of long-term dependencies in the time series [19]. An LSTM network consists of a chain of repeating memory

modules (Fig. 1).

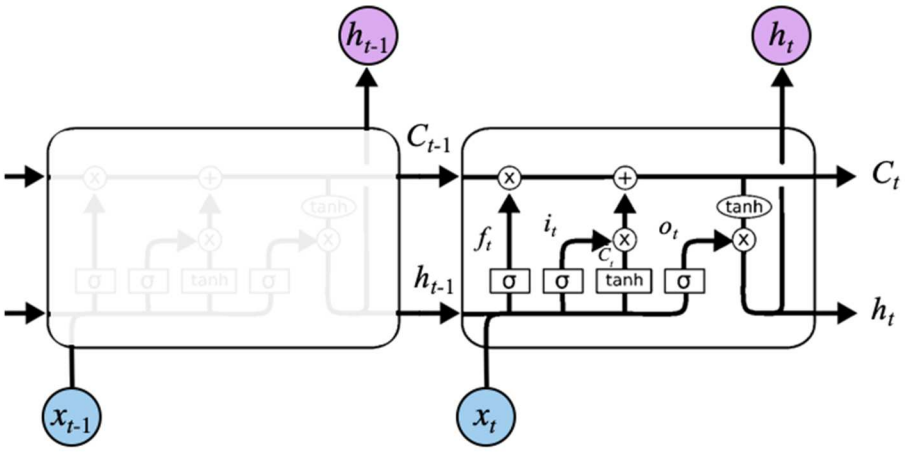


Fig. 1. Representation of a LSTM repeating memory module [35].

In each memory module, a cell state C_t, which is composed of a sigmoidal layer function σ and a pointwise multiplication operation, controls the network information using the forget, input and output gates. At time *t* when a new observation x_t is fed to the network, the forget gate decides to keep or remove the information of the preceding memory block output h_{t-1}. The ouput of the forget gate is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{9}$$

where W_f and b_f are the input weights and bias of the forget gate, respectively, and " . " denotes the multiplication operation. The input gate determines whether x_t is stored in the cell state C_t:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{10}$$

where W_i and b_i are the input weights and bias of the input gate, respectively. A tanh layer function is used to generate a new information vector $\tilde{C}_t$ to be added to C_t:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{11}$$

where W_c and b_c are the input weights and bias of the tanh layer function of C_t, respectively. The tanh activation function is used to normalize the values flowing through the network in the range [-1; 1]. The outputs of the forget and input gates and of the tanh layer function are used to update the cell

state C_t:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{12}$$

Finally, the output of the memory block h_t is generated by using the output gate and another tanh layer:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{13}$$

$$h_t = o_t * \tanh(C_t) \tag{14}$$

where W_o and b_o are the input weights and bias of the output gate, respectively.

2.3 Tree-structured Parzen Estimator (TPE) optimization

Automatic hyperparameter optimization plays a fundamental role in the development of machine learning models, especially when deep neural networks such as LSTM [36] are used. It allows reducing the human effort necessary to develop the model and improving the network performance by selecting hyperparameter values optimal for the target application at hand [37], [38]. In this study, we apply Tree-structured Parzen Estimator (TPE) [39], which is a Sequential Model-based Bayesian Optimization (SMBO) algorithm, to automatically select the hyperparameters of the LSTM model. The fitness function of our optimization problem is the Root Mean Square Error (RMSE) of the LSTM:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2}, \tag{15}$$

where N is the number of observations and x and $\hat{x}$ are the time series true and predicted values, respectively.

The TPE optimization process requires a number of function evaluations lower than other optimization techniques such as grid and random search, which means that it can achieve a faster convergence to the optimum. Also, differently from SMBO, it allows optimizing categorical and conditional hyperparameters, providing a wider range of hyperparameter choices [39].

The key idea of TPE is to use the Parzen-window density estimation (also known as kernel

density estimation) for building probability density functions in the hyperparameter search space. More specifically, each sample defines a Gaussian distribution in the hyperparameter space with a mean equal to the hyperparameter value and a properly set standard deviation. At the start-up iterations, a random search is performed to initialize the distributions by sampling the response surface $\{\theta^{(i)}, y^{(i)}\}$ $(i = 1, 2, ..., N_{init})$, where θ denotes the hyperparameter set and y is the corresponding value of the response surface (i.e. the fitness score) and N_{init} is the number of start-up iterations. Then, the hyperparameter space is divided into two groups, namely *good* and *bad* samples with respect to a threshold value $y*$ of the fitness score. The two groups are defined by the probability distributions Pr_G and Pr_B of the hyperparameter set θ:

$$p(\theta \mid y) = \begin{cases} \mathrm{Pr}_G(\theta) & if \ \ y < y* \\ \mathrm{Pr}_B(\theta) & if \ \ y \geq y* \end{cases} \tag{16}$$

Then, the expected improvement (EI) is computed at each iteration:

$$EI(\theta) = \frac{\mathrm{Pr}_G(\theta)}{\mathrm{Pr}_B(\theta)} \tag{17}$$

And the hyperparameter configuration $\theta*$ which maximizes EI is chosen. Therefore, TPE selects the optimal hyperparameters based on a set of best observations and their distributions, not only the best one. Fig. 2 describes the overall flowchart of the TPE algorithm, where N_{opt} denotes the number of TPE iterations.

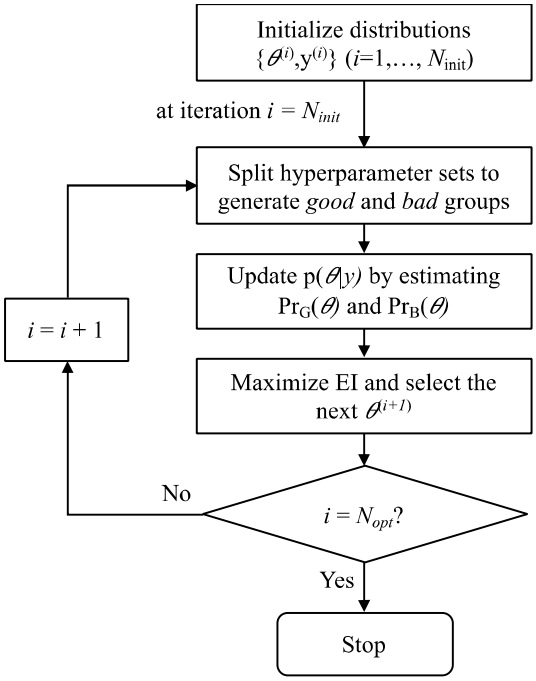


Fig. 2. Flowchart of the TPE optimization procedure.

2.4 Multi-step ahead prediction strategies

Multi-step ahead prediction aims at estimating the H next values of a time series $\{\hat{x}_{t+h}\}, h \in [1, H]$, given the current and previous observations $\{x_1, x_2, ..., x_t\}$. Three strategies are typically considered: recursive, direct and MIMO [6], [40].

2.4.1 Recursive prediction

It is based on the recursive use of a single model performing the one-step ahead prediction of the time series. In other words, being f_R the one-step ahead prediction model and d the embedding dimension, the multi-step ahead predictions are:

$$\begin{aligned} \hat{x}_{t+1} &= f_R(x_t, x_{t-1}, ..., x_{t-d+1}) \\ \hat{x}_{t+2} &= f_R(\hat{x}_{t+1}, x_t, ..., x_{t-d+2}) \\ &\vdots \qquad\qquad \vdots \\ \hat{x}_{t+H} &= f_R(\hat{x}_{t+H-1}, \hat{x}_{t+H-2}, ..., \hat{x}_{t+H-d+1}) \end{aligned} \tag{18}$$

One advantage of the recursive strategy is that the computational effort needed for its development is smaller than that of the other strategies, since it requires to train a single one-output prediction model. However, since intermediate predictions are used as inputs for predicting the next

values, the prediction accuracy decreases as the length of the time horizon increases due to error accumulation [40].

2.4.2 Direct prediction

It is based on H different models $f_{D,h}$, $h \in [1, H]$, each one dedicated to the prediction of the time series value $\hat{x}_{t+h}$ [41]:

$$\begin{aligned} \hat{x}_{t+1} &= f_{D,1}(x_t, x_{t-1}, ..., x_{t-d+1}) \\ \hat{x}_{t+2} &= f_{D,2}(x_t, x_{t-1}, ..., x_{t-d+1}) \\ &\vdots \qquad\qquad \vdots \\ \hat{x}_{t+H} &= f_{D,H}(x_t, x_{t-1}, ..., x_{t-d+1}) \end{aligned} \tag{19}$$

Since each model does not receive in input predictions, the accumulation of the prediction errors is avoided. The two main limitations of this strategy are: 1) the large computational cost associated to the training of the H models; 2) it performs the predictions at different horizons independently, without considering their temporal dependencies [40].

2.4.3 MIMO prediction

It is based on a single model f_{MIMO} which provides in output a vector formed by the predictions at the different horizons $h \in [1, H]$ [42]:

$$\{\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H}\} = f_{MIMO}(x_t, x_{t-1}, ..., x_{t-d+1}) \tag{20}$$

Since the loss function minimized during the training process simultaneously considers the prediction errors at several horizons, the MIMO strategy is able to preserve the temporal dependencies in the time series. Also, the problem of the accumulation of the prediction errors of the recursive strategy is avoided [6].

3 The multi-step ahead prediction method

The proposed prediction method is composed of two main parts: decomposition and multi-step ahead prediction (Fig. 3). The input is a time series $X_t = \{x_\tau\}\,(\tau = 1, 2, ..., t)$, which is formed by signal measurements collected from a component and provides in output the multi-step ahead predictions

$\{\hat{x}_{t+h}\}(h=1,2,...,H)$, where h represents the prediction horizon. The details of the method are described in the following Sections.

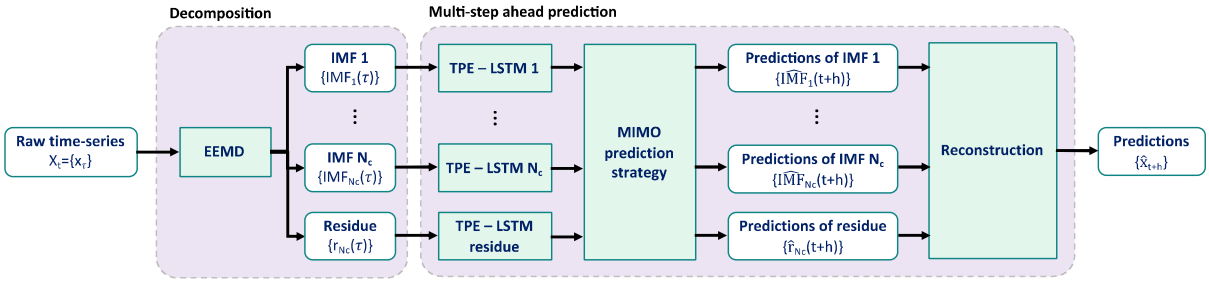
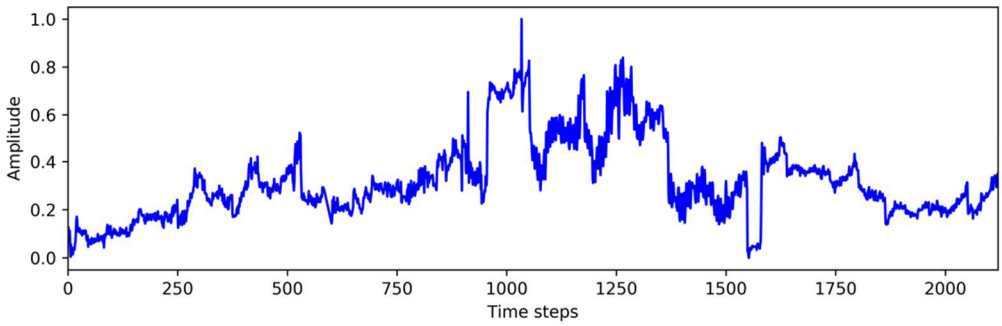


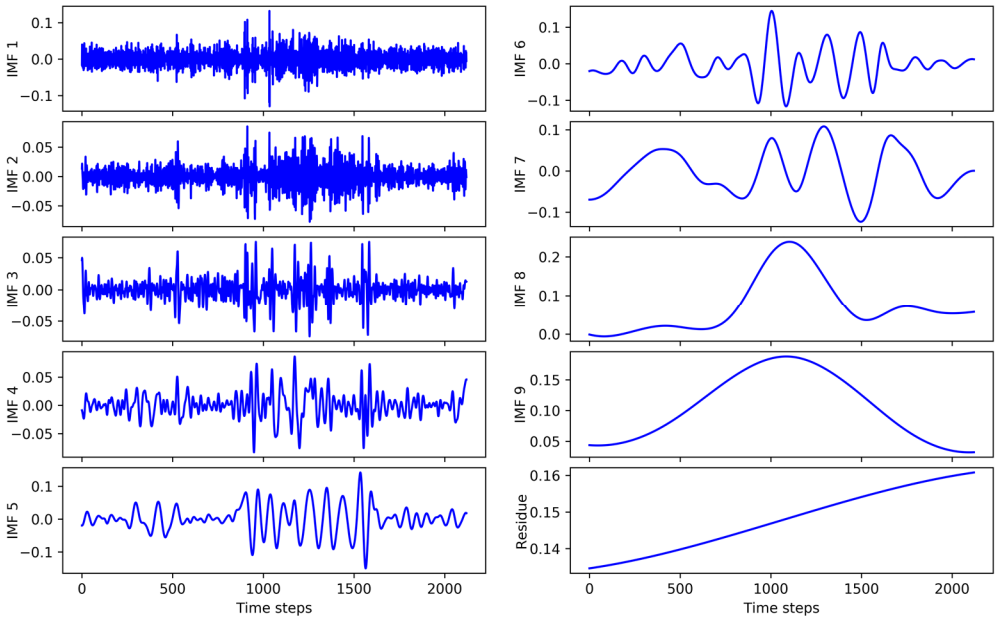
Fig. 3. Overview of the proposed multi-step ahead prediction method.

3.1 Decomposition of the original time series

EEMD is employed for decomposing the raw time series X_t into N_c separate frequency components $\{IMF_i(t)\}$ $(i=1,2,...,N_c)$. The number of IMFs N_c is automatically set by the method and depends on the time series characteristics. Fig. 4 shows an example of EEMD decomposition of a signal measured from a NPP reactor coolant pump (RCP), which is highly nonlinear, nonstationary and noisy. The number of noise realizations *J,* which determines the ensemble size, is set equal to 100 and the noise standard deviation σ_N to 0.05, based on trial and error. EEMD decomposes the original time series into N_c = 9 IMFs and one residue component, as shown in Fig. 4(b). Notice that the complexity of the original time series is reduced in the decomposed components, which appear easier to predict.



(a) Raw measurements obtained from a NPP RCP.



(b) Decomposed IMFs and residue.

Fig. 4. Time series decomposition obtained by using EEMD.

3.2 Multi-step ahead prediction

In the second stage of the proposed method, we develop a dedicated model for the multi-step ahead prediction $\widehat{IMF}_i(t+h)$ $(i=1,...,N_c; h=1,...,H)$ of the EEMD IMFs, based on LSTM and MIMO prediction. The hyperparameters of each prediction model are automatically set during the training phase by using the TPE procedure of Section 2.3. In the testing phase, the predictions of the components $\widehat{IMF}_i(t+h)$ are performed and aggregated to obtain the multi-step ahead prediction $\{\hat{x}_{t+h}\}$ of the original time series. The details of the hyperparameter optimization during the training phase and the MIMO prediction strategy are described in Sections 3.2.1 and 3.2.2, respectively.

3.2.1 Hyperparameter optimization

The three hyperparameters of the LSTM models optimized by the TPE are the activation $\phi(x)$ and optimization $G(\theta,\alpha)$ functions, and the learning rate α. The hyperparameters search spaces are reported in Table 1. The optimization process is performed with 30 iterations and we employ a *k*-fold cross-validation with $k=3$, to avoid overfitting in the computation of the objective function. The

number of epochs N_{epoch} considered for the LSTM training is 100.

Table 1. List of hyperparameters of the LSTM models optimized by TPE (column 1), types of distributions from which they are sampled (column 2) and corresponding domains (column 3).

Hyperparameter	Type of distribution	Domain
Activation function $\phi(x)$	Categorical	{Linear, Sigmoid, Tanh, ReLU}
Optimization function $G(\theta,\alpha)$	Categorical	{SGD, RMSprop, Adam}
Learning rate α	Uniform float	[0.0001, 0.1]

3.2.2 MIMO prediction strategy

As introduced in Section 2.4.3, the MIMO strategy for multi-step ahead prediction offers the following three main advantages with respect to the recursive and direct methods: 1) avoiding the problem of the recursive strategy of error accumulation in long-term predictions; 2) reducing the computational cost of training the models of the direct prediction strategy; 3) preserving the temporal dependencies. For these reasons, the MIMO strategy allows improving the prediction accuracy with respect to both the recursive and direct prediction strategies [6]. Further comparisons among the MIMO and the two other strategies in practical prognostic applications can be found in [43].

Fig. 5 illustrates the multi-step ahead prediction model based on the MIMO strategy where $\{f,[\theta]\}$ denotes the LSTM model and its hyperparameters, which are automatically optimized by TPE during the training process. In addition, the False Nearest Neighbor (FNN) algorithm [44] is employed to determine the optimal value of the embedding dimension *d*.

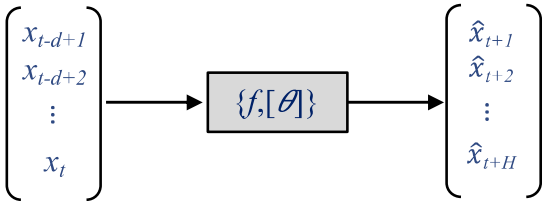
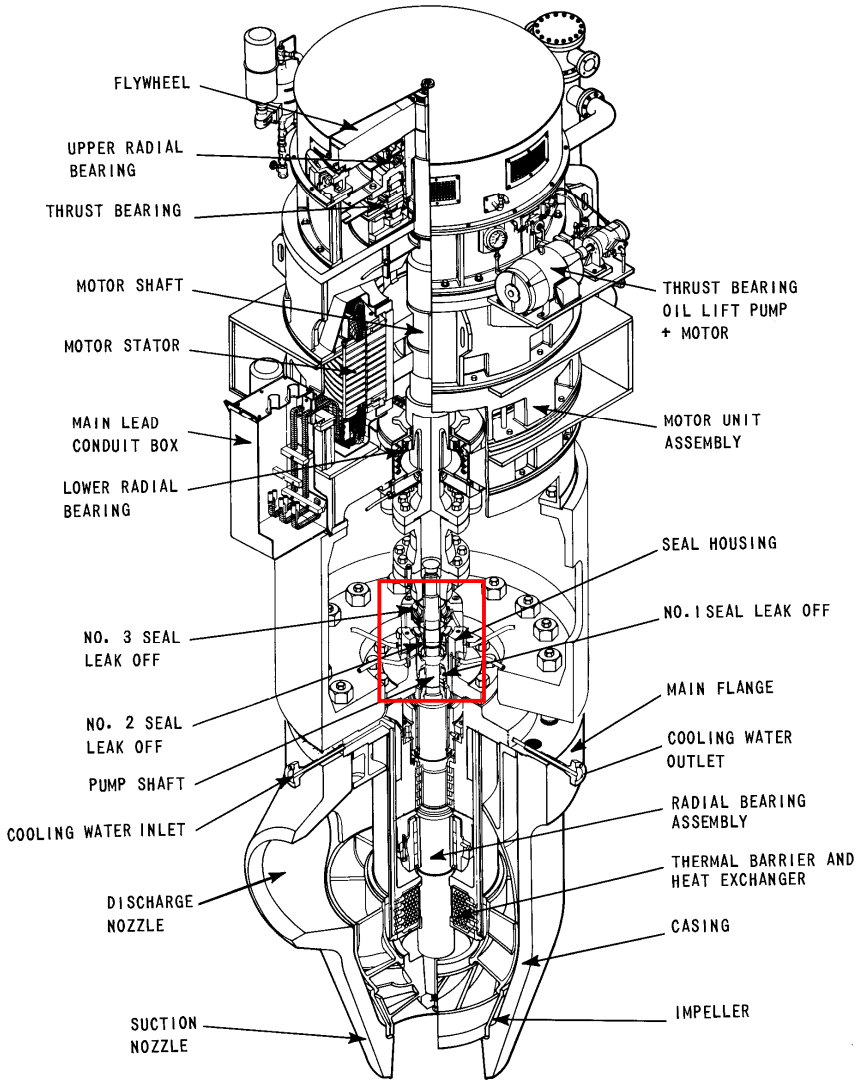


Fig. 5. Scheme of the MIMO strategy for multi-step ahead prediction.

4 Case study: Prediction of the leakage flow of NPP RCPs

The Reactor Coolant Pumps (RCPs) of a NPP is the most critical component of the Reactor Coolant System (RCS), given its functions of transferring the thermal energy generated in the reactor core to the primary coolant and circulating the coolant between the reactor and the steam generators. Fig. 6 depicts the structures of the RCS and the RCP of a Pressurized Water Reactor (PWR).

(a) Reactor Coolant System

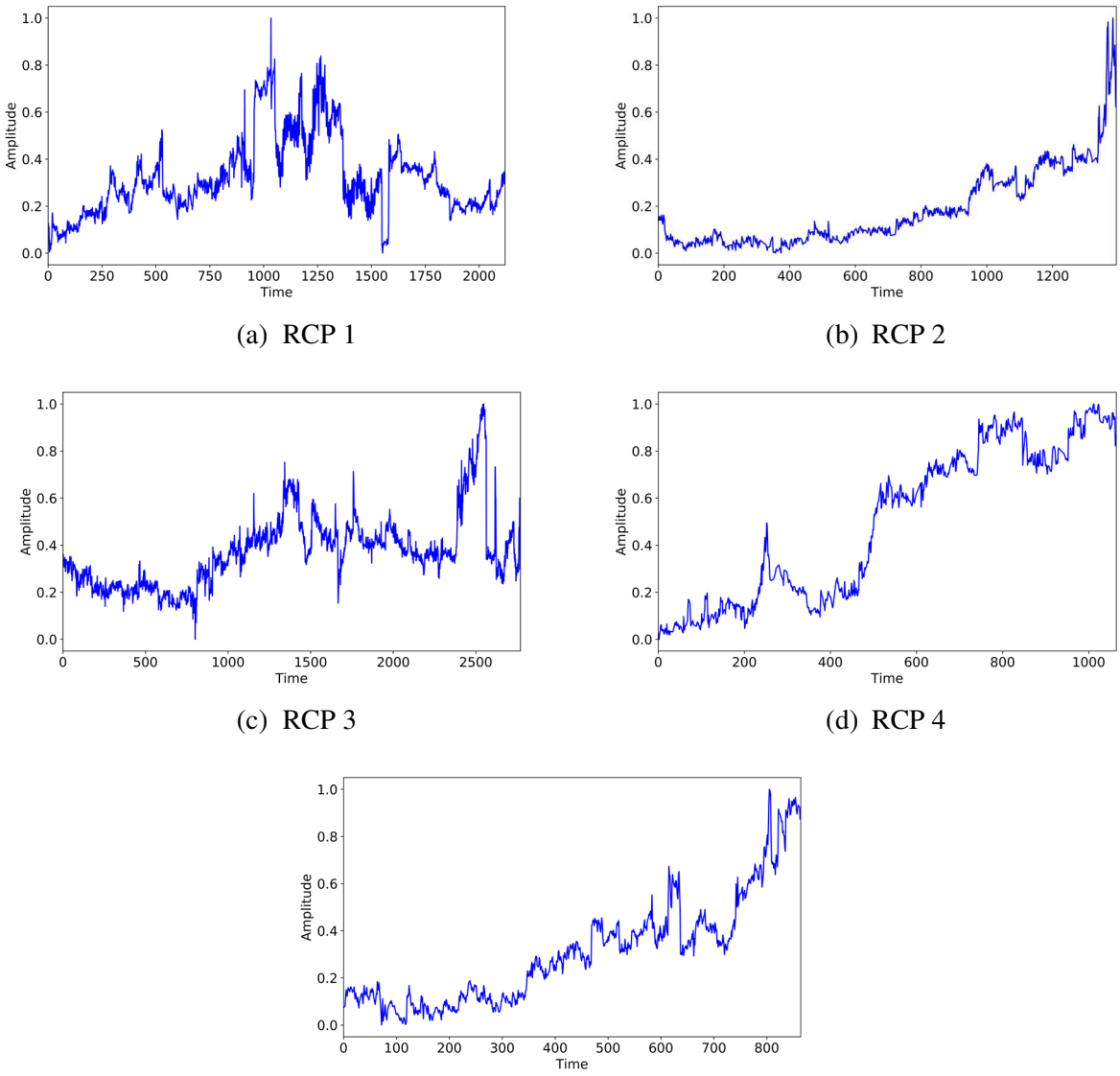


(b) Reactor Coolant Pump with the shaft seal system highlighted.

Fig. 6. Representation of the PWR Reactor Coolant System (RCS) (top) and of the Reactor Coolant Pump (RCP) (bottom). The images have been taken from [45].

One of the most vulnerable components of a RCP is the shaft seal system, which is shown by the red rectangle in Fig. 6(b). It is composed of three mechanical seals located between the electric motor and the impeller, and it plays an important role in limiting the leakages from the primary circuit to the ambient environment by collecting and routing them to the seal leakoff system [46]. A failure of the shaft seal system can cause a loss of reactor primary coolant, with potentially catastrophic consequences [47]. Therefore, as soon as the leakage flow exceeds a safety threshold, the plant is shut down to protect personnel and facilities and prevent environmental impacts due to radioactive releases from the nuclear reactor core.

The case study considers five scenarios of RCP seal leakages, which will be indicated by RCP 1, RCP 2, RCP 3, RCP 4 and RCP 5. Each scenario refers to a different NPP, whose name is omitted for confidentiality reasons. Fig. 7 shows the time series data, which contain the time evolution of the signal "leakage flow from the first RCP seals" during the whole scenario. The leakages are measured every four hours, have occurred in different periods of time and have different durations ranging from 861 (RCP 5) to 2767 (RCP 3) data points. The time series have been normalized in the range [0, 1] for confidentiality reasons. In the scenario RCP 1 (Fig. 7(a)), the operators have been able to successfully managed the seal failure and the pump was brought back to a normal condition. Each time series is divided into two parts: the first 70% of the time series is used for developing the prediction models (training set) and the latter 30% for evaluating the model performance (test set). A dedicated prediction model is developed for each one of the five time series. Notice the large level of noise affecting the time series, which is caused by the process noise and the measurement error. Since the signal evolution during the scenario is influenced by several factors, such as the leakage magnitude, the operator interventions, the operating conditions of the NPP at the time of the leakage, the characteristics of the specific NPP. A basic assumption behind the use of the proposed method is that the information content of the training set is sufficient to predict the signal evolution in the test set.



(a) RCP 1

(b) RCP 2

(c) RCP 3

(d) RCP 4

(e) RCP 5

Fig. 7. Evolution of the normalized leakage flow from the first RCP seals in the five RCP leakage scenarios of the case study.

5 Results and discussion

The objective of this Section is to validate the proposed method with respect to the effectiveness of: 1) decomposing the time series into IMFs components; 2) performing a separated multi-step ahead prediction for each IMF component; 3) using LSTM neural networks for multi-step ahead prediction.

For each one of the five experiments, the model performance has been evaluated considering

three different prediction horizons: 6 steps (1 day), 12 steps (2 days) and 18 steps (3 days) ahead. The following three accuracy metrics have been considered: RMSE (Eq. (15)), Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{x}_i - x_i}{x_i}\right| \times 100\% \tag{21}$$

and Mean Absolute Scaled Error (MASE):

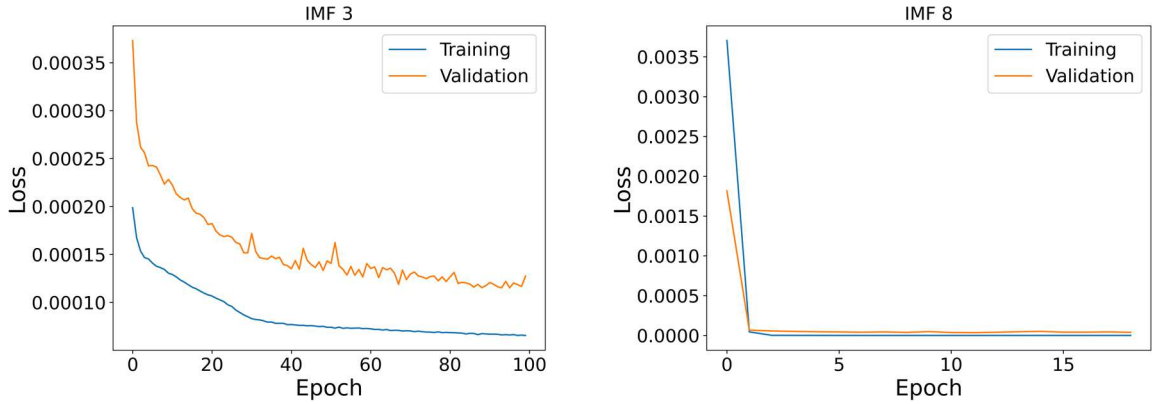
$$MASE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left|\hat{x}_i - x_i\right|}{\frac{1}{N-1}\sum_{j=2}^{N}\left|x_j - x_i\right|}\right), \tag{22}$$

where N is the number of test observations in the test set, and x and $\hat{x}$ are the observed and predicted values, respectively. All the experiments are performed using a GPGPU node composed of two Intel Xeon CPU E5-2695 (24 cores at 2.40 Hz, 32 GB of RAM) and two Nvidia Tesla K40m graphic cards (12 GB of GRAM).

The number of training epochs N_{epoch} for the LSTM neural network is set by using a trial-and-error approach. More specifically, training is stopped when the training loss does not decrease for $N_{patience}$ = 10 epochs, considering a maximum number of epochs N_{max_epoch} equal to 100. Table 2 reports the number of epochs required for training the prediction models for all the 9 IMFs and the residue extracted from the RCP 3 time series (Fig. 4(b)). Notice that training is stopped within 80 epochs for the lower IMFs (IMFs 6 – 9), which contain less noise, and only reach 100 epochs for IMF 3, which is a noise-dominant component. Fig. 8 shows a comparison of the training and validation losses of one fold of the cross-validation for training IMFs 3 and 8 of the RCP 3 time series. The training and validation losses of the IMF 3 (Fig. 8(a)) are almost stable at epoch 100, while the training and validation losses of the IMF 8 (Fig. 8(b)) converge within the first 10 epochs.

Table 2. The number of training epochs required for the IMFs of the RCP 3 time series.

	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	IMF 9	Residue
Num. training epochs	45	77	100	92	40	72	12	19	20	12



(a) (b)

Fig. 8. Training and validation losses of the models predicting IMFs 3 (left, a) and 8 (right, b) of the RCP 3 time series.

5.1 Validation of decomposing the time series into IMF components

We compare the proposed method with a method which does not perform the EEMD decomposition and directly feeds the LSTM model with the time series (Fig. 9). The LSTM architecture is with two layers of 64 neurons each. The hyperparameters are optimized by using the TPE algorithm. This method will be referred to as Comp-A.

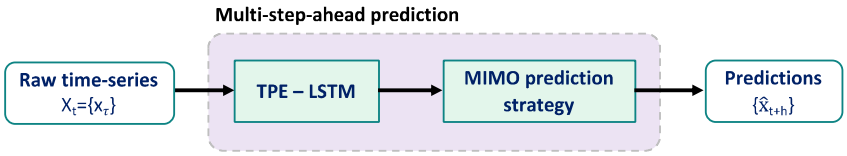
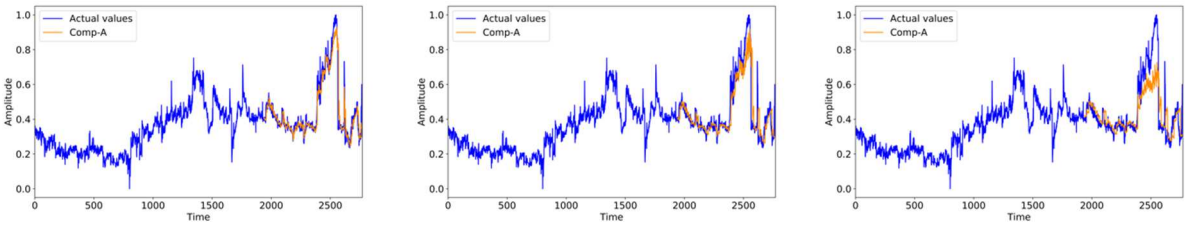


Fig. 9. Scheme of the method Comp-A used for the comparison.

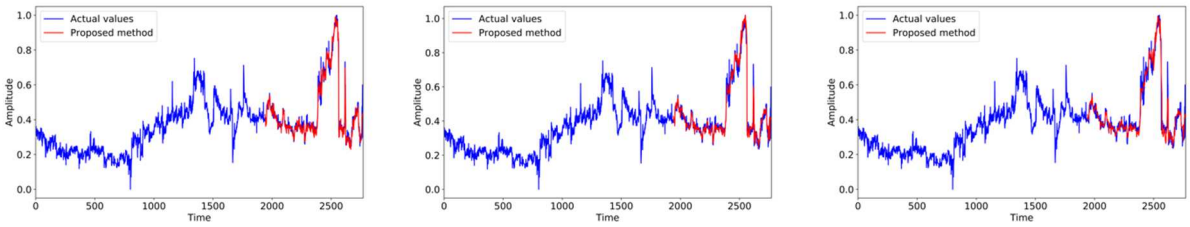
Fig. 10 shows the obtained predictions of the time series considering the three time horizons on the RCP 3 scenario, whereas Table 3 reports the corresponding performance metrics on all the scenarios. In Table 3, the more accurate results (the lower values of the metrics) are highlighted in bold. Notice that the introduction of the decomposition step allows significantly increasing the prediction accuracy on all the prediction horizons on all of the scenarios.



(a) Comparative method – 6 steps ahead

(b) Comparative method – 12 steps ahead

(c) Comparative method – 18 steps ahead



(d) Proposed method – 6 steps ahead

(e) Proposed method – 12 steps ahead

(f) Proposed method – 18 steps ahead

Fig. 10. Predictions of the method Comp-A used for the comparison (top) and of the proposed method (bottom) on the RCP 3 scenario.

Table 3. Performances of the method Comp-A and of the proposed method on the five RCP scenarios.

Scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Comp-A method	0.0405	13.8939	1.6168	0.0608	30.1245	2.3199	0.0667	30.4937	2.6540
	Proposed method	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Comp-A method	0.0776	11.6117	3.5690	0.0897	18.9838	5.2261	0.0893	16.2510	4.5966
	Proposed method	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Comp-A method	0.0627	7.9651	1.7586	0.0868	11.1782	2.5560	0.1081	14.2730	3.6001
	Proposed method	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Comp-A method	0.0568	5.4109	3.1283	0.0730	6.9583	4.1817	0.0891	8.3991	4.9783
	Proposed method	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Comp-A method	0.1583	16.7301	4.1357	0.1651	18.9645	4.5333	0.0988	12.9969	2.5915
	Proposed method	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**
Average	Comp-A method	0.0791	11.1223	2.8416	0.0950	17.2418	3.7634	0.0904	16.4827	3.6841
	Proposed method	**0.0256**	**4.7068**	**1.0468**	**0.0321**	**5.9571**	**1.2874**	**0.0413**	**8.5470**	**1.6091**

5.2 Validation of performing a separated multi-step prediction for each IMF component

We compare the proposed method with a method based on a single LSTM which receives in input

all the IMF components provided by the EEMD and provides in output the signal prediction using the MIMO strategy. This method will be referred to as Comp-B (Fig. 11). The LSTM architecture is with two layers of 64 neurons each and the LSTM hyperparameters are optimized using the TPE algorithm.

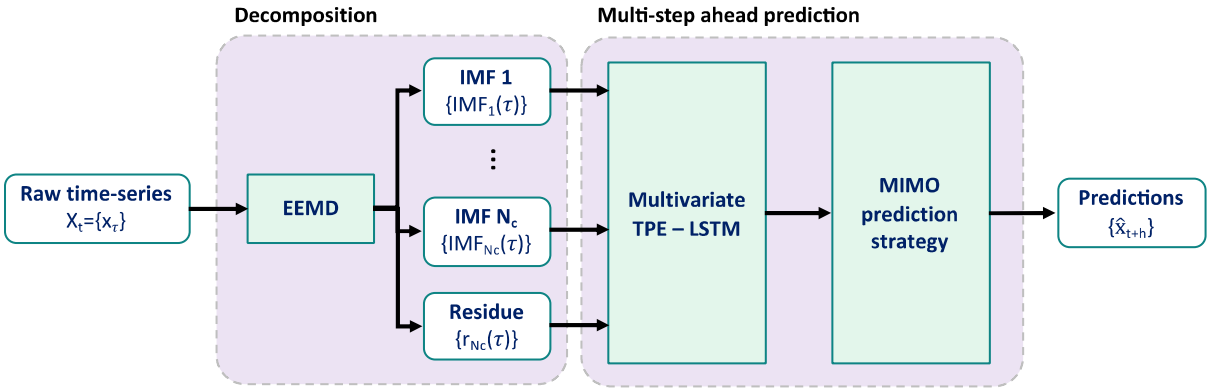
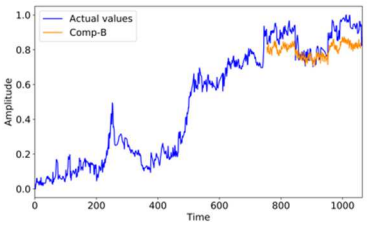
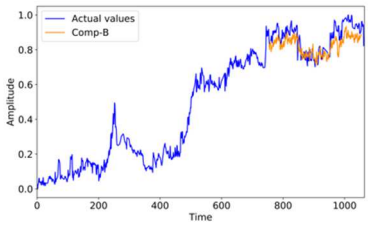


Fig. 11. Scheme of the method Comp-B used for the comparison.

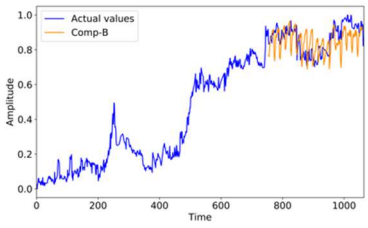
Fig. 12 shows the obtained predictions of the time series considering the three time horizons on the RCP 4 scenario, whereas Table 4 reports the corresponding performance metrics on all the scenarios. Notice that the proposed method provides more accurate prediction than the method Comp-B on all the five scenarios and the three time horizons. This is due to the use of an ensemble of models, which allows reducing the noise and spikes of the predictions obtained by the method Comp-B based on a single model, as shown in Fig. 12.



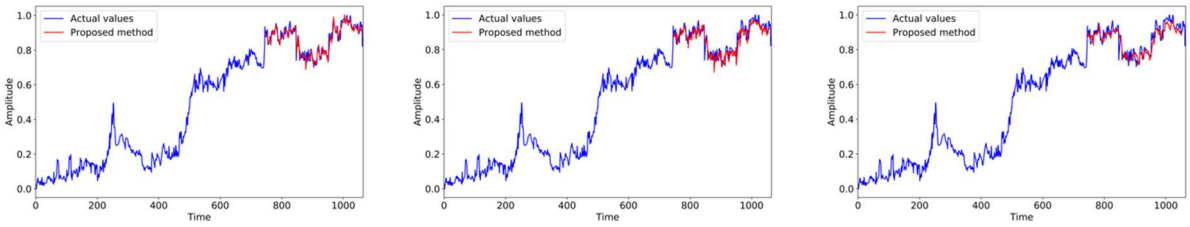
(a) Comparative method – 6 steps ahead



(b) Comparative method – 12 steps ahead



(c) Comparative method – 18 steps ahead



(d) Proposed method – 6 steps ahead

(e) Proposed method – 12 steps ahead

(f) Proposed method – 18 steps ahead

Fig. 12. Predictions of the method Comp-B used for the comparison (top) and of the proposed method (bottom) on the RCP 4 scenario.

Table 4. Performances of the method Comp-B and of the proposed method on the five RCP scenarios.

Scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Comp-B method	0.0249	9.3767	1.3214	0.0386	20.3446	1.9242	0.0455	17.8378	2.4970
	Proposed method	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Comp-B method	0.0709	9.2070	2.8689	0.0483	6.3916	1.9585	0.0916	20.2789	6.2618
	Proposed method	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Comp-B method	0.0500	8.1171	1.8032	0.0747	10.8237	2.6785	0.0760	10.5087	2.5134
	Proposed method	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Comp-B method	0.0851	8.1530	5.0001	0.0607	5.6678	3.4487	0.0819	7.7580	4.5046
	Proposed method	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Comp-B method	0.1375	18.8527	3.8056	0.3038	27.1751	6.1985	0.1340	18.1811	3.6363
	Proposed method	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**
Average	Comp-B method	0.0736	10.7413	2.9598	0.1052	14.0805	3.2416	0.0858	14.9129	3.8826
	Proposed method	**0.0256**	**4.7068**	**1.0468**	**0.0321**	**5.9571**	**1.2874**	**0.0413**	**8.5470**	**1.6091**

5.3 Validation of the use of LSTM

We compare the proposed method with a method based on the use of a state of the art model different from LSTM for performing the multi-step ahead prediction. We consider the Echo State Networks (ESNs) which are RNNs that have shown very satisfactory performances in the prediction of highly nonlinear and nonstationary time series [48], [49].

ESN is a RNN with a sparsely connected hidden layer [50]. The connectivity and weights of the hidden neurons (also known as reservoirs) are randomly assigned and fixed, whereas the weights of the output neurons are learned by using a linear regression algorithm. The advantages of ESN are the

simple network structure and a low computational cost compared to conventional RNNs. More details about ESN can be found in [50], [51].

The method used for this comparison follows the same scheme of the proposed method from which it differs only for the use of ESNs instead of LSTMs. It will be referred to as Comp-C (Fig. 13). TPE is used to optimize the two major hyperparameters of the ESN models, i.e. the number of reservoir neurons and the spectral radius. Table 5 reports the considered ranges of the ESN hyperparameters.

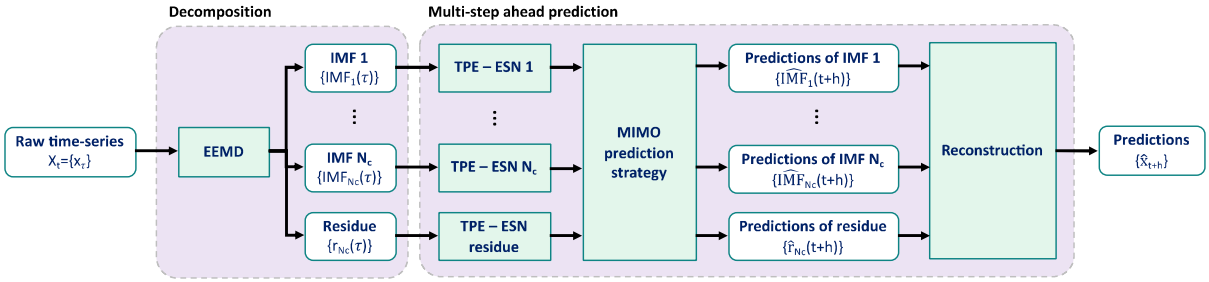
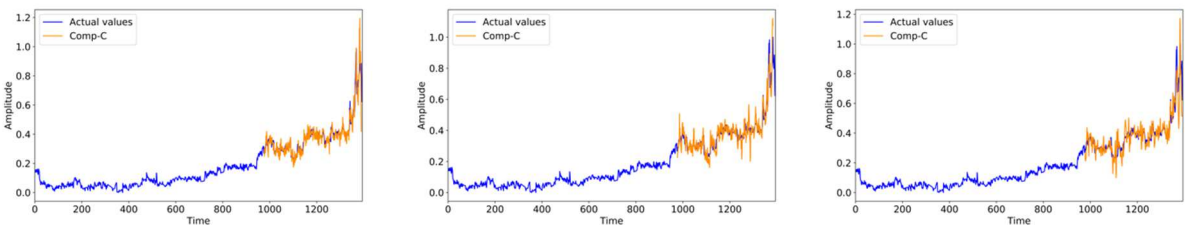


Fig. 13. Scheme of the method Comp-C used for the comparison.

Table 5. Hyperparameters of the ESN models optimized by TPE.

Hyperparameter	Type of distribution	Search space
Number of reservoir neurons	Uniform integer	[20, 500]
Spectral radius	Uniform float	[0.01, 1]

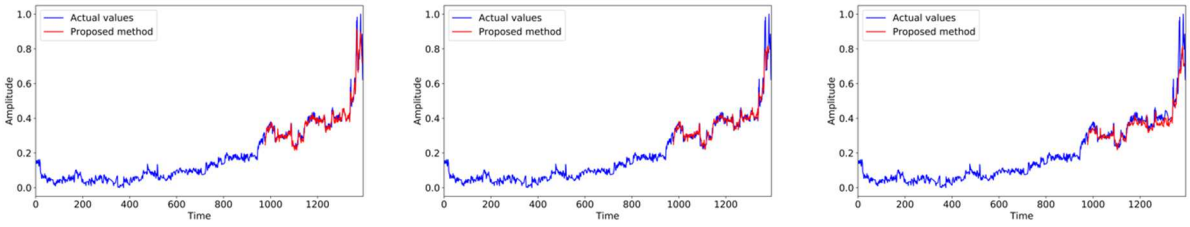
Fig. 14 shows the obtained predictions of the time series considering the three time horizons on the RCP 2 scenario, whereas Table 6 reports the corresponding performance metrics on all the scenarios. The obtained results show that the use of LSTMs allows improving the accuracy with respect to ESNs. The performance improvement is more significant when long-term prediction horizons are considered.



(a) Comparative method – 6 steps ahead

(b) Comparative method – 12 steps ahead

(c) Comparative method – 18 steps ahead



(d) Proposed method – 6 steps ahead

(e) Proposed method – 12 steps ahead

(f) Proposed method – 18 steps ahead

Fig. 14. Predictions of the method Comp-C used for the comparison (top) and of the proposed method (bottom) on the RCP 2 scenario.

Table 6. Performances of the method Comp-C and of the proposed method on the five RCP scenarios.

Scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Comp-C method	0.0450	18.2183	2.4145	0.0521	20.1742	2.7480	0.0544	21.2928	2.9608
	Proposed method	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Comp-C method	0.0496	7.7376	2.2372	0.0511	9.4669	2.6751	0.0672	11.1388	3.1238
	Proposed method	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Comp-C method	0.0647	11.3833	2.3444	0.0616	11.1641	2.3267	0.0750	12.7170	2.5916
	Proposed method	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Comp-C method	0.0419	3.7228	2.0821	0.0480	4.5109	2.5750	0.0675	6.0974	3.5221
	Proposed method	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Comp-C method	0.0380	5.1331	1.0158	0.0578	8.0459	1.5163	0.0835	13.3672	2.3941
	Proposed method	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**
Average	Comp-C method	0.0478	9.2390	2.0188	0.0541	10.6724	2.3682	0.0695	12.9226	2.9184
	Proposed method	**0.0256**	**4.7068**	**1.0468**	**0.0321**	**5.9571**	**1.2874**	**0.0413**	**8.5470**	**1.6091**

6 Discussion and conclusions

A method for the prediction of the future evolution of time series signals in energy systems over a long time horizon has been developed to help decision makers improving maintenance planning and minimizing unexpected shutdowns. It is based on the combined use of Ensemble Empirical Mode Decomposition and Long Short-Term Memory neural networks. Ensemble Empirical Mode Decomposition allows reducing the complexity of raw time series by breaking down them into

separate frequency components characterized by more linear and stationary trends, which facilitate their individual prediction. The multi-step ahead prediction of each one of the decomposed components is performed using Long Short-Term Memory neural networks with the Multi-Input Multi-Output prediction strategy, which allows preserving the temporal dependencies in the time series. The proposed method relies on the use of the Tree-structured Parzen Estimator algorithm to automatically select the hyperparameters of each prediction model during the training phase.

A practical case study has been considered, concerning the prediction over three different time horizons, up to 18 steps (3 days) ahead, of the time series evolution of Reactor Coolant Pump seal leakage flow in Nuclear Power Plants. The results obtained show that the average prediction accuracy of the proposed method is improved of 60.52% with respect to alternative state of the art approaches. It has also been shown that: 1) the multi-step ahead predictions obtained by an ensemble of separate prediction models are more accurate and less noisy than the predictions obtained by a single model and 2) the performance improvement is more significant when long-term prediction horizons, characterized by the presence of multiple and very different superposed dynamic trends, are considered.

Future work will include the embedding of the proposed predictive model into a practical prognostic context for the quantification of its effectiveness for operation and maintenance of energy production plants.

Acknowledgement

The participation of Piero Baraldi and Enrico Zio at this work has been partially funded by INAIL within the BRIC/2018, ID = 11 framework, project MAC4PRO.

References

[1] A. Coppola, “Reliability Engineering of Electronic Equipment: A Historical Perspective,” *IEEE Trans. Reliab.*, 1984.

[2] E. Zio, “Some Challenges and Opportunities in Reliability Engineering,” *IEEE Trans. Reliab.*, vol. PP, no. 99, pp. 1769–1782, 2016.

[3] B. Sun, S. Zeng, R. Kang, and M. G. Pecht, “Benefits and challenges of system prognostics,” *IEEE Trans. Reliab.*, 2012.

[4] I. Renewable Energy Agency, “Renewable Power Generation Costs in 2017,” *Int. Renew. Energy Agency*, 2018.

[5] European Parliament and the Council, “Report from the Commission to the European Parliament and the Council: On progress of clean energy competitiveness,” 2020.

[6] S. Ben Taieb and A. F. Atiya, “A Bias and Variance Analysis for Multistep-Ahead Time Series Forecasting,” *IEEE Trans. Neural Networks Learn. Syst.*, 2016.

[7] K. Moshkbar-Bakhshayesh and M. B. Ghofrani, “Development of a new method for forecasting future states of NPPs parameters in transients,” *IEEE Trans. Nucl. Sci.*, 2014.

[8] D. S. Kim, S. W. Lee, and M. G. Na, “Prediction of axial DNBR distribution in a hot fuel rod using support vector regression models,” *IEEE Trans. Nucl. Sci.*, 2011.

[9] D. Y. Kim, K. H. Yoo, J. H. Kim, M. G. Na, S. Hur, and C. H. Kim, “Prediction of leak flow rate using fuzzy neural networks in severe post-loca circumstances,” *IEEE Trans. Nucl. Sci.*, 2014.

[10] M. Marseguerra, E. Zio, P. Baraldi, and A. Oldrini, “Fuzzy logic for signal prediction in nuclear systems,” *Prog. Nucl. Energy*, 2003.

[11] Y. K. Liu, F. Xie, C. L. Xie, M. J. Peng, G. H. Wu, and H. Xia, “Prediction of time series of NPP operating parameters using dynamic model based on BP neural network,” *Ann. Nucl. Energy*, 2015.

[12] W. Yan, “Toward automatic time-series forecasting using neural networks,” *IEEE Trans. Neural Networks Learn. Syst.*, 2012.

[13] J. G. De Gooijer and R. J. Hyndman, “25 years of time series forecasting,” *Int. J. Forecast.*, 2006.

[14] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” *Int. J. Forecast.*, 1998.

[15] J. Liu and E. Zio, “SVM hyperparameters tuning for recursive multi-step-ahead prediction,” *Neural Comput. Appl.*, 2017.

[16] Z. Qu, W. Mao, K. Zhang, W. Zhang, and Z. Li, “Multi-step wind speed forecasting based on a hybrid decomposition technique and an improved back-propagation neural network,” *Renew. Energy*, 2019.

[17] M. Wu, C. Stefanakos, Z. Gao, and S. Haver, “Prediction of short-term wind and wave conditions for marine operations using a multi-step-ahead decomposition-ANFIS model and quantification of its uncertainty,” *Ocean Eng.*, 2019.

[18] N. Mohajerin and S. L. Waslander, "Multistep Prediction of Dynamic Systems with Recurrent Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, 2019.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, 1997.

[20] Z. Zhang *et al.*, "Wind speed prediction method using Shared Weight Long Short-Term Memory Network and Gaussian Process Regression," *Appl. Energy*, 2019.

[21] S. Ghimire, R. C. Deo, N. Raj, and J. Mi, "Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms," *Appl. Energy*, 2019.

[22] Y. Bai, B. Zeng, C. Li, and J. Zhang, "An ensemble long short-term memory neural network for hourly PM2.5 concentration forecasting," *Chemosphere*, 2019.

[23] Y. Baek and H. Y. Kim, "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Syst. Appl.*, 2018.

[24] W. Wang, T. Hong, X. Xu, J. Chen, Z. Liu, and N. Xu, "Forecasting district-scale energy dynamics through integrating building network and long short-term memory learning algorithm," *Appl. Energy*, 2019.

[25] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, 2019.

[26] Z. Hajirahimi and M. Khashei, "Hybrid structures in time series modeling and forecasting: A review," *Eng. Appl. Artif. Intell.*, 2019.

[27] Ü. Ç. Büyükşahin and Ş. Ertekin, "Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition," *Neurocomputing*, 2019.

[28] Y. Li, H. Shi, F. Han, Z. Duan, and H. Liu, "Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy," *Renew. Energy*, 2019.

[29] Z. Qian, Y. Pei, H. Zareipour, and N. Chen, "A review and discussion of decomposition-based hybrid models for wind energy forecasting applications," *Applied Energy*. 2019.

[30] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2011.

[31] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, 1998.

[32] P. Flandrin, G. Rilling, and P. Goncalves, "Empirical mode decomposition as a filter bank," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 112–114, 2004.

[33] P. Nguyen and J. M. Kim, "Adaptive ECG denoising using genetic algorithm-based thresholding and ensemble empirical mode decomposition," *Inf. Sci. (Ny).*, 2016.

[34] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data

analysis method," *Adv. Adapt. Data Anal.*, 2009.

[35] C. Olah, "Understanding LSTM Networks," 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[36] M. Feurer and F. Hutter, "Hyperparameter Optimization," 2019.

[37] R. Kohavi and G. H. John, "Automatic Parameter Selection by Minimizing Estimated Error," in *Machine Learning Proceedings 1995*, 1995.

[38] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[39] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, vol. 24, p. 2546.

[40] R. Gouriveau and N. Zerhouni, "Connexionist-systems-based long term prediction approaches for prognostics," *IEEE Trans. Reliab.*, 2012.

[41] D. R. Cox, "Prediction by Exponentially Weighted Moving Averages and Related Methods," *J. R. Stat. Soc. Ser. B*, 1961.

[42] S. Ben Taieb, G. Bontempi, A. Sorjamaa, and A. Lendasse, "Long-term prediction of time series by combining direct and MIMO strategies," in *Proceedings of the International Joint Conference on Neural Networks*, 2009.

[43] H. P. Nguyen, J. Liu, and E. Zio, "A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators," *Appl. Soft Comput. J.*, 2020.

[44] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A*, 1992.

[45] "Reactor Cooling System (PWR)." [Online]. Available: http://www.nucleartourist.com/systems/rcs1.htm.

[46] R. Loehberg, W. Ullrich, and K. Gaffal, "Shafts of main coolant pumps - failure analysis and remedies," *Nucl. Eng. Des.*, 1989.

[47] J. Liu and E. Zio, "An adaptive online learning approach for Support Vector Regression: Online-SVR-FID," *Mech. Syst. Signal Process.*, 2016.

[48] Z. Wang, Y. R. Zeng, S. Wang, and L. Wang, "Optimizing echo state network with backtracking search optimization algorithm for time series forecasting," *Eng. Appl. Artif. Intell.*, 2019.

[49] N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of Echo State Network parameters for time series forecasting," *Appl. Soft Comput. J.*, 2017.

[50] H. Jaeger, “Echo state network,” *Scholarpedia*, 2007.

[51] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.*, 2009.