Model-based and data-driven prediction methods for prognostics

Hoang-Phuong Nguyen

université PARIS-SACLAY

Model-based and data-driven prediction methods for prognostics

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°573 approches interdisciplinaires, fondements, applications et innovation (Interfaces)
Spécialité de doctorat: Ingéniérie des systèmes complexes
Unité de recherche: Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel, 91190, Gif-sur-Yvette, France

Thèse présentée et soutenue à Gif-sur-Yvette, le 25 mai 2020, par

Hoang-Phuong NGUYEN

Composition du Jury

Anne BARROS Professeure, CentraleSupélec, Université Paris-Saclay	Présidente
Eleni CHATZI Professeure associé, ETH Zurich	Examinatrice
Emmanuel REMY Ingénieure-chercheur experte, EDF R&D	Examinateur
Jamie COBLE Professeure associé, University of Tennessee	Rapporteur & Examinatrice
Kamal MEDJAHER Professeur, Tarbes National School of Engineering (ENIT)	Rapporteur & Examinateur
Enrico ZIO Professeur, MINES ParisTech et Politecnico di Milano	Directeur de thèse

Thèse de doctorat

NNT : 2020UPASC021

This page intentionally left blank.

Acknowledgement

First of all, I would like to express my most sincere appreciation to my Thesis supervisor, Prof. Enrico Zio, for accepting me as his student four years ago and guiding me throughout the entire PhD journey. More than a supervisor, Prof. Zio has been my career role model with his deeply insightful understanding of several domains from risk analysis, resilience to PHM, his rigorous organization of the work and time and his strong enthusiasm for the research. He has always been the hardest working one that I have ever met. I have been very fortunate to do my PhD with him. Thank you very much for all these three years, Professor!

I would like to thank Prof. Jie Liu at Beihang University, who joined this thesis as the co-supervisor in the first year of my PhD. He was always willing and patient to answer all my questions, both technical and personal ones, and give me great advice with an extremely nice and sincere manner, even when we worked in different locations very far from each other.

I would like also to thank Prof. Piero Baraldi for welcoming me at LASAR team of Politecnico di Milano in the autumn 2019 where I had the opportunity to work closely with Prof. Zio and him, and to meet great people, including Prof. Di Maio, Ahmed, Xuefei, Luca, Federico, Dario and many others. Although it was only three months, what Prof. Zio and Prof. Baraldi had taught allowed me to progress a lot, and I had so much enjoyable and happy time with everyone, especially having morning coffee and lunch together everyday and our conference trip in Rome. Thank you, Prof. Baraldi and Prof. Di Maio, I will miss Macondo a lot!

I am deeply grateful to all the jury members, Prof. Anne Barros, Prof. Eleni Chatzi, Prof. Jamie Coble, Prof. Kamal Medjaher and Mr. Emmanuel Remy, who agreed to be part of the committee, even though it was in the difficult times of the virus and the confinement. I am very pleased and honored to have presented my PhD work in front of such high qualified jury. Special thanks to the reviewers, Prof. Jamie Coble and Prof. Kamal Medjaher, for the evaluation of the manuscript and all the constructive and helpful remarks.

I would like thank all my colleagues of the Chair on Systems Science and Energy Challenge (SSEC) and the Laboratoire Génie Industriel (LGI). It was my great pleasure to be working here with everyone. And in particular, many thanks to Daogui and Hongping, for having been my colleagues, my friends and my office neighbors for three years.

PhD is a long journey and I was fortunate enough to meet great people in France. In particular, I would like to thank Haythem, Hichame, Sadeque, anh Thanh, and Cẩm An for being my close friends throughout all these three years, helping me when troubles came, raising me up when I was down, sharing with me a lot of ramens (Haythem and Hichame) and beers (anh Thanh and An). Thank you very much, everyone. I will always remember our memories.

Finally, to my family:

Gửi cha, mẹ và chị 2. Con để mọi người đợi lâu quá rồi, từ lúc con quyết định đi học thạc sĩ đến giờ. 2013, đã gần như 7 năm. Mọi chuyện giờ đã xong. Con đang trở về với nhà. Cảm ơn cha mẹ và chị 2.

Orsay, 10 June 2020.

Abstract

Degradation is an unavoidable phenomenon that affects engineering components and systems, which may lead to their failures with potentially catastrophic consequences depending on the application. The motivation of this Thesis is trying to model, analyze and predict failures with prognostic methods that can enable a predictive management of asset maintenance. This would allow decision makers to improve maintenance planning and minimize unexpected shutdowns, thus increasing system availability and safety. To this aim, research during the Thesis has been devoted to the tailoring and use of both model-based and data-driven approaches to treat the degradation processes that can lead to different failure modes in industrial components, making use of different information and data sources for performing predictions on the degradation evolution and estimating the Remaining Useful Life (RUL).

The main contributions of the Ph.D. work have been divided into two parts addressing two specific prognostic applications, including model-based prognostics for fatigue crack growth prediction and data-driven prognostics for multi-step ahead predictions of time series data of Nuclear Power Plant (NPP) components, respectively.

The performance of a model-based prognostic approach depends on the choice of the adopted Physics-of-Failure (PoF) models. However, each degradation model is appropriate only to certain degradation process under certain operating conditions, which are often not precisely known. To address this problem, we develop two model-based methods based on the ensemble of multiple degradation models, in order to take advantage of the complementarity of different models, specific on the degradation trends to be predicted. The main contributions of the proposed ensemble of models-based methods are two novel weighted ensemble strategies, which take into account the prediction accuracies of the individual models at previous time instances. In addition, recursive Bayesian filtering and Particle Filtering (PF) are employed to dynamically predict and update the degradation evolution and the component RUL at each prediction step. To validate the performances of the proposed methods, different case studies of fatigue crack growth generated with time-varying operating conditions are considered.

In the nuclear industry, components and systems are designed to guarantee very high reliability levels given the potentially catastrophic consequences of their failures, and prognostic capabilities are sought to accurately predict the long-term degradation behaviors of the components and systems, allowing maintenance interventions of critical components to be planned well in advance and reducing maintenance costs. However, the further one attempts to predict into the future, the harder it is to achieve an accurate and stable prediction due to increasing uncertainty and error accumulation. For this reason, multi-step ahead prediction has remained a difficult task in many prognostic applications, particularly in the nuclear industry. To address this problem, this Thesis proposes two novel multi-step

ahead prediction methods based on Long Short-Term Memory (LSTM), a deep neural network developed for dealing with the long-term dependencies in time series data. The first multi-step ahead prediction method is applied for predicting up to 45 days ahead the operating parameters of NPP Steam Generators (SGs). The method also addresses the additional issues of automatic hyperparameter optimization and prediction uncertainty quantification by using a Tree-structured Parzen Estimator (TPE) optimization algorithm and a Monte Carlo (MC) Dropout technique, respectively. A case study concerning SG data acquired from different NPPs is carried out to validate the performance of the proposed method. On the other hand, the second multi-step ahead prediction method is developed based on a hybrid framework integrating Ensemble Empirical Mode Decomposition (EEMD) and LSTM neural network and applied to a case study concerning time series data acquired from Reactor Coolant Pumps (RCPs) of NPPs. In this prediction framework, EEMD is used to decompose time series into a set of components which allow effectively describing the system dynamics and therefore facilitates the prediction task. Then, LSTM neural network models are developed for predicting the multi-step ahead behavior of the individual components and the obtained predictions are aggregated to reconstruct the time series. A TPE algorithm is employed for automatic hyperparameter optimization. The performance of the proposed method is validated by considering three different long-term prediction horizons on a practical case study of NPP RCPs.

Keywords: prognostics and health management, model-based prognostics, data-driven prognostics, particle filter, recurrent neural network, long short-term memory, time series prediction, multi-step ahead prediction, automatic hyperparameter optimization, fatigue crack growth, nuclear power plant, steam generator, reactor coolant pump

Resume

La dégradation est un phénomène inévitable qui affecte les composants et les systèmes, qui peut entraîner leurs défaillances avec des conséquences potentiellement catastrophiques selon l'application. La motivation de cette Thèse est d'essayer de modéliser, d'analyser et de prédire les défaillances par des méthodes pronostiques qui peuvent permettre une gestion prédictive de la maintenance des actifs. Cela permettrait aux décideurs d'améliorer la planification de la maintenance et de minimiser les arrêts imprévus, augmentant ainsi la disponibilité et la sécurité du système. Dans cet objectif, la recherche pendant la thèse a été consacrée à l'adaptation et à l'utilisation d'approches basées sur des modèles et d'approches pilotées par les données pour traiter les processus de dégradation qui peuvent conduire à différents modes de défaillance dans les composants industriels, en utilisant différentes sources d'informations et de données pour effectuer des prédictions sur l'évolution de la dégradation et estimer la durée de vie utile restante.

Les principales contributions de cette thèse ont été divisées en deux parties traitant de deux applications pronostiques spécifiques, y compris les pronostics basés sur des modèles pour la prédiction de la croissance des fissures par fatigue et les pronostics pilotées par les données pour les prédictions à pas multiples des données de séries chronologiques des composants des Centrales Nucléaires, respectivement.

La performance d'une approche pronostique basée sur des modèles dépend du choix des modèles adoptés de Physics-of-Failure (PoF). Cependant, chaque modèle de dégradation ne convient qu'à certains processus de dégradation dans certaines conditions de fonctionnement, qui souvent ne sont pas connues avec précision. Pour résoudre ce problème, nous développons deux méthodes basées sur des modèles qui reposent sur l'ensemble de plusieurs modèles de dégradation, afin de tirer parti de la complémentarité de différents modèles, spécifiques aux tendances de dégradation à prévoir. Les principales contributions de l'ensemble proposé de méthodes basées sur des modèles sont deux nouvelles stratégies d'ensemble pondérées, qui prennent en compte les précisions de prédiction des modèles individuels lors d'instances de temps précédentes. De plus, le filtrage Bayésien récursif et le filtrage particulaire sont utilisés pour prédire et mettre à jour dynamiquement l'évolution de la dégradation et la durée de vie utile restante du composant à chaque étape de prédiction. Pour valider les performances des méthodes proposées, différentes études de cas de croissance des fissures par fatigue générées avec des conditions de fonctionnement variables dans le temps sont considérées.

Dans l'industrie nucléaire, les composants et les systèmes sont conçus pour garantir des niveaux de fiabilité très élevés étant donné les conséquences potentiellement catastrophiques de leurs défaillances, et des capacités pronostiques sont recherchées pour prédire avec précision les comportements de dégradation à long terme des composants et des systèmes, permettant de planifier les interventions de maintenance des composants critiques bien à l'avance et de réduire les coûts de maintenance.

Cependant, plus loin on tente de prédire l'avenir, plus il est difficile d'obtenir une prédiction précise et stable en raison de l'augmentation de l'incertitude et de l'accumulation d'erreurs. Pour cette raison, la prédiction à plusieurs étapes est restée une tâche difficile dans de nombreuses applications pronostiques, en particulier dans l'industrie nucléaire. Pour résoudre ce problème, cette thèse propose deux nouvelles méthodes de prédiction à étapes multiples basées sur la Long Short-Term Memory (LSTM), un réseau de neurones profond développé pour traiter les dépendances à long terme dans les données de séries chronologiques. La première méthode de prédiction à plusieurs étapes est appliquée pour prédire jusqu'à 45 jours à l'avance les paramètres de fonctionnement des Générateurs de Vapeur de Centrales Nucléaires. La méthode aborde également les problèmes supplémentaires d'optimisation automatique des hyperparamètres et de quantification de l'incertitude de prédiction en utilisant respectivement un algorithme d'optimisation Tree-structured Parzen Estimator (TPE) et une technique de Monte Carlo (MC) Dropout. Une étude de cas concernant les données des Générateurs de Vapeur acquises auprès de différentes Centrales Nucléaires est réalisée pour valider les performances de la méthode proposée. D'autre part, la deuxième méthode de prédiction à plusieurs étapes est développée sur la base d'un cadre hybride intégrant la Ensemble Empirical Mode Decomposition (EEMD) et le réseau de neurones LSTM, et appliquée sur une étude de cas concernant les données de séries chronologiques acquises à partir des Pompes de Refroidissement de Réacteurs de Centrales Nucléaires. Dans ce cadre de prédiction, EEMD est utilisée pour décomposer des séries temporelles en un ensemble de composants qui permettent de décrire efficacement la dynamique du système et facilitent donc la tâche de prédiction. Ensuite, des modèles de réseaux de neurones LSTM sont développés pour prédire le comportement à plusieurs étapes des composants individuels et les prédictions obtenues sont agrégées pour reconstruire des données de séries chronologiques. Un algorithme TPE est utilisé pour l'optimisation automatique des hyperparamètres. La performance de la méthode proposée est validée en considérant trois horizons de prédiction à long terme sur une étude de cas pratique des Pompes de Refroidissement de Réacteurs de Centrales Nucléaires.

Mots clés: prognostics and health management, pronostics basés sur des modèles, pronostics pilotées par les données, filtrage particulaire, réseau de neurones artificiels, long short-term memory, prédiction de séries chronologiques, prédictions à pas multiples, optimisation automatique des hyper-paramètres, croissance des fissures par fatigue, centrales nucléaires, générateurs de vapeur, pompes de refroidissement de réacteurs.

Contents

Chapter 1 Introduction 11

1.1 Prognostics and Health Management (PHM) 11

1.2 Prognostic methods 13
1.2.1 Model-based prognostics 13
1.2.2 Data-driven prognostics 14

1.3 Aims and contributions 14
1.3.1 Model-based prognostics 15
1.3.2 Data-driven prognostics 15

1.4 Structure of the Thesis 16

Chapter 2 Background of Model-Based Fatigue Crack Prognostics 19

2.1 Introduction 19

2.2 PoF models of fatigue crack growth 20
2.2.1 Paris-Erdogan model 20
2.2.2 Polynomial model 21
2.2.3 Global model 21
2.2.4 Curve fitting function 21

2.3 Predictions of the degradation evolution by Bayesian filtering algorithms 22
2.3.1 Estimating the degradation state at the present time 22
2.3.2 Predicting the future degradation evolution and the RUL 23

Chapter 3 An Ensemble Method For Predicting Fatigue Crack Degradation 27

3.1 Proposed ensemble framework for fatigue crack growth prediction 27

3.2 Illustrative case study of fatigue crack growth 29

3.3 Results and discussion 30

Chapter 4 An Emsemble Method Based On Particle Filtering for Fatigue Crack Prognostics . 36

4.1 Ensemble-based framework for fatigue crack prognostics 36
4.1.1 Previous estimation accuracy-based output weight 37
4.1.2 Previous prediction accuracy-based output weight 37
4.1.3 Overall ensemble weight 38

4.2 Case study 39

4.3 Performance evaluation 40

Chapter 5 Background of Multi-Step Ahead Predictions For NPP Time Series Data 45

5.1 Multi-step ahead predictions 45
5.1.1 Introduction 45
5.1.2 Multi-step ahead prediction strategies 46

5.2 Long Short-Term Memory (LSTM) 48

5.3 Tree-structured Parzen Estimator (TPE) hyperparameter optimization 49

5.4 Dropout regularization 50

Chapter 6 A Multi-Step Ahead Prediction Method For NPP Steam Generator Signals 56

6.1 Proposed LSTM-based prognostic framework 56
6.1.1 Data preprocessing 57
6.1.2 Model selection 58
6.1.3 Multi-step ahead prediction 58
6.2 Experimental study 59
6.2.1 SG data 59
6.2.2 Data preprocessing 61
6.3 Results and discussion 61
6.3.1 Automatic hyperparameter optimization 63
6.3.2 Dropout regularization 65
6.3.3 Performance evaluation 65

Chapter 7 A Multi-Step Ahead Prediction Method for NPP Reactor Coolant Pump Signals ... 71

7.1 Hybrid prediction approaches 71
7.2 Signal decomposition methods 72
7.2.1 Empirical Mode Decomposition (EMD) 72
7.2.2 Ensemble Empirical Mode Decomposition (EEMD) 73
7.3 Proposed multi-step ahead prediction method 74
7.3.1 Decomposition of the original time series 74
7.3.2 Multi-step ahead prediction step 75
7.4 Case study: Prediction of the leakage flow of NPP RCPs 77
7.5 Results and discussion 79
7.5.1 Effectiveness of applying the decomposition technique for multi-step ahead prediction 79
7.5.2 Univariate model versus multivariate model 81
7.5.3 Performance evaluation of the LSTM prediction models 83

Chapter 8 Conclusion 87

8.1 Original contributions and limitations 87
8.1.1 Model-based prognostic methods for fatigue crack growth prediction 87
8.1.2 Multi-step ahead prediction methods of NPP time series data 87
8.2 Future research and development perspectives 88

Chapter 1 Introduction

1.1 Prognostics and Health Management (PHM)

Over the last decade, we have been witnessing the rapid and massive growth of the fourth industrial revolution with breakthroughs in many emerging technologies, such as Artificial Intelligence (AI), Big Data and the Internet of Things (IoT). The innovations are changing industrial and production systems, making them more efficient, more flexible and faster to create completely new and innovative products and services. However, no matter how good and reliable components and systems are designed, their degradation over time caused by operating stress and load in the real environment has always been a major concern in all industrial sectors, because if not controlled degradation can lead to failures with potentially catastrophic consequences [1]. In this context, maintenance engineering has played a fundamental role for maintaining the reliability, availability and safety of engineering components and systems during their useful lives [2].

In the old days, maintenance was regarded as repair work. Machines were operated until they broke down, and, then, they were repaired. Maintenance activities were passive (unplanned) and carried out after a failure had already occurred. This is what is known as corrective maintenance. Large downtimes can occur because the replacement or repair of failed units, with significant decreases of the system availability. To overcome this limitation, preventive maintenance has been developed. Differently from corrective maintenance, preventive maintenance is active (planned) and takes place at fixed time intervals, even if a failure has not yet occurred. The key is to decide when to perform the maintenance, so as to reduce the number of failures while avoiding unnecessary maintenance [3].

In the context of preventive maintenance, condition-based maintenance (CBM) has been developed, whereby the decision of when to perform maintenance is taken based on the actual conditions of the components, based on condition monitoring data observed. By so doing, in principle, components and systems can operate as long as they are healthy, repairs or replacements are performed only when needed, so that system availability is increased and maintenance costs are minimized [1]. The development of maintenance engineering is summarized in Fig. 1.1 [4].

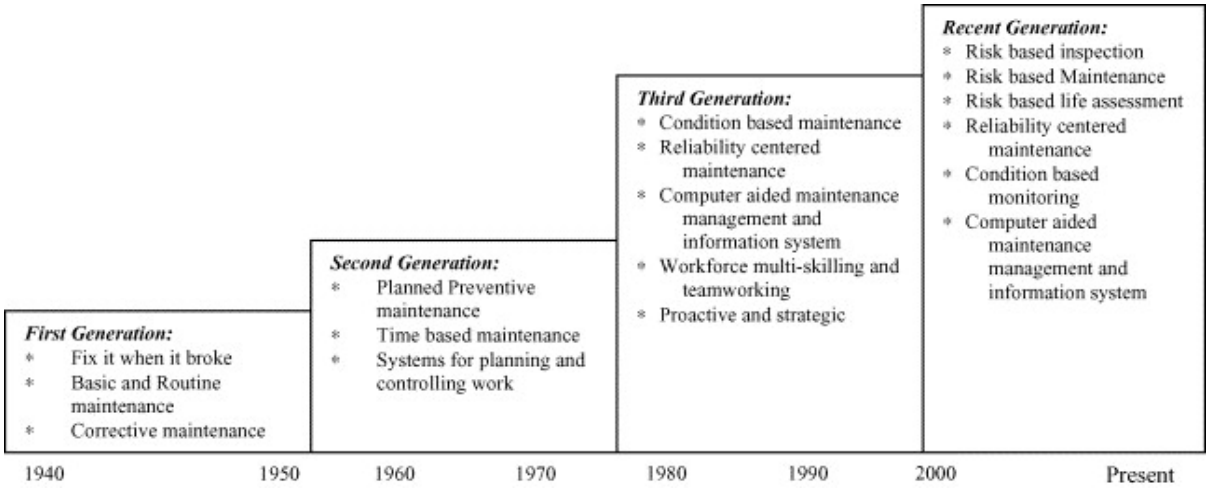


Fig. 1.1 The development of maintenance philosophies [4].

In the years 2000, Prognostics and Health Management (PHM) has been introduced as an evolved form of CBM with a strong focus on prognostics [5]. Since then, PHM has developed to a field of research and application for detecting the degradation of engineering components (fault detection), diagnosing the type of faults (fault diagnostics), predicting the future evolution of the degradation and the Remaining Useful Life (RUL) (prognostics), and proactively managing the failure process [6]. Its strong development is due to the increased capacity in monitoring sensors and the significant advancements in the techniques of signal and data analysis, including data mining and AI, which enable the intelligent reading of the recorded signals and data for fault detection and diagnostics, and failure prediction.

Fig. 1.2 shows an example of the degradation evolution in a component, where an initial defect propagates over time and eventually reaches a critical condition (failure threshold) that causes the component failure. PHM implemented at the present time can provide decision makers with advance failure warning, by the prediction of the degradation evolution in the future and the estimation of the component RUL, which is the time interval between the present time and the predicted failure time. Accurate and reliable predictions provided by PHM allow maintenance actions to be scheduled at the most convenient and inexpensive time, with great benefits on system safety and availability [7].

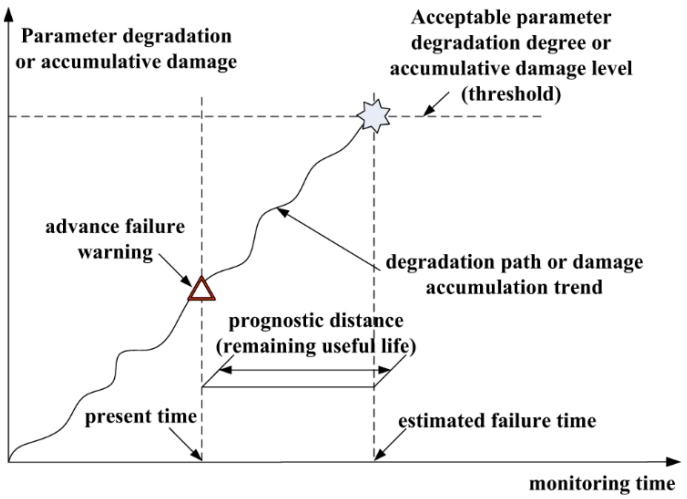


Fig. 1.2. RUL prediction of a degrading component using prognostics [7].

1.2 Prognostic methods

Depending on the information and data available for developing the predictive model, prognostic approaches can generally be classified into two main categories: model-based and data-driven approaches [3].

1.2.1 Model-based prognostics

Model-based approaches utilize the knowledge of the life cycle loading and Physics-of-Failure (PoF) models, control models, or other descriptive models of a component or system to assess its reliability and estimate the RUL. The use of actual operating conditions and PoF models allows model-based approaches to achieve accurate predictions of the degradation-to-failure evolution [7]. However, their applications are typically constrained by the lack of sufficient information, such as explicit PoF models with proper values of the specific parameters of materials, geometry and operating loading, which may be difficult to obtain in practice.

Daigle and Kai [8] have presented a model-based prognostic approach for a pneumatic valve from the Space Shuttle cryogenic refueling system in the aerospace industry and a PoF model of the pneumatic valve was developed based on mass and energy balances in which the damages depend on sliding velocity. Girard et al. [9] have applied model-based prognostics for the Steam Generators (SGs) of Nuclear Power Plants (NPPs), by utilizing the dynamic responses of the Wide Range Level (WRL) parameter to model the evolution of the tube supporting plate clogging. Baraldi et al. [10] have developed a model-based method based on Kalman Filter (KF) and Bootstrapping Aggregating (BAGGING) techniques and applied to turbine creep prognostics. Zhang et al. [11] have used model-based prognostics for the prediction of the RUL of a Proton Exchange Membrane Fuel Cell (PEMFC) stack by using an ensemble of two degradation indicators. Model-based prognostics has also been

applied in the biopharmaceutical industry for predicting the gradual degradation process of the Ethylene Propylene Diene Monomer (EPDM) diaphragms [12].

1.2.2 Data-driven prognostics

Differently from model-based approaches, data-driven approaches make use of historical condition monitoring data to determine data correlations, establish patterns of evolution and learn trends of degradation to failure, by means of statistical and machine learning models, such as Autoregressive Integrated Moving Average (ARIMA), Artificial Neural Network (ANN), Neuro-Fuzzy (NF) and Support Vector Machine (SVM). The advantage of data-driven approaches is their adaptive nature to available condition monitoring data, without the need of pre-specifying PoF models and operating parameters. Therefore, they are appropriate for prognostic applications where PoF models are not available and obtaining condition monitoring data is feasible. For example, a data-driven approach based on Extreme Learning Machine (ELM) has been implemented for the prediction of the degradation of rolling element bearings in rotating machines [13]. Liu et al. [14] have utilized data-driven prognostics for predicting the time series data of NPP Reactor Coolant Pumps (RCPs); the prognostic framework was built based on the ensemble of Probabilistic Support Vector Regression (PSVR) models. Zhang et al. [15] have employed a Long Short-Term Memory (LSTM) recurrent neural network for learning the long-term dependencies among the degraded capacities of lithium-ion batteries and predicting their RULs. To address the problem of different data availability in practical industrial applications, even in presence of mixed information sources, Baraldi et al. [16] have presented a strategy for selecting the model-based and data-driven prognostic approach which best suits the information setting, and applied the method to the prediction of the RUL of turbine blades affected by a developing creep.

1.3 Aims and contributions

Prognostic methods can widely vary for different types of components, failure modes and data available for the model development. The proper selection of prognostic methods for a particular domain plays a fundamental role for the effective implementation of a PHM system. For this reason, this Ph.D. Thesis aims at:

- Investigating the challenges and problems in the implementation of prognostic approaches for industrial components with respect to both model-based and data-driven approaches
- Proposing the appropriate prognostic methods for different domains
- Evaluating the effectiveness of the proposed methods in addressing practical problems of industrial applications.

Correspondingly, the main contributions of the Ph.D. work can be divided into two parts dealing with model-based and data-driven prognostic methods, respectively.

1.3.1 Model-based prognostics

The performance of a model-based prognostic approach depends on the choice of the adopted PoF models [17], [18]. Many researches have indicated that each degradation model has its own specific applicability, appropriate to certain degradation processes under certain operating conditions [19]–[22]. To address this problem, we develop two model-based methods based on the ensemble of multiple degradation models, in order to take advantage of the complementarity of different models, specific on the degradation trends to be predicted. The main contributions of the proposed ensemble of models-based methods are two novel weighted ensemble strategies, which take into account the prediction accuracies of the individual models at previous time instances. In addition, recursive Bayesian filtering and Particle Filtering (PF) are employed to dynamically predict and update the degradation evolution and the component RUL at each prediction step. Eventually, the performances of the proposed methods are validated by case studies of fatigue crack growth simulated with time-varying operating conditions.

1.3.2 Data-driven prognostics

The second main contribution of the Ph.D. work is the development of a data-driven prognostic method for multi-step ahead prediction of time series data of NPP components.

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. Given the long-term horizons of the degradation processes, prognostic capabilities are sought to accurately predict multi-step ahead the components and systems degradation behaviors. 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. Long-term predictions of the components degradation process can also allow deciding whether a component can safely operate until the next planned plant outage, for opportunistic maintenance [23]–[25]. However, multi-step ahead prediction is a difficult task because uncertainty increases 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 have a big influence on the degradation process [26], [27]. For this reason, prognostics in nuclear applications has been limited to short-term prognostics, based on one-step ahead prediction [24], [25], [28]–[30].

In this Thesis, we propose two novel multi-step ahead prediction methods based on LSTM, a deep neural network developed for dealing with the long-term dependencies in time series data. The first proposed method is applied for predicting up to 45 days ahead the values of the operating parameter of NPP SGs. The second proposed method is validated by considering different long-term prediction horizons on a practical case study concerning time series data acquired from RCPs of NPPs. In the proposed methods, additional practical issues are also addressed, including anomaly detection,

automatic hyperparameter optimization and prediction uncertainty quantification.

1.4 Structure of the Thesis

The Thesis is composed of two Parts. Part I consists of eight Chapters which present the introduction, the details of the original contributions, the integral conclusion, and the future research and development perspectives. Part II includes four journal papers, containing the details of the main original contributions. The structure of the Thesis is illustrated in Fig. 1.3.

The rest of Part I is organized as follows:

a) Chapter 2 presents an introduction to the prognostics of fatigue crack growth with model-based method and the approaches used for developing the proposed model-based prognostic methods in Chapters 3 and 4.

b) Chapter 3 (Paper I) presents a model-based prognostic method for fatigue crack growth prediction, based on an integration of recursive Bayesian filtering and a novel weighted ensemble of multiple degradation models. To validate the performance of the proposed method, a case study concerning fatigue crack growth with evolving operating conditions is carried out and the results are compared with those obtained by applying single degradation models.

c) Chapter 4 (Paper II) proposes an improved ensemble method for predicting the evolution to failure and the RUL of a simulated component undergoing fatigue crack growth. A weighted ensemble strategy based on the prediction accuracies of individual models in the previous time steps is proposed. Multiple prognostic performance indicators (PPIs) are employed to validate the prediction capability of the proposed method in a case study concerning multiple fatigue crack trajectories.

d) Chapter 5 introduces the research background of multi-step ahead predictions and the methodologies of LSTM neural networks, Tree-structured Parzen Estimator (TPE) optimization and Dropout regularization, which are employed in Chapters 6 and 7.

e) Chapter 6 (Paper III) presents the proposed multi-step ahead prediction method based on LSTM neural networks and validated in a case study considering SG data acquired from different French NPPs.

f) Chapter 7 (Paper IV) presents a hybrid prediction framework based on Ensemble Empirical Mode Decomposition (EEMD) and LSTM neural networks. The performance of the proposed method is validated by considering different long-term prediction horizons on a practical case study concerning time series data acquired from RCPs of NPPs.

g) Finally, Chapter 8 concludes the Thesis and discusses the future research and development perspectives of our research work.

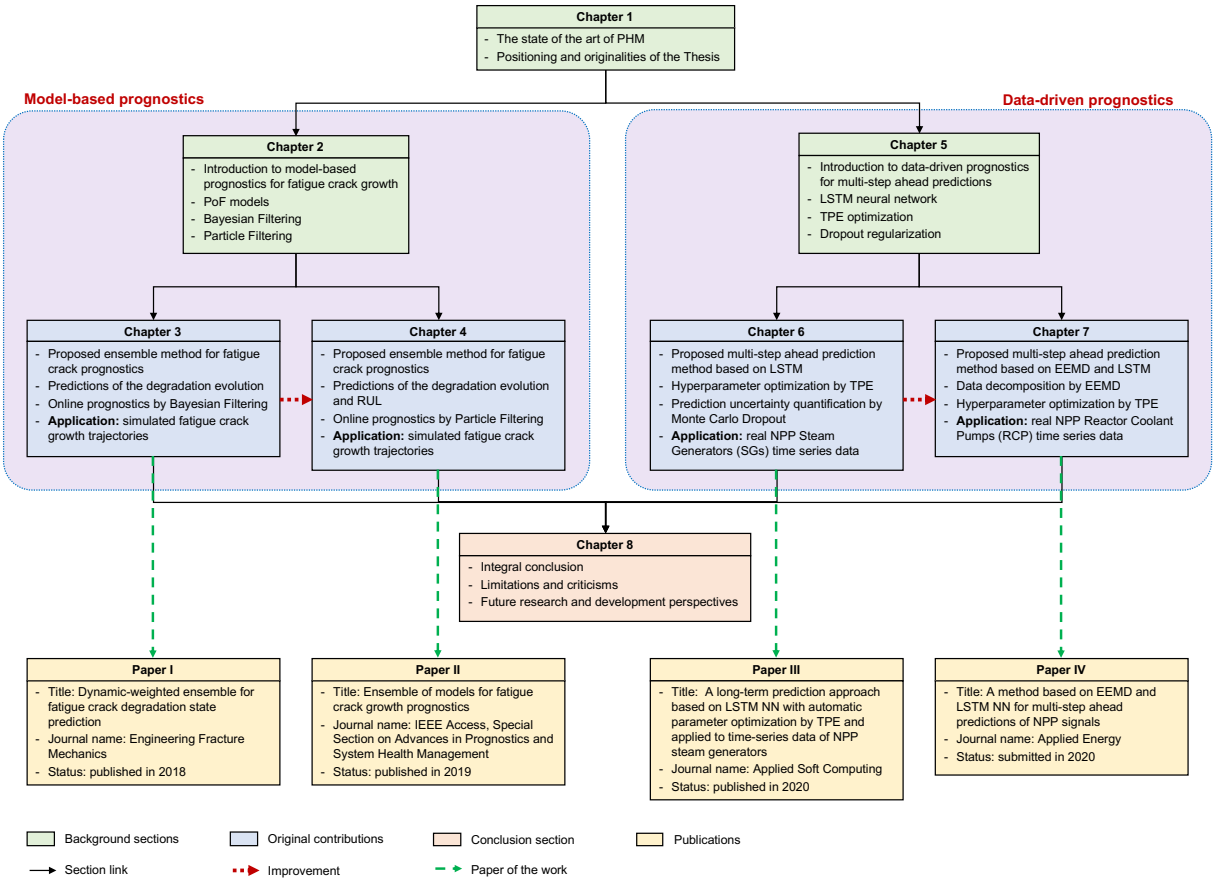


Fig. 1.3. The structure of the Ph.D. Thesis.

References

[1] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*. 2006.

[2] A. Coppola, "Reliability Engineering of Electronic Equipment: A Historical Perspective," *IEEE Trans. Reliab.*, 1984.

[3] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications," *Mech. Syst. Signal Process.*, 2014.

[4] N. S. Arunraj and J. Maiti, "Risk-based maintenance-Techniques and applications," *J. Hazard. Mater.*, 2007.

[5] M. Pecht, "Prognostics and health monitoring of electronics," in *2006 International Conference on Electronic Materials and Packaging, EMAP*, 2006.

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

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

[8] M. J. Daigle and K. Goebel, "A model-based prognostics approach applied to pneumatic valves," *Int. J. Progn. Heal. Manag.*, 2011.

[9] S. Girard, T. Romary, F. Jean-Melaine, P. Stabat, and H. Wackernagel, "Sensitivity analysis and dimension reduction of a steam generator model for clogging diagnosis," *Reliab. Eng. Syst. Saf.*, vol. 113, pp. 143–153, May 2013.

[10] P. Baraldi, F. Mangili, and E. Zio, "A Kalman filter-based ensemble approach with application to turbine creep prognostics," *IEEE Trans. Reliab.*, 2012.

[11] D. Zhang, P. Baraldi, C. Cadet, N. Yousfi-Steiner, C. Bérenguer, and E. Zio, "An ensemble of models for integrating dependent sources of information for the prognosis of the remaining useful life of Proton Exchange Membrane Fuel Cells," *Mech. Syst. Signal Process.*, 2019.

[12] M. Compare, P. Baraldi, I. Bani, E. Zio, and D. McDonnell, "Industrial Equipment Reliability Estimation: a Bayesian Weibull Regression Model with Covariate Selection," *Reliab. Eng. Syst. Saf.*, p. 106891, 2020.

[13] K. Javed, R. Gouriveau, N. Zerhouni, and P. Nectoux, "Enabling health monitoring approach based on vibration data for accurate prognostics," *IEEE Trans. Ind. Electron.*, 2015.

[14] J. Liu, V. Vitelli, E. Zio, and R. Seraoui, "A Novel Dynamic-Weighted Probabilistic Support Vector Regression-Based Ensemble for Prognostics of Time Series Data," *IEEE Trans. Reliab.*, 2015.

[15] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Trans. Veh. Technol.*, 2018.

[16] P. Baraldi, F. Cadini, F. Mangili, and E. Zio, "Model-based and data-driven prognostics under different available information," *Probabilistic Eng. Mech.*, vol. 32, pp. 66–79, 2013.

[17] J. Bogdanoff and F. Kozin, "Probabilistic models of cumulative damage," *Wiley, New York*, Oct. 1985.

[18] W.-F. Wu, "On the Markov approximation of fatigue crack growth," *Probabilistic Eng. Mech.*, vol. 1, no. 4, pp. 224–233, Dec. 1986.

[19] W. F. Wu and C. C. Ni, "Probabilistic models of fatigue crack propagation and their experimental verification," *Probabilistic Eng. Mech.*, vol. 19, no. 3, pp. 247–257, Jul. 2004.

[20] K. Sobczyk and B. F. (Billie F. Spencer, *Random fatigue : from data to theory*. Academic Press, 1992.

[21] M. M. Rocha and G. I. Schuëller, "A probabilistic criterion for evaluating the goodness of fatigue crack growth models," *Eng. Fract. Mech.*, vol. 53, no. 5, pp. 707–731, Mar. 1996.

[22] J. N. Yang and S. D. Manning, "Stochastic crack growth analysis methodologies for metallic structures," *Eng. Fract. Mech.*, vol. 37, no. 5, pp. 1105–1124, Jan. 1990.

[23] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering and System Safety*. 2009.

[24] M. G. Na *et al.*, "Prediction of major transient scenarios for severe accidents of nuclear power plants," *IEEE Trans. Nucl. Sci.*, 2004.

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

Chapter 2 Background of Model-Based Fatigue Crack Prognostics

This Chapter presents an introduction to the prognostics of fatigue crack growth based on model-based approaches. Four popular Physics-of-Failure (PoF) models of fatigue crack growth are presented in Section 2.2. Section 2.3 describes the detailed methods of recursive Bayesian filtering and Particle Filtering (PF) for dynamically estimating the degradation states and predicting the equipment RUL. The four fatigue crack growth models and the two filtering methods will be used for developing our proposed model-based prognostic methods in Chapters 3 and 4.

2.1 Introduction

In practice, the reliability of equipment usually starts decreasing due to gradual degradation, e.g., delamination [1], fatigue crack [2]–[5], corrosion [6], [7], etc., under periodic cyclic loads and eventually leads to failure. Fatigue crack growth is one of the most frequent degradation process affecting components and systems in various major industries, including energy [3], [8], automotive [4], aerospace [5], etc. For this reason, the demand of prognostic systems for dealing with fatigue crack growth has recently increased.

Model-based prognostic approaches utilize the physical knowledge of the degradation for constructing a quantitative analytical model of the equipment behavior and have been applied for fatigue crack growth prognostics [9]–[11]. In [10], a failure prognostic scheme for fatigue crack growth prediction was introduced, which employed a stochastic crack growth model and a Bayesian technique to timely update the equipment degradation state based on a sequence of monitored measurements. Another Bayesian-based prognostic approach was presented to estimate the stress intensive range of the degradation in an online manner [11]; the capability of Bayes theorem was fully exploited for updating the degradation state of the target equipment and estimate the unknown parameters in the physical model, when a new measurement becomes available.

Among Bayesian-based prognostic techniques, a sequential Monte Carlo (SMC) method, known as Particle Filtering (PF) method, has become very popular due to its capability of effectively handling non-linear systems and non-Gaussian noises. The key idea behind this method is to represent the posterior distribution of the equipment state by a random set of weighted samples, also called *particles*, and, then, compute the estimated state based on the particles and their associated weights. This methodology has been used for state estimation and prediction of crack growth [12]–[14], Lithium-ion batteries [15], [16], PEM fuel cells [17], bearings [18], etc.

On the other hand, the performance of model-based prognostic approaches for fatigue crack growth largely depends on the available physics-of-failure model [19], [20]. Models of fatigue crack growth have been extensively developed [2], [21]–[23]. In [21], a comprehensive comparison of stochastic models for fatigue crack growth, including the Markov chain model, the Yang's power law-based model, and a polynomial model, was carried out. The results indicated that each degradation model has its own specific range of applicability, that is, each model is only appropriate to certain degradation processes under certain conditions. To the best knowledge of the authors, there is no general consensus on a prognostic model for fatigue crack growth under different degradation processes. Recently, hybrid and multi-degradation model ensembles have attracted the attention of industrial practitioners and researchers, due to their superiority over individual degradation models in terms of higher accuracy and better generalization capability [16], [24]. The fundamental idea of these empirical frameworks is to exploit the diversity of different degradation models, which can offer complementary information about the degradation states to be estimated. In an application of Lithium-ion battery prognostics, an Interacting Multiple Model Particle Filter (IMMPF) has been presented to combine the estimations from three different battery capacity degradation models [24]. The results experimentally indicated that the ensemble approach can yield a promising performance in terms of smaller estimation errors and more accurate predictions than single models.

2.2 PoF models of fatigue crack growth

2.2.1 Paris-Erdogan model

The widely used Paris-Erdogan model describes the dynamic evolution of the crack depth x as a function of the load cycle number N as follows [25]:

$$\frac{dx}{dN} = C(\Delta K)^m \tag{2.1}$$

where C and m are constants related to the material properties, and ΔK is the Irwin's stress intensity factor defined by [26]:

$$\Delta K = \Delta\sigma\sqrt{\pi x} \tag{2.2}$$

where $\Delta\sigma$ is the cyclic stress amplitude. In practice, the statistical variability of the crack growth rate can be addressed by modifying Eq. (2.1) with an intrinsic process stochasticity [27]:

$$\frac{dx}{dN} = e^{\omega} C(\Delta K)^m \tag{2.3}$$

where $\omega \sim N(0, \sigma_\omega^2)$ is a white Gaussian noise. For a sufficiently small Δt, the Markov chain state-space model of the degradation state x in Eq. (2.3) can be discretized as follows:

$$x_t = x_{t-1} + e^{\omega} C(\Delta K)^m \Delta t \tag{2.4}$$

2.2.2 Polynomial model

The polynomial models were first introduced for fatigue crack growth in order to solve the mismatch between the traditional power function-based models, i.e. Paris-Erdogan, and the practical median crack growth curves [21], [28]:

$$\frac{dx}{dN} = e^{\omega}(a_0 + a_1 x + a_2 x^2) \tag{2.5}$$

where $a_i, i \in [0;2]$ are the second-degree polynomial parameters. Indeed, various works showed that the polynomial models are able to yield the best fit of the linear stage of a degradation process, compared to conventional models [16], [28]. Specifically, the Markov process representation for a polynomial crack growth model can be given as follows:

$$x_t = x_{t-1} + e^{\omega}(a_0 + a_1 x + a_2 x^2)\Delta t \tag{2.6}$$

2.2.3 Global model

Considering again the Paris-Erdogan model Eq. (2.4) and the fact that fatigue crack growth generally depends not only on material properties but also on equipment geometry, a so-called global model was introduced by reformulating the stress intensity factor [29]:

$$\Delta K = h(x)\Delta\sigma\sqrt{\pi x} \tag{2.7}$$

where $h(x)$ denotes the geometric factor of fatigue crack, defined by:

$$h(x) = \alpha_0 + \alpha_1 \frac{x}{d} + \alpha_2 \left(\frac{x}{d}\right)^2 + \alpha_3 \left(\frac{x}{d}\right)^3 \tag{2.8}$$

where $\alpha_i, i \in [0;3]$ and d are geometric coefficients and the width of the specimen, respectively. The global function-based model for fatigue crack growth can be, then, written as follows:

$$x_t = x_{t-1} + e^{\omega} C(h(x)\Delta\sigma\sqrt{\pi x})^m \Delta t \tag{2.9}$$

2.2.4 Curve fitting model

In [29], an empirical crack growth model based on a curve fitting function was presented to address the computational cost problem of the conventional models based on power functions, such as Paris-Erdogan. The curve fitting model is given as follows:

$$\frac{dx}{dN} = e^{\omega}\left(\frac{1}{b_1 x^m + b_2}\right) \tag{2.10}$$

where b_1, b_2 are model constants. The discretized Markov process representation for the model can be given as follows:

$$x_t = x_{t-1} + e^{\omega}\left(\frac{1}{b_1 x^m + b_2}\right)(\Delta K)^m \Delta t \tag{2.11}$$

2.3 Predictions of the degradation evolution by Bayesian filtering algorithms

This Section describes the methods of recursive Bayesian filtering and PF for: 1) estimating the actual degradation state at the present time based on available measurements; 2) predicting the future degradation states over a long-term horizon, in which future measurements are not available.

2.3.1 Estimating the degradation state at the present time

A. Recursive Bayesian filtering

Consider a dynamic system process which describes the evolution of the degradation state x_t and the measurement z_t at time *t* as:

$$x_t = f_t(x_{t-1}, \omega_{t-1}) \tag{2.12}$$

$$z_t = g_t(x_t, v_t) \tag{2.13}$$

where ω_t is the state noise sequence, and v_t is the measurement noise sequence at time *t*.

The system state x_t can be estimated by constructing its posterior probability density function (PDF) $p(x_t \mid z_{1:t})$ via two consecutive steps, namely *prediction* and *update*. In the prediction step, the previous state estimation x_{t-1} and the state transition model f_t are utilized to obtain the prior distribution of the system state x_t at the current time *t* by using the Chapman-Kolmogorov equation:

$$\begin{aligned} p(x_t \mid z_{1:t-1}) &= \int p(x_t \mid x_{t-1}, z_{1:t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \\ &= \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \end{aligned} \tag{2.14}$$

where $p(x_t \mid x_{t-1})$ is the conditional probability distribution and is defined by the state model in Eq. (2.12). In the update step, a new measurement z_t is collected and the posterior distribution of the current state x_t is obtained by updating the prior distribution via Bayes theorem:

$$p(x_t \mid z_{1:t}) = \frac{p(x_t \mid z_{1:t-1}) p(z_t \mid x_t)}{p(z_t \mid z_{1:t-1})} \tag{2.15}$$

where $p(z_t \mid x_t)$ is the likelihood function defined by the measurement model in Eq. (2.13) and $p(z_t \mid z_{1:t-1})$ is a normalizing constant given by:

$$p(z_t \mid z_{1:t-1}) = \int p(x_t \mid z_{1:t-1}) p(z_t \mid x_t) dx_t \tag{2.16}$$

B. Particle Filtering

However, no general consensus has been reached on the analytical solution of Eqs. (2.14) and (2.15) [16]. To address this problem, PF was developed as an improved Bayesian filtering algorithm, which utilizes Monte Carlo (MC) simulation to approximate the true probability distribution with a set of weighted particles $\{x_t^i, w_t^i, i = 1,..,N_P\}$, where N_p is the total number of particles [30]. These random particles evolve statistically independently of each other, according to the probabilistic state model Eq. (2.12). Then, the posterior distribution at time *t* can be approximated as:

$$p(x_t \mid z_{1:t}) \approx \sum_{i=1}^{N_P} w_t^i \delta\left(x_t - x_t^i\right) \tag{2.17}$$

where $\delta(.)$ is the Dirac Delta function. The particle x_t^i is sampled from the importance sampling distribution $q(x_t \mid z_{1:t})$ with a weight w_t^i computed as:

$$w_t^i = \frac{p(z_{1:t} \mid x_t^i) p(x_t^i)}{q(x_t^i \mid z_{1:t})} \tag{2.18}$$

By setting $q(x_t \mid z_{1:t}) = p(x_t \mid x_{t-1})$, the particle weight w_t^i can be updated with the new measurement z_t as follows:

$$w_t^i = w_{t-1}^i p(z_t \mid x_t^i) \tag{2.19}$$

where $p(z_t \mid x_t^i)$ is the likelihood of measurement z_t given the particle x_t^i. It is important to note that the weights are normalized as $\sum_i w_t^i = 1$.

2.3.2 Predicting the future degradation evolution and the RUL

Once the posterior distribution $p(x_t \mid z_{1:t})$ of the current degradation state is estimated, it is possible to predict the future degradation evolution and the RUL of the equipment. However, note that there is no available information for estimating the likelihoods of the future degradation states, because future measurements $z_{t+l}, l \in [1; T-t]$, where *T* is the time horizon of interest for the analysis, have not been collected yet. The only available information is the dynamic state model Eq. (2.12). Then, the *l*-step ahead posterior distribution $p(x_{t+1} \mid z_{1:t})$ can be written as follows:

$$p(x_{t+l} \mid z_{1:t}) = \int \ldots \int \prod_{j=t+1}^{t+l} p(x_j \mid x_{j-1}) p(x_t \mid z_{1:t}) \prod_{j=t}^{t+l-1} dx_j \tag{2.20}$$

The numerical evaluation of the integrals in Eq. (2.20) requires significant computational effort. In this work, an approach presented in [31] is adopted with the assumption that the particle weights do not change from time *t* to time *t+l*, i.e., $w_t^j = w_{t+1}^j = ... = w_{t+l}^j$. Accordingly, the predicted distribution at time *t+l* is given by:

$$p(x_{t+l} \mid z_{1:t}) \approx \sum_{i=1}^{N_P} w_t^j \delta(x_{t+l} - x_{t+l}^i) \tag{2.21}$$

where the particle x_{t+l}^i is obtained by iteratively applying the state model Eq. (2.12) to the corresponding particle of the current state x_t^i.

Finally, the RUL associated to each particle at the present time *t* can be calculated with reference to the earliest time that the degradation state exceeds the failure threshold x_{th}:

$$\hat{RUL}_t^i = \left\{ (T_t^i - 1 - t) \middle| g(x_{T_t^i - 1}, p_t^i, v_t) < x_{th}, g(x_{T_t^i}, p_t^i, v_t) \geq x_{th} \right\} \tag{2.22}$$

where T_t^i is obtained by simulating the particle evolution via the state model Eq. (2.12). The predicted RUL distribution is, then, given by:

$$p(RUL \mid z_{1:t}, x_i < x_{th}) \approx \sum_{i=1}^{N_P} w_t^j \delta(\hat{RUL}_t - \hat{RUL}_t^i) \tag{2.22}$$

More details can be found in [32], [33].

References

[1] H. Nykyforchyn, O. Zvirko, O. Tsyrulnyk, and N. Kret, "Analysis and mechanical properties characterization of operated gas main elbow with hydrogen assisted large-scale delamination," *Engineering Failure Analysis*, 2017.

[2] J. N. Yang and S. D. Manning, "Stochastic crack growth analysis methodologies for metallic structures," *Eng. Fract. Mech.*, vol. 37, no. 5, pp. 1105–1124, Jan. 1990.

[3] T. Prusek, E. Moleiro, F. Oukacine, A. Adobes, M. Jaeger, and M. Grandotto, "Deposit models for tube support plate flow blockage in Steam Generators," *Nucl. Eng. Des.*, vol. 262, pp. 418–428, Sep. 2013.

[4] H. K. Yang, V. Doquet, and Z. F. Zhang, "Fatigue crack growth in two TWIP steels with different stacking fault energies," *Int. J. Fatigue*, vol. 98, pp. 247–258, 2017.

[5] M. A. Haile, J. C. Riddick, and A. H. Assefa, "Robust Particle Filters for Fatigue Crack Growth Estimation in Rotorcraft Structures," *IEEE Transactions on Reliability*, vol. 65, no. 3. pp. 1438–1448, 2016.

[6] F. Li, Y. Qu, and J. Wang, "Bond life degradation of steel strand and concrete under combined corrosion and fatigue," *Eng. Fail. Anal.*, vol. 80, pp. 186–196, 2017.

[7] A. Vazdirvanidis, G. Pantazopoulos, and A. Rikos, "Corrosion investigation of stainless steel water pump components," *Engineering Failure Analysis*, 2016.

[8] G. Yang, V. Pointeau, E. Tevissen, and A. Chagnes, "A review on clogging of recirculating steam generators in Pressurized-Water Reactors," *Prog. Nucl. Energy*, vol. 97, pp. 182–196, May 2017.

[9] M. E. Orchard and G. J. Vachtsevanos, "A particle-filtering approach for on-line fault diagnosis and failure

prognosis," *Trans. Inst. Meas. Control*, vol. 31, no. 3–4, pp. 221–246, 2009.

[10] F. Cadini, E. Zio, and D. Avram, "Model-based Monte Carlo state estimation for condition-based component replacement," *Reliab. Eng. Syst. Saf.*, vol. 94, no. 3, pp. 752–758, Mar. 2009.

[11] B. A. Zárate, J. M. Caicedo, J. Yu, and P. Ziehl, "Bayesian model updating and prognosis of fatigue crack growth," *Eng. Struct.*, vol. 45, pp. 53–61, Dec. 2012.

[12] P. Baraldi, M. Compare, S. Sauco, and E. Zio, "Ensemble neural network-based particle filtering for prognostics," *Mech. Syst. Signal Process.*, vol. 41, no. 1–2, pp. 288–300, 2013.

[13] M. E. Orchard, F. A. Tobar, and G. J. Vachtsevanos, "Outer Feedback Correction Loops in Particle Filtering-based Prognostic Algorithms: Statistical Performance Comparison," *Stud. Informatics Control*, vol. 18, no. 4, pp. 295–304, 2009.

[14] L. Tang, J. Decastro, G. Kacprzynski, K. Goebel, and G. Vachtsevanos, "Filtering and prediction techniques for model-based prognosis and uncertainty management," in *2010 Prognostics and System Health Management Conference, PHM '10*, 2010.

[15] Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht, "Remaining useful life prediction of lithium-ion battery with unscented particle filter technique," *Microelectron. Reliab.*, vol. 53, no. 6, pp. 805–810, 2013.

[16] Y. Xing, E. W. M. Ma, K.-L. Tsui, and M. Pecht, "An ensemble model for predicting the remaining useful performance of lithium-ion batteries," *Microelectron. Reliab.*, vol. 53, no. 6, pp. 811–820, Jun. 2013.

[17] M. Jouin, R. Gouriveau, D. Hissel, M. C. Péra, and N. Zerhouni, "Prognostics of PEM fuel cell in a particle filtering framework," *Int. J. Hydrogen Energy*, vol. 39, no. 1, pp. 481–494, 2014.

[18] B. Zhang, C. Sconyers, C. Byington, R. Patrick, M. E. Orchard, and G. Vachtsevanos, "A probabilistic fault detection approach: Application to bearing fault detection," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 2011–2018, 2011.

[19] J. Bogdanoff and F. Kozin, "Probabilistic models of cumulative damage," *Wiley, New York*, Oct. 1985.

[20] W.-F. Wu, "On the Markov approximation of fatigue crack growth," *Probabilistic Eng. Mech.*, vol. 1, no. 4, pp. 224–233, Dec. 1986.

[21] W. F. Wu and C. C. Ni, "Probabilistic models of fatigue crack propagation and their experimental verification," *Probabilistic Eng. Mech.*, vol. 19, no. 3, pp. 247–257, Jul. 2004.

[22] K. Sobczyk and B. F. (Billie F. Spencer, *Random fatigue : from data to theory*. Academic Press, 1992.

[23] M. M. Rocha and G. I. Schuëller, "A probabilistic criterion for evaluating the goodness of fatigue crack growth models," *Eng. Fract. Mech.*, vol. 53, no. 5, pp. 707–731, Mar. 1996.

[24] X. Su, S. Wang, M. Pecht, L. Zhao, and Z. Ye, "Interacting multiple model particle filter for prognostics of lithium-ion batteries," *Microelectron. Reliab.*, vol. 70, pp. 59–69, 2017.

[25] P. Paris and F. Erdogan, "A Critical Analysis of Crack Propagation Laws," *J. Basic Eng.*, vol. 85, no. 4, p. 528, Dec. 1963.

[26] G. R. Irwin, "Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate," *J. Appl. Mech.*, 1957.

[27] E. Myötyri, U. Pulkkinen, and K. Simola, "Application of stochastic filtering for lifetime prediction," *Reliab. Eng. Syst. Saf.*, vol. 91, no. 2, pp. 200–208, Feb. 2006.

[28] C. C. Ni, "Formulation of a Polynomial Stochastic Fatigue Crack Growth Model," *Adv. Mater. Res.*, vol. 909, pp. 467–471, 2014.

[29] H. Salimi, M. Pourgol-Mohammad, and S. Kiad, "Assessment of Stochastic Fatigue Failures Based On Deterministic Functions," in *13th International conference on Probabilistic Safety Assessment and Management (PSAM 13)*, 2016.

[30] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[31] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[32] P. Baraldi, F. Cadini, F. Mangili, and E. Zio, "Model-based and data-driven prognostics under different

available information," *Probabilistic Eng. Mech.*, vol. 32, pp. 66–79, 2013.

[33] E. Zio and G. Peloni, "Particle filtering prognostic estimation of the remaining useful life of nonlinear components," *Reliab. Eng. Syst. Saf.*, vol. 96, no. 3, pp. 403–409, 2011.

Chapter 3 An Ensemble Method For Predicting Fatigue Crack Degradation

In this Chapter, a prognostic framework for fatigue crack growth is proposed by integrating a recursive Bayesian filtering technique and a dynamic ensemble of models. The degradation state of the component is estimated based on the condition monitoring data collected until the current load cycle, and short-term degradation state prediction is performed to anticipate and proactively prevent sudden breakdowns of the component. The main contribution of the work is the dynamic ensemble which combines different Physics-of-Failure (PoF) models of fatigue crack growth with dynamic weights. The dynamic weights are computed based on the historical estimation error for a predefined number of the latest load cycles. To our knowledge, this ensemble framework has been here developed and applied for the first time for a prognostic problem of fatigue crack growth. To validate the performance of the proposed framework, a case study concerning fatigue crack growth with evolving operation conditions is carried out and the results are compared with those obtained by applying single degradation models.

3.1 Proposed ensemble framework for fatigue crack growth prediction

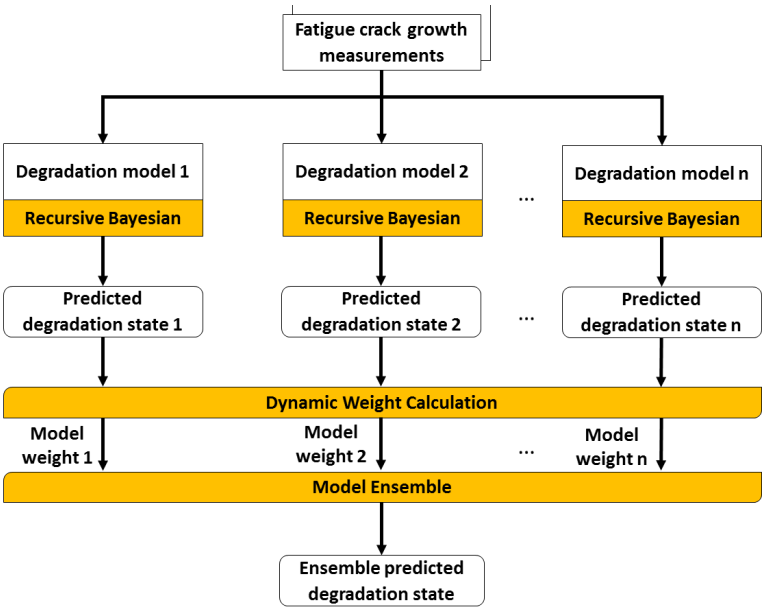


Fig. 3.1. Flow chart of the proposed prognostic framework.

In this Section, a weighted ensemble framework is presented to dynamically find the optimal combination of different crack growth models with respect to their estimation performances. Fig. 3.1 illustrates the flow chart of the proposed prognostic framework, consisting of the following three steps:

a) At the load cycle *t*, when a new measurement is available, the model parameters and the estimated degradation states given by each model are updated by using the recursive Bayesian filtering algorithm described in Chapter 2, Section 2.3.1.A.

b) The estimated degradation state of each individual model is weighted by their estimation errors for the last cycles as follows:

$$w_t^i = \frac{(\phi_t^i)^{-2}}{\sum_i (\phi_t^i)^{-2}} \tag{3.1}$$

where w_t^i and ϕ_t^i are the dynamic weight and estimation error coefficient of the *i*th model at time *t*, respectively. The estimation error coefficient ϕ_t^i is computed:

$$\phi_t^i = \frac{1}{\delta} \sum_{k=t-\delta}^{t} (z_k - \hat{x}_k^i)^2 \tag{3.2}$$

where δ is the time horizon (δ = 50 load cycles in the case study that follows) and $\hat{x}_k^i$ is the estimated degradation state of the *i*th model at time *k*. Thus, the highest weight is given to the model in the ensemble with lowest error at the present time *t*, whereas the smallest weight in constructing the ensemble is assigned to the least accurate model at time *t*.

c) Once the dynamic weights for all models at the current time *t* are calculated, the predicted degradation state of the ensemble is computed as follows:

$$\tilde{x}_t = \sum_i^{N_M} (\hat{x}_t^i \times w_t^i) \tag{3.3}$$

where $\tilde{x}_t$ is the ensemble predicted state at time *t* and N_M is the number of degradation models.

The model diversity plays an important role in deciding the generalization performance of an ensemble framework. For this reason, four different stochastic PoF models are considered to develop our ensemble model for fatigue crack growth prognostics, including: Paris-Erdogan, polynomial, global, and curve fitting models, as described in Chapter 2, Section 2.2.

The proposed ensemble framework is applied for estimating the degradation states at the current time *t*, i.e. state regression, and at fixed predictions horizons $t + \tau$, i.e. short-term state prediction where τ is the prediction horizon. In the following case study, three scenarios are considered for validating the proposed method, including: 1) degradation state regression at the current time *t*; 2) short-term prediction at time *t*+100; and 3) short-term prediction at time *t*+300. It is important to note that for short-term predictions, the weights of individual models in the ensemble at prediction horizons $t + \tau$ are kept the same as the ones at the current time *t* because there is no measurement available to update

the weights at the future times.

3.2 Illustrative case study of fatigue crack growth

Numerical simulations of fatigue crack growth have been carried out with an initial crack length x_0 of 10^{-4} mm and a test frequency of 1 Hz. The total number of fatigue load cycles is $N = 2000$ cycles. To explore the capabilities of the proposed approach under time-varying degradation processes, the fatigue lifetime of the simulated crack growth is split into four continuous and equivalent periods, where the crack length is generated according to different crack growth models as follows:

a) In the first 500-load-cycle period, the Paris-Erdogan model is employed to simulate the crack propagation process as described in Chapter 2, Eq. (2.4). In this regard, the Paris-Erdogan model can provide a linear relationship between $\log(dx/dN)$ and $\log(\Delta K)$, in other words, the stress intensity factor (SIF) range of simulated data is compatible to the Region II (Paris region).
b) In the following period, the polynomial crack growth model in Chapter 2, Eq. (2.6) is used to generate the crack length.
c) The global function-based crack growth model in Chapter 2, Eq. (2.9) is, then, employed for the third period.
d) Finally, in the last period, from the cycle 1501 to 2000, the curve fitting function-based crack growth model in Chapter 2, Eq. (2.11) is utilized.

The model parameters in this work are first initialized based on empirical knowledge, as detailed in Table 3.1. Subsequently, a Bayesian-based parameter identification approach is applied to adaptively update the parameters according to the real-time information from measured data at each load cycle.

Table 3.1 Detailed settings of model characteristics of fatigue crack growth case studies.

State noise variance	$\sigma_\omega^2 = 0.49$
Measurement noise variance	$\sigma_v^2 = 0.16$
Paris-Erdogan model	$C = 0.1$, $m = 1.3$
Polynomial model	$p_0 = 1.4 \times 10^{-3}, p_1 = 1.5 \times 10^{-3}, p_2 = 1 \times 10^{-5}$
Global function-based model	$C = 0.005$, $m = 0.245$
Curve fitting function-based model	$w = 1$ mm, $C_1 = 250$, $C_2 = 0.3$, $m = -0.7$

For the purpose of extensively validating the effectiveness of the proposed approach for drifting degradation processes, two crack growth profiles under different conditions of load ratio, $R = 0.1$ and $R = 0.15$, are artificially integrated to expand the case study of the simulated crack growth, as illustrated in Fig. 3.2. Specifically, the load ratio R, or the stress ratio, measures the general influence of the mean

stress on the fatigue crack growth behavior, and is defined by the ratio of the minimum to the maximum stresses experienced during a cycle. As the ratio R increases, the fatigue crack growth rate curve is shifted toward higher dx/dN [1].

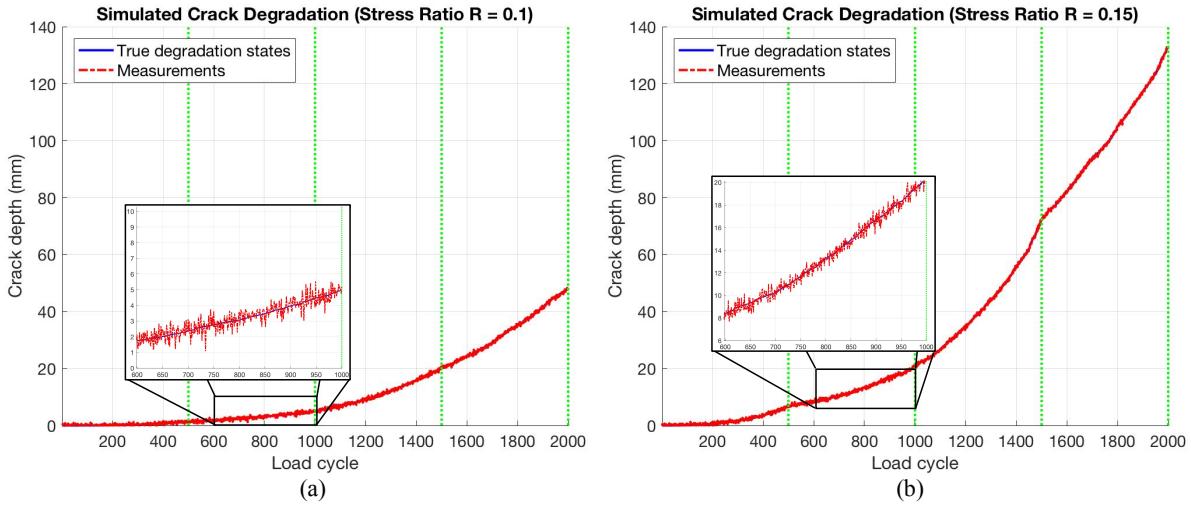


Fig. 3.2. Simulated crack depth evolution profiles with different load ratios (a) R = 0.1 and (b) R = 0.15.

3.3 Results and discussion

The proposed prognostic framework based on recursive Bayesian technique and dynamic-weighted ensemble is applied to determine the best combination of multiple crack growth models in terms of degradation prediction performance. More specifically, when a new monitored measurement is available, the degradation states and crack growth model parameters are estimated online via the Bayesian technique. The four stochastic crack growth models described in Chapter 2, Section 2.2 are considered. The weight for each individual model is updated at the current load cycle and the ensemble is obtained by integrating the individual degradation models. The mean square error (MSE) is considered as the performance evaluation index to indicate prediction accuracy:

$$MSE_i = \frac{1}{N}\sum_{t=1}^{N}(x_t - \hat{x}_t^i)^2 \tag{3.4}$$

where x_t and $\hat{x}_t^i$ denote the true degradation state and the prediction of the ith crack growth model at time t, respectively.

The degradation state estimations at different load ratios are shown in Figs. 3.3 and 3.4. As expected, the estimated degradation states among different models are similar for the current time, as shown in Figs. 3.3(a) and 3.4(a), but obviously separated for short-term state predictions. This indicates the effectiveness of the prediction error-based dynamic weights in reflecting the performance of different degradation models. An interesting observation in Figs. 3.3 and 3.4 is that the polynomial model can exhibit satisfactory performance when the fatigue crack depth is small, but its performance is rapidly degraded when the crack becomes longer. This is mainly because of the fact that the

polynomial function only fits in the linear least square stage of the degradation process, as mentioned in Chapter 2, Section 2.2.2. In other words, the polynomial model is able to achieve satisfactory performance only for linear and deterministic fatigue crack growth processes. In contrast, by combining dynamically different degradation models, the proposed ensemble approach can achieve superior performance to the individual models in predicting the degradation states of fatigue crack growth. Table 3.2 shows that the proposed ensemble framework outperforms the individual crack growth models, yielding a prediction accuracy of 2.07 and 33.14 in terms of MSE for short-term degradation state prediction at time t+300 under the load ratios R = 0.1 and R = 0.15, respectively.

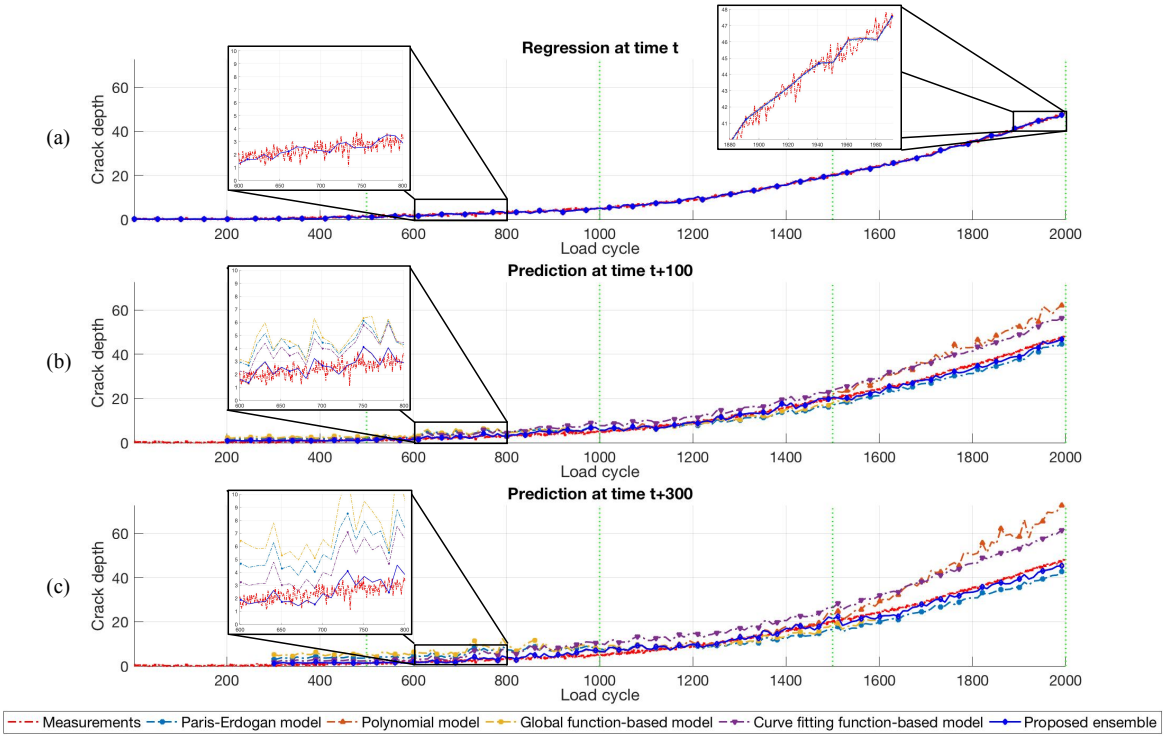


Fig. 3.3. Estimation of degradation state at load ratio R = 0.1 in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

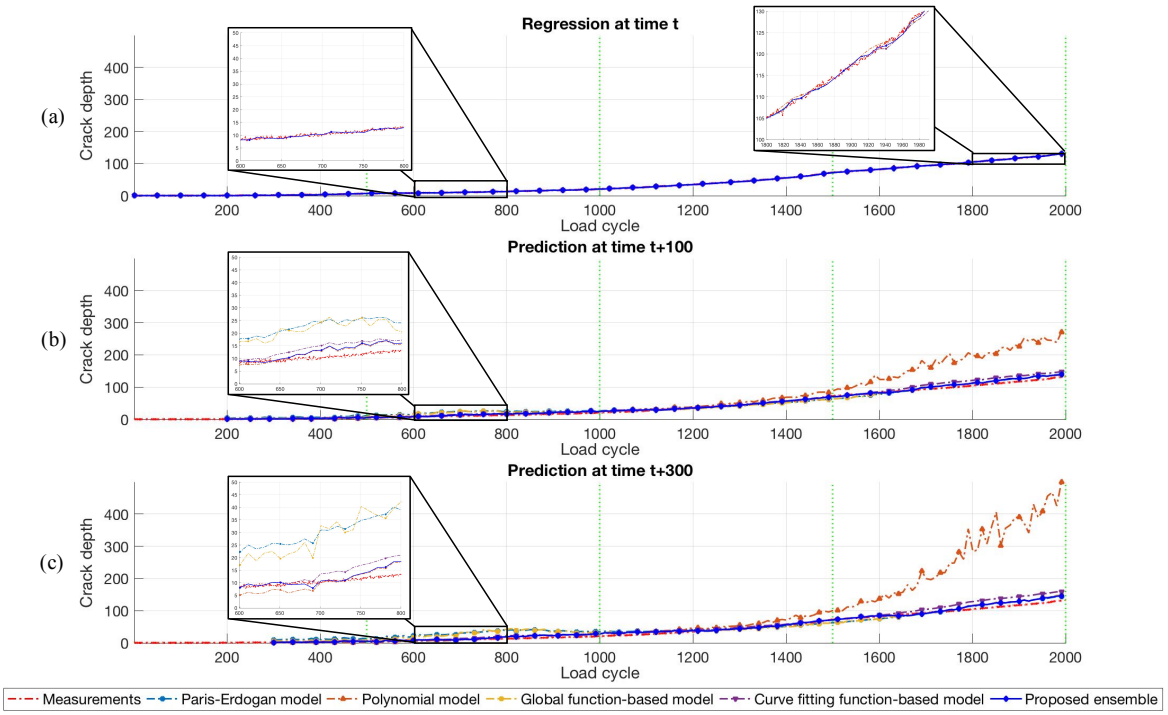


Fig. 3.4. Estimation of degradation state at load ratio R = 0.15 in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

Table 3.2 MSE results of the degradation state regression and short-term prediction at different load conditions.

	R = 0.1			R = 0.15		
	t	t+100	t+300	t	t+100	t+300
Paris-Erdogan model	0.10	1.09	12.79	0.15	10.35	151.04
Polynomial function-based model	0.10	4.51	60.90	0.15	278.25	9764.72
Global function-based model	0.10	0.69	11.92	0.15	8.94	140.21
Curve fitting function-based model	0.10	3.54	42.03	0.15	14.33	119.09
Proposed ensemble	0.10	0.38	2.07	0.12	5.23	33.14

The impact of uncertainty on the performance of the proposed framework in case of unknown initial state of degradation has also been investigated. In this case, the monitored data are assumed unavailable from time 1 to t_0 (t_0 = 500 in this study), and the true degradation state of the system x_{t_0} is also assumed unknown. The performance of the proposed approach under different load ratio conditions is shown in Figs. 3.5 and 3.6. The dashed line with marker are the predicted degradation states of the proposed approach while the dotted lines are the 95% confidence intervals. The results in Figs. 3.5 and 3.6 show that the proposed framework can yield accurate state predictions even without knowledge of the initial degradation state. In Fig. 3.6(c), some abnormal spikes in the confidence intervals can be observed.

Note that these abnormal spikes exist only in the case study in which the degradation state prediction is performed at time t+300 with no available measurement until the time t_0 = 500. For the relative shorter prediction horizon, no spike is observed. This can be explained by the fact that for a long-term prediction time t+300, the performances of individual models in the ensemble can be unexpectedly degraded due to the propagation of uncertainty. As can be seen in Figs. 3.7 and 3.8 below, in the last 400 load cycles, the performance diversity between the polynomial model and the others is clearly observed, and, furthermore, the variance of the polynomial model also rapidly increases, resulting in unsatisfactory performance in the estimation of the confidence intervals.

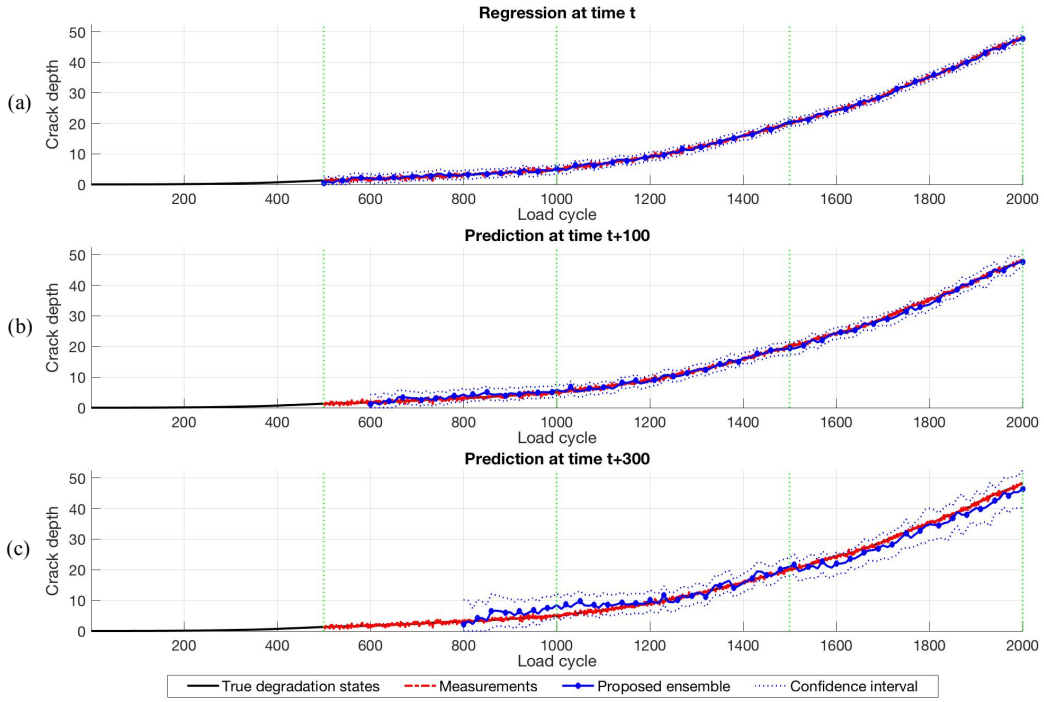


Fig. 3.5. Estimation of degradation state with measurements that are not available until the time t_0 = 500 at load ratio R = 0.1, in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

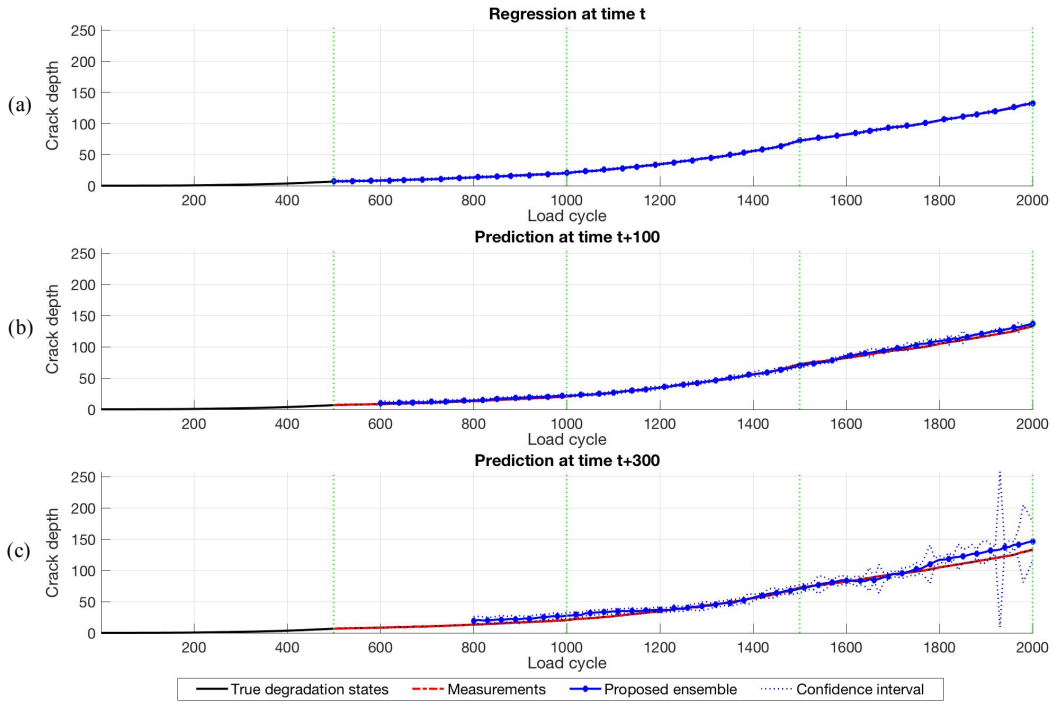


Fig. 3.6. Estimation of degradation state with measurements that are not available until the time t_0 = 500 at load ratio R = 0.15, in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

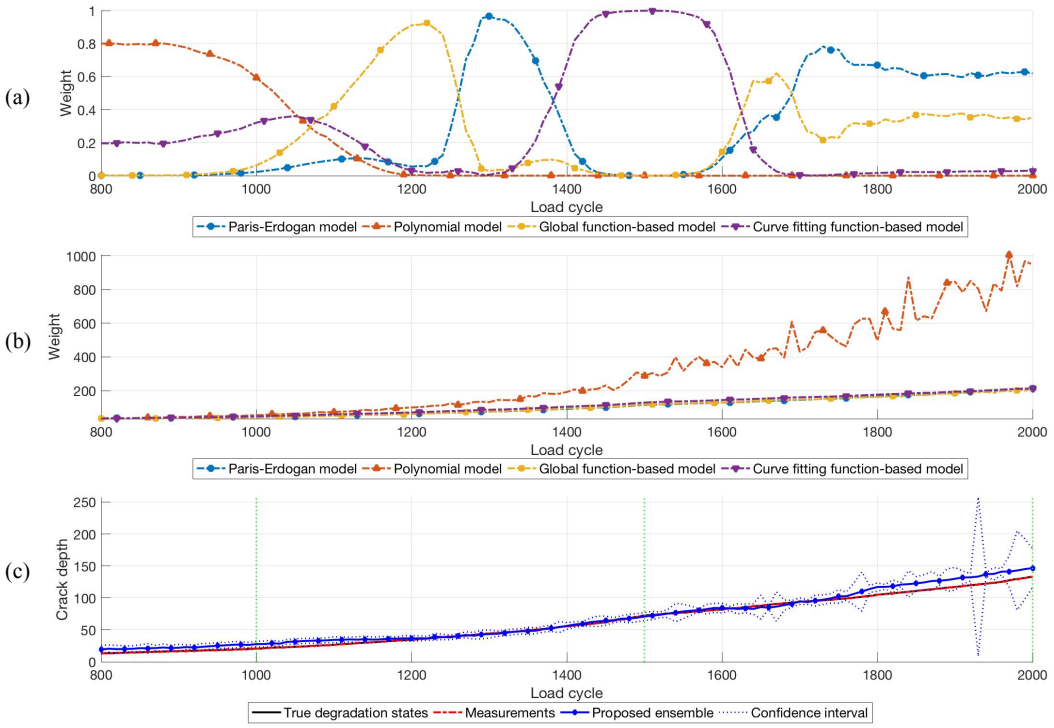


Fig. 3.7. Degradation state prediction at time t+300 with measurements that are not available until the time t_0 = 500 at load ratio R = 0.15: (a) Ensemble weights of individual models; (b) Degradation state prediction of individual models; (c) Degradation state prediction of the proposed ensemble.

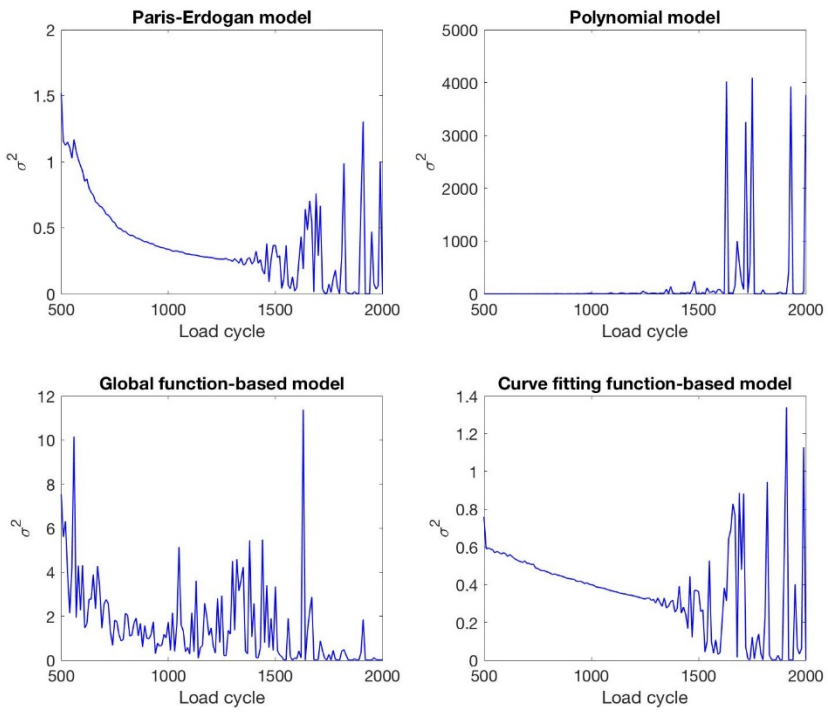


Fig. 3.8. The variance of degradation state predictions at time t+300 with measurements that are not available until the time t_0 = 500 at load ratio R = 0.15.

References

[1] Gdoutos, E. (2005). *Fracture Mechanics*. Dordrecht: Springer.

Chapter 4 An Ensemble Method Based On Particle Filtering for Fatigue Crack Prognostics

In this Chapter, an improved ensemble framework is presented for predicting the evolution to failure and the Remaining Useful Life (RUL) of an equipment undergoing fatigue crack growth. To maximize diversity in the ensemble, four stochastic degradation models of fatigue crack growth are considered. Moreover, Particle Filtering (PF) is used to track the crack propagation process with nonlinear and non-Gaussian characteristics, and eventually to predict the RUL of the equipment before breakdown. To further enhance the performance of the proposed framework, a dynamic weighted ensemble strategy is proposed, based on the previous accuracy performance in degradation state estimation and RUL prediction of each single model in the ensemble. Finally, a set of prognostic performance indicators (PPIs) is employed to validate the prediction capability of the proposed framework in a case study concerning multiple fatigue crack degradation processes.

4.1 Ensemble-based framework for fatigue crack prognostics

The flow chart of proposed ensemble-based framework for fatigue crack prognostics is illustrated in Fig. 4.1. In this framework, we aim at addressing three main issues: 1) how to select the degradation models for the ensemble; 2) how to use the degradation models for estimating the degradation states and predicting the RUL of the equipment; 3) how to combine the outputs of the individual models for achieving maximum accuracy. The details of the proposed framework are given as follows.

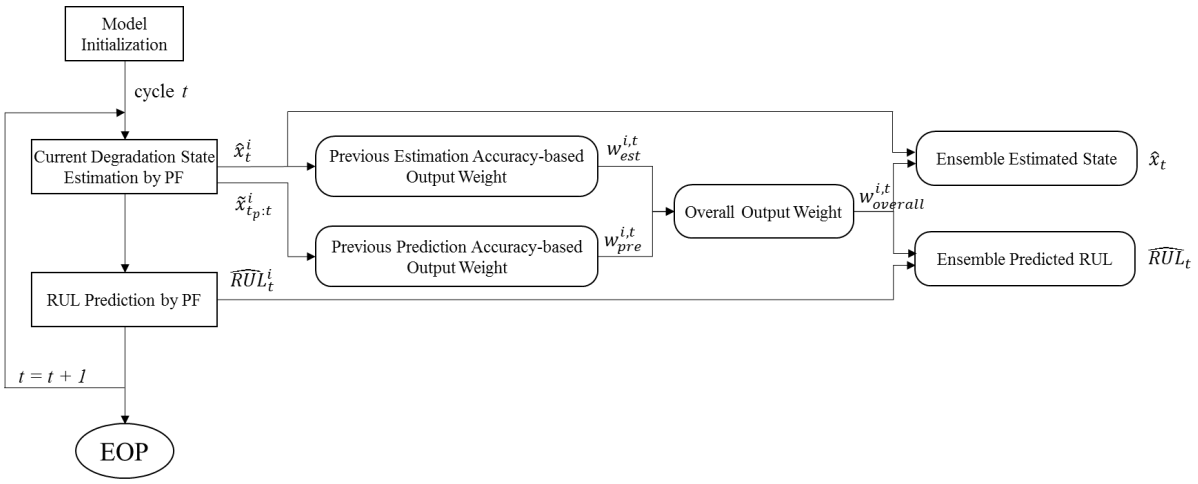


Fig. 4.1. Flow chart of the proposed prognostic framework.

With respect to the way of calculating the weights of the models in an ensemble, ensemble approaches can generally be classified into three categories: 1) simple vote ensemble [1], where all individual models outputs are given the same weight coefficients in the voting strategy; in this approach, majority vote is the most popularly used rule; 2) weighted ensemble [2], which combines individual models with different weight coefficients, which means that each individual is assumed to have a different contribution to the performance of the ensemble model; 3) selective ensemble [3], which includes only an optimal subset of models. Recently, the selective ensemble approach has attracted increasing interest, due to its capability of significantly reducing the bias and variance in the ensemble estimation [3].

In this work, we propose a selective ensemble approach for prognostics of fatigue crack growth based on a Best-Worst Weighted Vote (BWWV) strategy. A novel ensemble weight constructed by using both previous estimation and prediction accuracies of each individual model in the population is presented.

4.1.1 Previous estimation accuracy-based output weight

Consider a sequence of measurements collected until the current time t, $\{z_j, j=1,...,t\}$, the degradation states described by individual models, $\{\hat{x}_j^i, i=1,...,N_M, j=1,...,t\}$, where N_M is the number of individual models ($N_M=4$ in this work), can be estimated by using PF as described in Chapter 2, Section 2.3.1.B. The weight coefficients of individual models are computed as the Root Mean Square Error (RMSE) of their previous estimates with respect to the corresponding measurements:

$$\varepsilon_t^i = \sqrt{\frac{1}{\delta_{est}} \sum_{k=t-\delta_{est}}^{t} (z_k - \hat{x}_k^i)^2} \tag{4.1}$$

where δ_{est} is the time horizon of previous estimates considered ($\delta_{est}=50$ load cycles in the case study that follows). The previous estimation accuracy-based output weight of each single model is, then, obtained based on the BWWV as follows:

$$w_{est}^{j,t} = 1 - \frac{\varepsilon_t^i - \varepsilon_t^{\min}}{\varepsilon_t^{\max} - \varepsilon_t^{\min}} \tag{4.2}$$

where $\varepsilon_t^{\min} = \min_i\{\varepsilon_t^i\}$ and $\varepsilon_t^{\max} = \max_i\{\varepsilon_t^i\}$. Specifically, the maximum weight $w_{est}^t = 1$ is assigned to the model in the ensemble with the highest accuracy at the time t, whereas the null weight $w_{est}^t = 0$ is given to the least accurate model, equivalent to temporarily removing it from the ensemble at time t.

4.1.2 Previous prediction accuracy-based output weight

Because there is no measurement available to predict the future states, the prediction accuracy of each model in the ensemble for the previous time steps is used to calculate the corresponding output

weight. We first identify a time instant t_p before the present time t in the time horizon, where $t = t_p + \delta_{pre}$ ($\delta_{pre} = 100$ load cycles in the following case study), as illustrated in Fig. 4.2. The state prediction $\tilde{x}_{t_p:t}$ (the dashed line) of one model at time t_p is obtained by iteratively applying the system model to the estimated state $\tilde{x}_{t_p}$, which is set to z_{t_p} in this study. The weight coefficients of individual models are computed as the RMSE of their predictions for degradation states between time t_p and t with respect to the measurements:

$$\varepsilon_t^i = \sqrt{\frac{1}{\delta_{pre}} \sum_{k=t_p}^{t} (z_k - \tilde{x}_k^i)^2} \tag{4.3}$$

Thus, the previous prediction accuracy-based output weights of individual models are computed as:

$$w_{pre}^{i,t} = 1 - \frac{\varepsilon_t^i - \varepsilon_t^{\min}}{\varepsilon_t^{\max} - \varepsilon_t^{\min}} \tag{4.4}$$

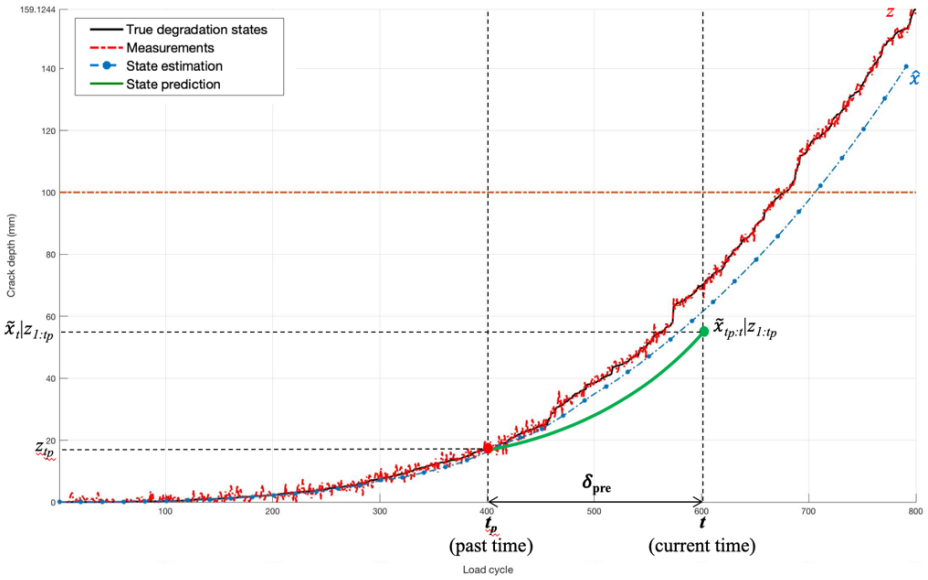


Fig. 4.2. An illustration for calculating the previous prediction accuracy-based output weight.

4.1.3 Overall ensemble weight

The complete ensemble weights of individual models are computed as the average of the previous estimation accuracy-based and previous prediction accuracy-based weights:

$$w_{overall}^{i,t} = \frac{w_{est}^{i,t} + w_{pre}^{i,t}}{2} \tag{4.5}$$

and are normalized in the range [0; 1]:

$$\tilde{w}_{overall}^{i,t} = \frac{w_{overall}^{i,t}}{\sum_{i}^{N_M} w_{overall}^{i,t}} \tag{4.6}$$

Finally, the predictions of the degradation state and RUL of the ensemble framework are computed:

$$\hat{x}_t = \sum_{i=1}^{N_M} \hat{x}_t^i \times \tilde{w}_{overall}^{i,t} \tag{4.7}$$

$$\widehat{RUL}_t = \sum_{i=1}^{N_M} \widehat{RUL}_t^i \times \tilde{w}_{overall}^{i,t} \tag{4.8}$$

where $\hat{x}_t$ and $\widehat{RUL}_t$ are the predictions of the degradation state and RUL of the ensemble framework at time t, respectively, and $\widehat{RUL}_t^i$ is the RUL predictions of individual models, which are obtained by using PF as described in Chapter 2, Section 2.3.2.

4.2 Case study

In this work, a case study considering 100 simulated degradation trajectories of fatigue crack growth is carried out, as shown in Fig. 4.3. The Paris-Erdogan Physics-of-Failure (POF) model as described in Chapter 2, Eq. (2.4) is employed to generate the crack depth trajectories with the parameters predefined as follows:

- The model constants are $C = 0.1$ and $m = 1.3$.
- The state and measurement noise variances are $\sigma_\omega^2 = 1.10$ and $\sigma_v^2 = 2.25$, respectively.
- The initial crack depth is 10^{-4} mm.

The crack depth measurements are recorded at every load cycle. A failure threshold x_{th} is set to 100 mm. The fatigue simulation for each degradation trajectory is performed with a total 800 load cycles.

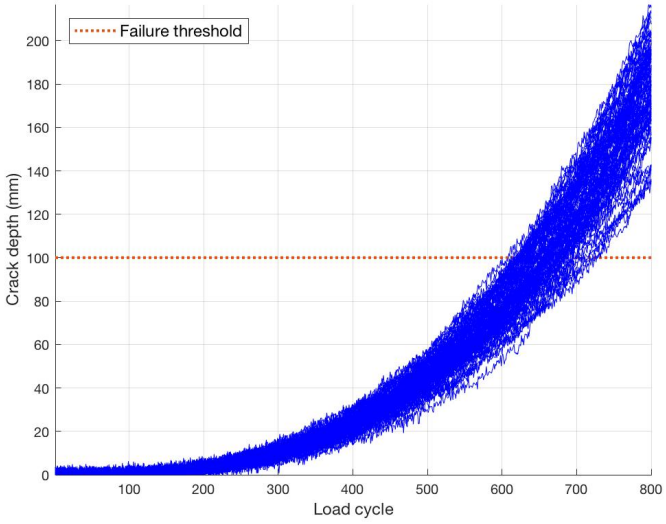


Fig. 4.3. 100 fatigue crack growth degradation trajectories.

4.3 Performance evaluation

In this Section, the performance of the proposed ensemble-based prognostic framework is validated in predicting the degradation evolution and the RUL of the component. The results are compared with four single PoF models of fatigue crack growth with respect to five widely used PPIs, including Timeliness Weighted Error Bias (TWEB), Sample Mean Error (SME), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), and Sample Median Error (SMeE). The definitions of the PPIs are given in Appendix 1 of this Chapter.

When a new measurement is collected, the degradation state estimations of individual models are updated by using PF as described in Chapter 2, Section 2.3.1.B. Fig. 4.4 shows the estimation results of the four PoF models for the 1st simulated degradation trajectory, as mentioned in Section 4.2, over 800 load cycles. The four single model shows distinctive characteristics in different stages of the degradation evolution of the fatigue crack, which is suitable for the diversity of the proposed ensemble framework.

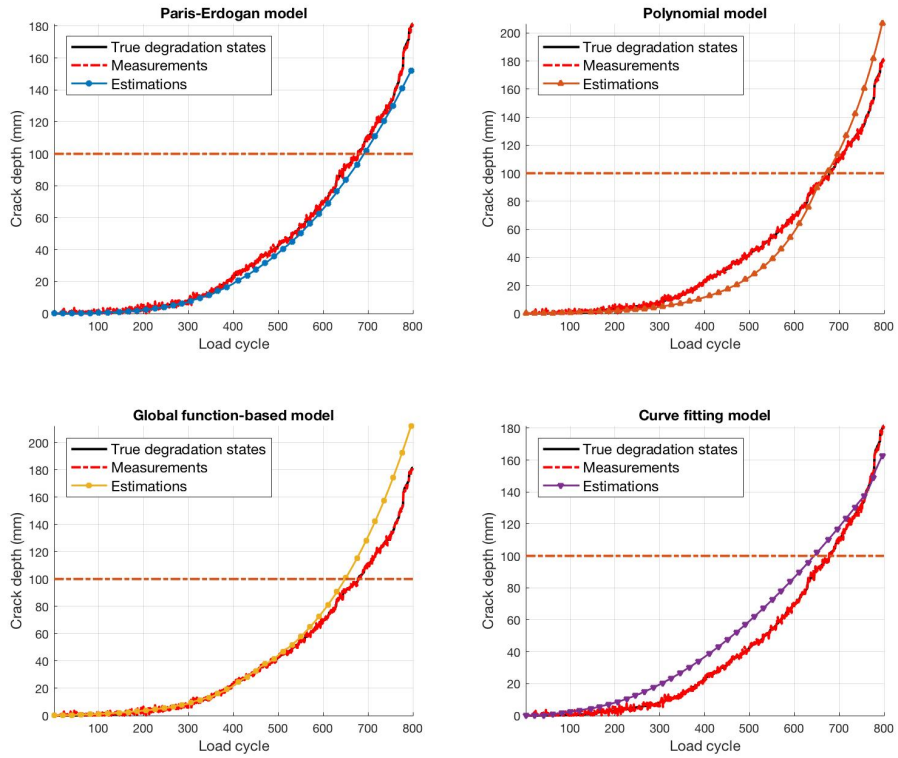


Fig. 4.4. State estimation results obtained by the individual PoF models for the 1st degradation trajectory.

Based on the estimations of the individual models, the ensemble weights are computed and used to update the predictions of the degradation state and RUL of the proposed ensemble framework, as shown

in Figs. 4.5 and 4.6, respectively. In Fig. 4.6, the individual PoF models are quite inaccurate in predicting the RUL throughout the whole time horizon, which can be explained due to their low accuracies in predicting the degradation states, as shown in Fig. 4.5. On the other hand, the proposed framework shows a superior performance and obtain the predicted RUL close to the true one.

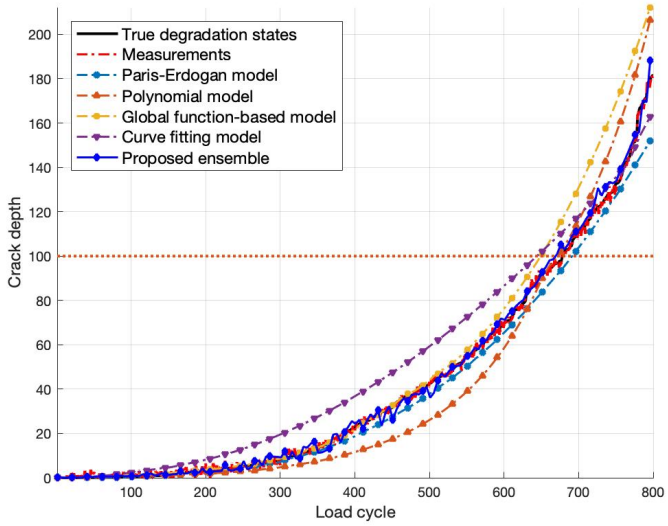


Fig. 4.5. State estimation results obtained by all of the models for the 1st degradation trajectory.

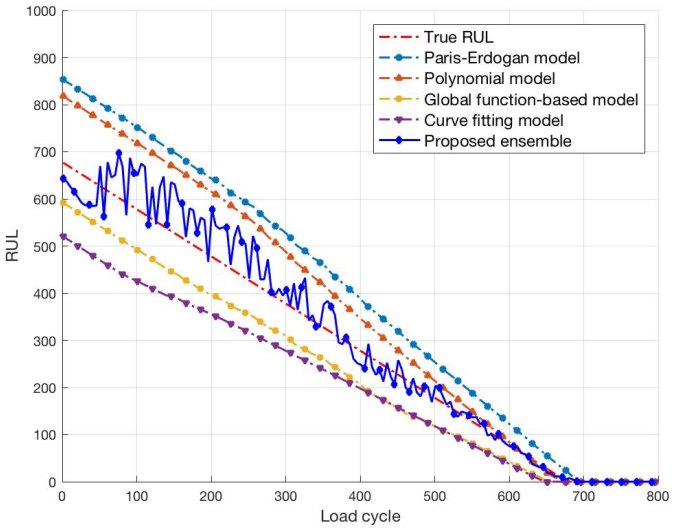


Fig. 4.6. RUL prediction results obtained by all of the models for the 1st degradation trajectory.

To further validate the performance of the proposed ensemble framework, four scenarios are randomly chosen. The results of the proposed framework are shown in Figs. 4.7 and 4.8. As shown in these Figures, the proposed framework shows satisfactory performance in accurately predicting the equipment crack growth trend and the equipment RUL. In addition, in Fig. 4.8, the confidence intervals show that the RUL prediction accuracy of the proposed approach is improved with more available data.

This can be explained by the fact that the proposed approach which benefits from the diverse accuracy of the individual models by a weighting scheme that can adaptively select the best set of models.

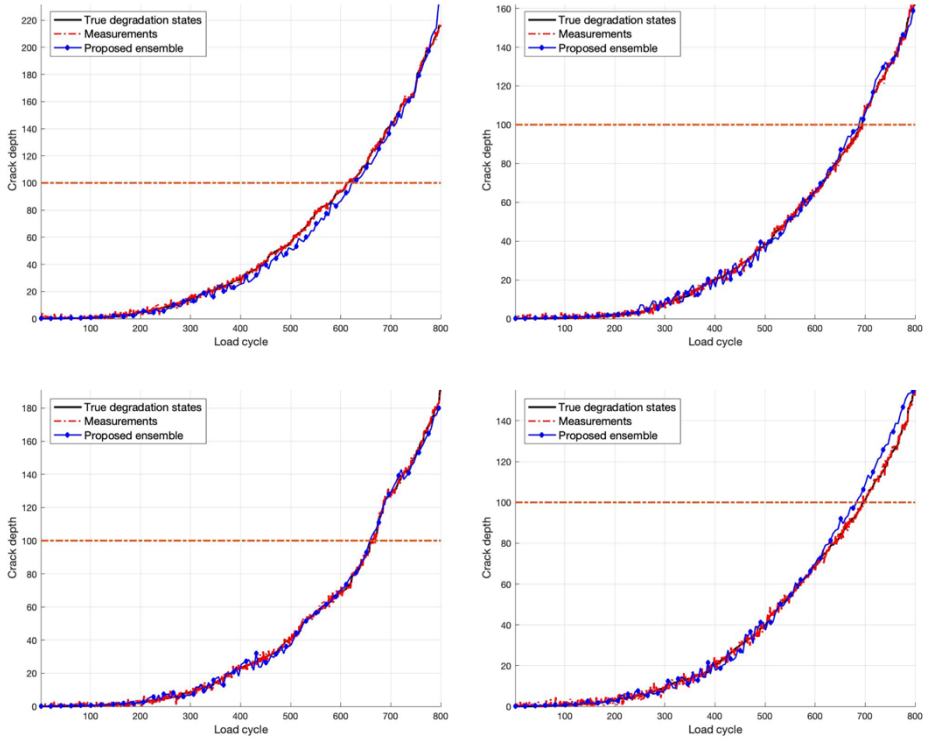


Fig. 4.7. Degradation state estimation using the proposed ensemble with different available measurements.

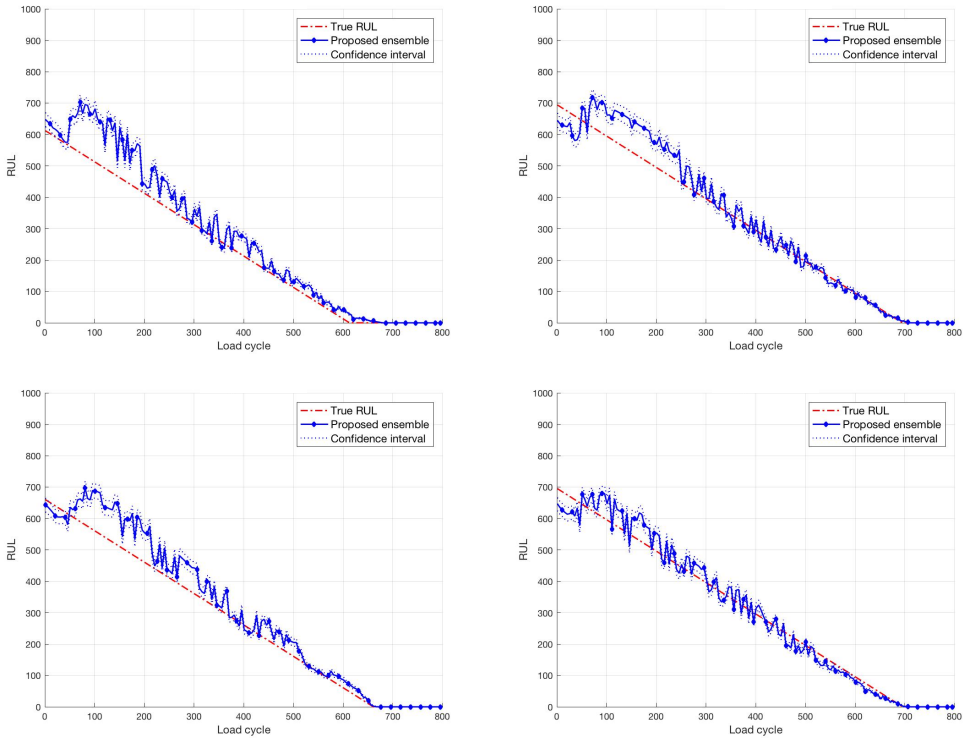


Fig. 4.8. RUL prediction using the proposed ensemble with different available measurements.

Tables 4.1 and 4.2 present the average performances in terms of the predictions of the degradation states and RUL, which have been calculated over 100 crack depth growth scenarios. The results clearly show that the proposed ensemble framework consistently outperforms the individual models for all of the prognostic metrics.

Table 4.1 Performance comparison for the estimations of degradation states.

	Paris-Erdogan	Polynomial	Global function	Curve fitting	Proposed ensemble
Avg. MSE (std)	117.72 (102.68)	166.30 (80.39)	138.64 (74.91)	102.90 (69.38)	8.85 (5.04)

Table 4.2 Performance comparison for the RUL predictions.

	TWEB	SME	MAPE	MSE	SMeE
Paris-Erdogan	0.09	115.25	0.62	18.28×10^3	114.63
Polynomial	0.07	85.68	0.37	11.56×10^3	85.43
Global function	0.02	45.79	0.20	3.11×10^3	45.86
Curve fitting	0.03	65.18	0.23	7.01×10^3	64.18
Proposed ensemble	0.01	29.41	0.16	3.03×10^3	31.81

Appendix 1. Detailed definitions of the PPIs.

Formula	Description
1. Timeliness weighted error bias (TWEB) $TWEB = \frac{1}{N_S}\sum_{j=1}^{N_S}\phi\left(\sum_{t=1}^{T_j}\gamma_{j,t}\frac{\hat{RUL}_{j,t}-RUL_{j,t}}{T_j}\right)$ $\phi(y)=\begin{cases}\exp\left(\frac{\lvert y\rvert}{e_1}\right)-1, & for\ y<0\\ \exp\left(\frac{\lvert y\rvert}{e_2}\right)-1, & for\ y\geq 0\end{cases}$ $e_1>e_2>0$	Measure the weighted prediction error over the lifetime T_j by using a penalty function $\phi(y)$ and a weighting function $\gamma_{j,t}$. $\gamma_{j,t}$ is defined as a Gaussian kernel function with a mean value T_j and a standard deviation $0.5T_j$. The optimal value for TWEB is 0, which indicates that the predicted RUL is centered on the true one. Higher values of TWEB indicate a great discrepancy between the predicted RUL and the true one.
2. Sample mean error (SME) $SME=\left\lvert\frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{T_j}\sum_{t=1}^{T_j}\left(\hat{RUL}_{j,t}-RUL_{j,t}\right)\right\rvert$	Calculate the average errors of all sample points during the lifetime T_j. The optimal value for SME is 0, which indicates that the average errors of all samples is 0, that is, the predicted RUL is centered on the true one. Higher values of SME indicate a great discrepancy between the predicted RUL and the true one.
3. Mean absolute percentage error (MAPE) $MAPE=\frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{T_j}\sum_{t=1}^{T_j}\left\lvert\frac{\hat{RUL}_{j,t}-RUL_{j,t}}{RUL_{j,t}}\right\rvert$	Measure the average absolute percentage error of all samples throughout the lifetime T_j. The optimal value for MAPE is 0, which indicates a negligible error for all samples during their lifetime. Higher values of MAPE indicate a great discrepancy between the predicted RUL and the true one.
4. Mean square error (MSE)	Take into account the average quadratic error of the predicted RUL of all samples during the lifetime T_j. The optimal value for MSE is 0, which indicates that the predicted RUL is equal

$MSE = \frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{T_j}\sum_{t=1}^{T_j}\left(\hat{RUL}_{j,t} - RUL_{j,t}\right)^2$	to the true one for all samples. Higher values of MSE indicate high errors in the predicted RUL.
5. Sample median error (SMeE) $SME = \left\| \underset{j=1,\ldots,N_S}{median}\left(\frac{1}{T_j}\sum_{t=1}^{T_j}\left(\hat{RUL}_{j,t} - RUL_{j,t}\right)\right)\right\|$	Exploit the absolute median of average errors of all samples over the lifetime T_j. The optimal value for SMeE is 0, which indicates that the median error of all samples is zero. Higher values of SMeE indicate that most predicted RULs are wrong.

References

[1] D. W. Opitz and J. W. Shavlik, "Actively Searching for an Effective Neural Network Ensemble," *Conn. Sci.*, vol. 8, no. 3–4, pp. 337–354, Dec. 1996.

[2] X. Su, S. Wang, M. Pecht, L. Zhao, and Z. Ye, "Interacting multiple model particle filter for prognostics of lithium-ion batteries," *Microelectron. Reliab.*, vol. 70, pp. 59–69, 2017.

[3] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, no. 1–2, pp. 239–263, May 2002.

Chapter 5 Background of Multi-Step Ahead Predictions For NPP Time Series Data

This Chapter introduces the research background and literature review of multi-step ahead prediction approaches, following by the detailed methodologies of Long Short-Term Memory (LSTM) neural network, Tree-structured Parzen Estimator (TPE) hyperparameter optimization and Dropout regularization approaches. These approaches will be applied in the development of our two proposed multi-step ahead prediction methods for the real time series data of different Nuclear Power Plant (NPP) components, as given in Chapters 6 and 7.

5.1 Multi-step ahead predictions

5.1.1 Introduction

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 processes, and the prediction horizon, i.e. how far into the future the model should predict and with what accuracy [1]. 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 horizons of months [2]–[4]. Despite its importance, multi-step ahead prediction remains a difficult task of PHM, because uncertainty increases 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 have a big influence on the degradation process in complex systems [1], [5]. Large prediction uncertainty has limited the development of prognostics in nuclear applications to only short-term prognostics, based on one-step ahead prediction [3], [4], [6]–[8].

In general, multi-step ahead prediction models can be classified as statistical or machine learning

approaches [9], [10]. 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 [11]. 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 predefining the model form [12]. The most widely used machine learning approaches for multi-step ahead predictions are Support Vector Regression (SVR) [13]–[16], Artificial Neural Network (ANN) [9], [10], [17]–[19], Neuro-Fuzzy [5], [20], [21] and Recurrent Neural Network (RNN) [22]–[24]. 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 [25]. 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 different fields, such as the forecasting of wind speed [26]–[31], solar energy [32]–[34], air quality [35]–[37], stock market [38], [39], electricity and gas demand [40]–[42], and oil and petroleum production [43], [44].

5.1.2 Multi-step ahead prediction strategies

Given a univariate time series of the observations collected up to time t, $\{x_1, x_2, ..., x_t\}$, the main goal is to predict the H next observations $\{\hat{x}_{t+h}\}, h \in [1, H]$, which can be formulated as below:

$$\{\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H}\} = f(x_t, x_{t-1}, ..., x_{t-d+1}), \tag{5.1}$$

where f is the prediction model and d is the embedding dimension (or the number of lagged values).

Depending on the desired horizon H, a prediction method can be classified into short-, medium-, or long-term prediction. As aforementioned, the further in the future one attempts to predict, the harder it is to achieve an accurate prediction due to the increasing uncertainty and accumulation of errors. To address this problem, there are three popular prediction strategies, namely recursive, direct and MIMO predictions, which are described as follows [5].

A. Recursive prediction strategy

The recursive strategy attempts to train a model focused solely on one-step ahead prediction:

$$\hat{x}_{t+1} = f_R(x_t, x_{t-1}, ..., x_{t-d+1}) \tag{5.2}$$

where f_R is the one-step ahead prediction model.

After the model is trained, the predictions are recursively estimated. In other words, intermediate predictions are used as inputs for predicting next values until the prediction at the time horizon H, $\hat{x}_{t+H}$, is obtained:

$$\begin{aligned}
\hat{x}_{t+1} &= f_R(x_t, x_{t-1}, ..., x_{t-d+1}) \\
\hat{x}_{t+2} &= f_R(\hat{x}_{t+1}, x_t, ..., x_{t-d+2}) \\
&\vdots \qquad\qquad \vdots \\
\hat{x}_{t+H} &= f_R(\hat{x}_{t+H-1}, \hat{x}_{t+H-2}, ..., \hat{x}_{t+H-d+1})
\end{aligned} \tag{5.3}$$

An advantage of the recursive strategy is its low computational cost since only one single model is required for training. However, the prediction errors of the previous steps can easily accumulate in the next predictions, resulting in the decrease of accuracy in the long run. Besides, this prediction strategy does not take into account the data dependencies among time steps.

B. Direct prediction strategy

In contrast to the recursive strategy which uses a single model, the direct strategy [45] constructs a set of *H* different models for different time steps and the same input data are used for feeding all the models as below:

$$\begin{aligned}
\hat{x}_{t+1} &= f_{D,1}(x_t, x_{t-1}, ..., x_{t-d+1}) \\
\hat{x}_{t+2} &= f_{D,2}(x_t, x_{t-1}, ..., x_{t-d+1}) \\
&\vdots \qquad\qquad \vdots \\
\hat{x}_{t+H} &= f_{D,H}(x_t, x_{t-1}, ..., x_{t-d+1})
\end{aligned} \tag{5.4}$$

where $f_{D,h}$ is the direct prediction model tuned to perform the prediction $\hat{x}_{t+h}$ at time $t+h, h \in [1, H]$.

In the direct strategy, each prediction model is trained and dedicated to a certain horizon, so the error accumulation can be avoided. However, training different prediction models will greatly increase the prediction complexity and time consumption, and, like the recursive strategy, the direct strategy does not take into account the dependencies among time-series observations.

C. MIMO prediction strategy

Unlike the recursive and direct approaches, the MIMO approach is a multiple output strategy, in which the output of the prediction model is a vector of future values predicted by using only one model [46]:

$$\{\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H}\} = f_{MIMO}(x_t, x_{t-1}, ..., x_{t-d+1}) \tag{5.5}$$

where f_{MIMO} is the multiple output prediction model. In this sense, the objective function during the model training is to simultaneously minimize the prediction errors on different horizons. By so doing, the MIMO strategy can preserve the temporal stochastic dependencies of sequential data, addressing the limitation of the recursive and direct approaches. On the other hand, the computational cost of the

MIMO approach is less than that of the direct approach because it requires only one model to be trained.

5.2 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 [25]. An LSTM network consists of a chain of repeating memory modules (Fig. 5.1).

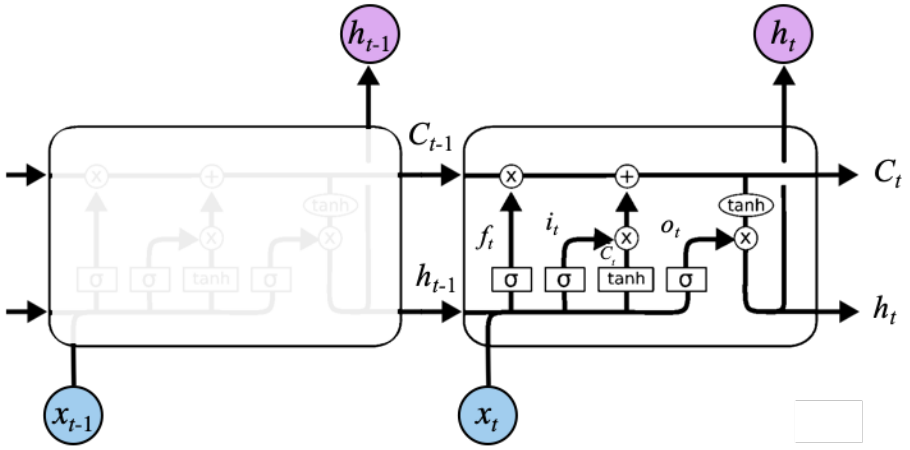


Fig. 5.1. The structure of LSTM repeating memory modules [47].

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:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.6}$$

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:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.7}$$

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:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5.8}$$

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:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5.9}$$

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

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5.10}$$

$$h_t = o_t * \tanh(C_t) \tag{5.11}$$

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

5.3 Tree-structured Parzen Estimator (TPE) hyperparameter optimization

Automatic hyperparameter optimization plays a fundamental role in the development of machine learning models, especially when deep neural networks such as LSTM [48] 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 [49], [50]. In this study, we apply Tree-structured Parzen Estimator (TPE) [51], 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:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2} \tag{5.12}$$

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 [51].

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:

$$p(\theta \mid y) = \begin{cases} \mathrm{Pr}_G(\theta) & if\ y < y^* \\ \mathrm{Pr}_B(\theta) & if\ y \geq y^* \end{cases} \tag{5.13}$$

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

$$EI(\theta) = \frac{\mathrm{Pr}_G(\theta)}{\mathrm{Pr}_B(\theta)} \tag{5.14}$$

And the hyperparameter configuration θ^* 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. 5.2 describes the overall flowchart of the TPE algorithm, where N_{opt} denotes the number of TPE iterations.

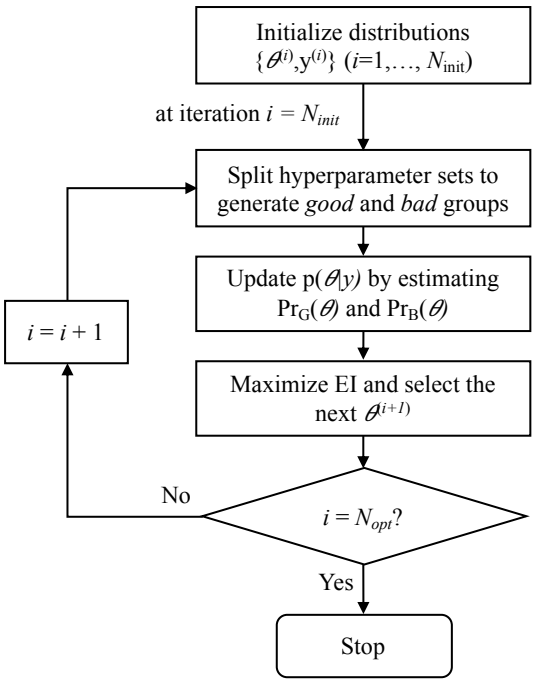


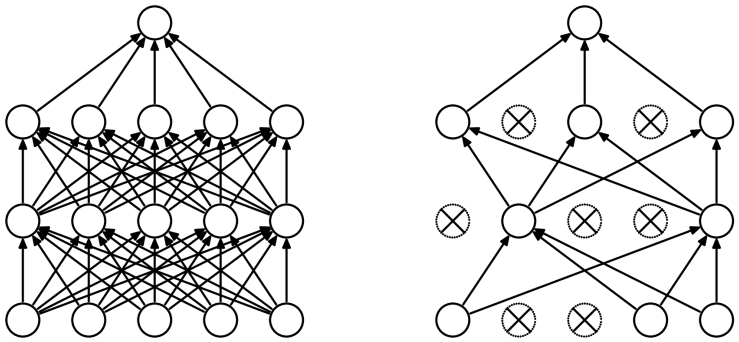
Fig. 5.2. Flowchart of the TPE optimization procedure.

5.4 Dropout regularization

A well-known and critical problem of deep neural networks such as LSTM is overfitting [52]. That is, when the training data is limited, complicated mappings between the inputs and outputs that are learned by the network might be the result of sampling noise, which only exist in the training set but not in the real test set. One way to regularize such a network is averaging the outputs of all possible configurations of the parameters, in which each configuration is weighted by its posterior probability given by the training data [53]. This method can be applied only for simple or small networks. With

large neural networks, the computation for training many different network architectures or training one architecture on different data sets is very expensive. Dropout is a technique that addresses this issue [53].

A motivation for dropout comes from a theory of sexual reproduction [54], in which new genes are naturally selected to spread throughout the population based on their competitiveness and less co-adaptation which may reduce the chance of a new gene improving the fitness of an individual. Likewise, dropout aims to train each hidden unit in a neural network with a randomly chosen sample of other units. By dropping a unit out, we temporarily remove it from the network along with all its connections during the training process as illustrated in Fig. 5.3, in order to prevent units from high co-adaptation. By so doing, each hidden unit becomes more robust and is able to create useful features on its own without relying on other units, which helps the network avoid overfitting.



(a) (b)

Fig. 5.3. An example of a dropout network model [53]: (a) A fully connected 2-hidden layers network; (b) The network obtained by applying dropout. Crossed units are excluded.

Consider a neural network with L hidden layers, in which the input and output vectors of layer l (for $l \in \{1, \ldots, L\}$) are denoted as $z^{(l)}$ and $y^{(l)}$, respectively. $w^{(l)}$ and $b^{(l)}$ are the weights and biases of layer l, respectively. For a standard neural network, the feed-forward operation can be described as:

$$z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)}, \tag{5.15}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}), \tag{5.16}$$

where f is the activation function and i denotes the index of hidden unit, as illustrated in Fig. 5.4(a).

With a dropout network (Fig. 5.4(b)), a vector of independent Bernoulli random variables $r^{(l)}$ with probability p is used at each hidden layer l to generate the thinned outputs $\tilde{y}^{(l)}$ as follows:

$$r_j^{(l)} \sim Bernoulli(p), \tag{5.17}$$

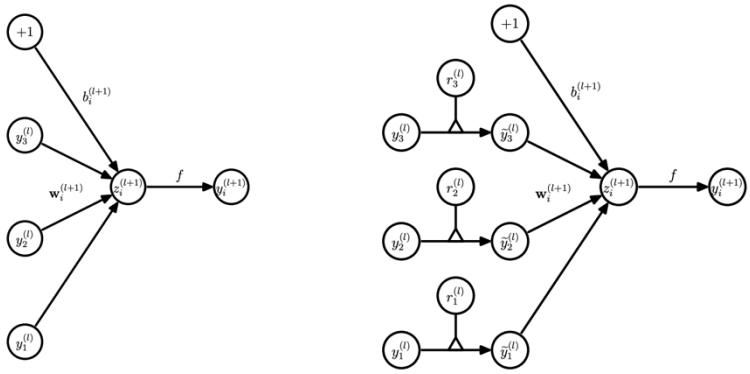
$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)}, \tag{5.18}$$

where * denotes an element-wise product. The thinned outputs are, then, used as inputs to the next layer

of the feed-forward operation:

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}, \tag{5.19}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}), \tag{5.20}$$



(a) Standard network (b) Dropout network

Fig. 5.4. Comparison of the basic operations of a standard and dropout network [53].

References

[1] 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.

[2] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering and System Safety*. 2009.

[3] M. G. Na *et al.*, "Prediction of major transient scenarios for severe accidents of nuclear power plants," *IEEE Trans. Nucl. Sci.*, 2004.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] M. Qi and G. P. Zhang, "Trend time-series modeling and forecasting with neural networks," *IEEE Trans. Neural Networks*, 2008.

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

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

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

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

[14] J. Liu and E. Zio, "A SVR-based ensemble approach for drifting data streams with recurring patterns," *Appl. Soft Comput. J.*, 2016.

[15] J. Liu, V. Vitelli, E. Zio, and R. Seraoui, "A Novel Dynamic-Weighted Probabilistic Support Vector Regression-Based Ensemble for Prognostics of Time Series Data," *IEEE Trans. Reliab.*, 2015.

[16] Y. Bao, T. Xiong, and Z. Hu, "Multi-step-ahead time series prediction using multiple-output support vector regression," *Neurocomputing*, 2014.

[17] D. Wang, H. Luo, O. Grunder, and Y. Lin, "Multi-step ahead wind speed forecasting using an improved wavelet neural network combining variational mode decomposition and phase space reconstruction," *Renew. Energy*, 2017.

[18] 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.

[19] B. Doucoure, K. Agbossou, and A. Cardenas, "Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data," *Renew. Energy*, 2016.

[20] 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.

[21] V. T. Tran, B. S. Yang, and A. C. C. Tan, "Multi-step ahead direct prediction for the machine condition prognosis using regression trees and neuro-fuzzy systems," *Expert Syst. Appl.*, 2009.

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

[23] 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.

[24] 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.

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

[26] H. Liu, X. wei Mi, and Y. fei Li, "Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network," *Energy Convers. Manag.*, 2018.

[27] J. Wang and Y. Li, "Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy," *Appl. Energy*, 2018.

[28] Y. Li, H. Wu, and H. Liu, "Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEWT reconstruction," *Energy Convers. Manag.*, 2018.

[29] Y. L. Hu and L. Chen, "A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm," *Energy Convers. Manag.*, 2018.

[30] I. A. Araya, C. Valle, and H. Allende, "A Multi-Scale Model based on the Long Short-Term Memory for

day ahead hourly wind speed forecasting," *Pattern Recognit. Lett.*, 2019.

[31] X. Yuan, C. Chen, M. Jiang, and Y. Yuan, "Prediction interval of wind power using parameter optimized Beta distribution based LSTM model," *Appl. Soft Comput. J.*, 2019.

[32] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, 2018.

[33] S. Srivastava and S. Lessmann, "A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data," *Sol. Energy*, 2018.

[34] K. Wang, X. Qi, and H. Liu, "Photovoltaic power forecasting based LSTM-Convolutional Network," *Energy*, 2019.

[35] Q. Wu and H. Lin, "Daily urban air quality index forecasting based on variational mode decomposition, sample entropy and LSTM neural network," *Sustain. Cities Soc.*, 2019.

[36] Y. Zhou, F. J. Chang, L. C. Chang, I. F. Kao, and Y. S. Wang, "Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts," *J. Clean. Prod.*, 2019.

[37] 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.

[38] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A Stat. Mech. its Appl.*, 2019.

[39] 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.

[40] H. Su, E. Zio, J. Zhang, M. Xu, X. Li, and Z. Zhang, "A hybrid hourly natural gas demand forecasting method based on the integration of wavelet transform and enhanced Deep-RNN model," *Energy*, 2019.

[41] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Appl. Energy*, 2019.

[42] O. Laib, M. T. Khadir, and L. Mihaylova, "Toward efficient energy systems based on natural gas consumption prediction with LSTM Recurrent Neural Networks," *Energy*, 2019.

[43] L. Yu, S. Liang, R. Chen, and K. K. Lai, "Predicting monthly biofuel production using a hybrid ensemble forecasting methodology," *Int. J. Forecast.*, 2019.

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

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

[46] 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.

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

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

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

[50] 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.

[51] 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.

[52] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," *arXiv e-prints*, p. arXiv:1409.2329, Sep. 2014.

[53] I. Sutskever, G. Hinton, A. Krizhevsky, and R. R. Salakhutdinov, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, 2014.

[54] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman, "Sex, mixability, and modularity," *Proc. Natl. Acad. Sci. U. S. A.*, 2010.

Chapter 6 A Multi-Step Ahead Prediction Method For NPP Steam Generator Signals

Developing an accurate and reliable multi-step ahead prediction model is a key problem in many Prognostics and Health Management (PHM) applications. Inevitably, the further one attempts to predict into the future, the harder it is to achieve an accurate and stable prediction due to increasing uncertainty and error accumulation. In this Chapter, we address this problem by proposing a predictive model based on Long Short-Term Memory (LSTM), a deep neural network developed for dealing with the long-term dependencies in time-series data. The proposed prediction model tackles two additional issues. Firstly, the hyperparameters of the proposed model are automatically tuned by a Bayesian optimization algorithm, called Tree-structured Parzen Estimator (TPE). Secondly, the proposed model allows assessing the uncertainty of the prediction. To validate the performance of the proposed model, a case study considering steam generator data acquired from different French Nuclear Power Plants (NPPs) is carried out. Alternative prediction models are also considered for comparison purposes.

6.1 Proposed LSTM-based prognostic framework

In this Section, we present a prognostic framework for the multi-step ahead prediction of the time-series data from steam generators (SGs), as illustrated in Fig. 6.1.

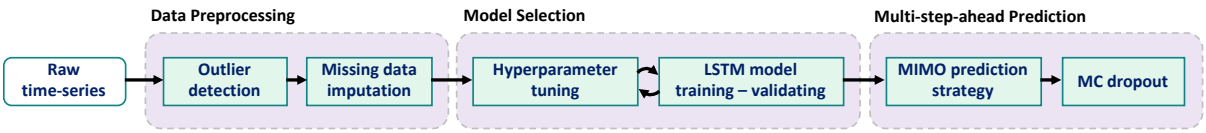


Fig. 6.1. The flowchart of the proposed multi-step ahead prediction framework for SGs.

The proposed framework consists of three main stages: data preprocessing, model selection and multi-step ahead prediction. Firstly, the data preprocessing stage is responsible for preparing the data for training and testing the prediction model. Then, in the second stage, a LSTM-based model is built for the Multi-Input Multi-Output (MIMO) prediction using the training data and its hyperparameters are automatically optimized with the objective function of minimizing the validation error. In the last stage, the performance of the trained prediction model is validated for multi-step ahead prediction and a Monte Carlo (MC) dropout technique is used to capture the prediction uncertainty. The procedure of the proposed framework can be summarized as in Algorithm 6.1, where *max_iter* is the number of

optimization iterations and N_{MC} is the number of MC dropout realizations. The details of each stage are given in the following sections.

Algorithm 6.1. Procedure of the proposed multi-step prediction framework

Input: A raw time series data collected up to time t: $\{x_1, x_2, ..., x_t\}$

Output: Predictions of H next observations and their uncertainty information

Preprocessing stage

1. Detect and remove outliers
2. Impute missing data points

Model selection stage

3. **for** i in {1,…,*max_iter*} **do**
    a. Select the optimal network hyperparameters at the ith trial with TPE
    b. Validate the hyperparameters by using k-fold cross-validation
    c. Update the fitness value with the average training error measured over k folds
4. Select the best hyperparameter setting with the lowest fitness value

Multi-step ahead prediction stage

5. **for** i in {1,…,N_{MC}} **do**
    a. Build a LSTM-based prediction model with the selected hyperparameters
    b. Perform the predictions for H steps ahead $\{\hat{x}_{t+h}\}, h \in [1, H]$ by using the MIMO prediction strategy
6. Calculate the mean and confidence interval of the predictions over N_{MC} realizations

6.1.1 Data preprocessing

The quality of the observation data for training is one of the most important factors for the successful performance of a prediction model. Due to the errors during sensor measurements or signal transmission, the acquired observations may include missing and anomalous data points, e.g. outliers, which can negatively impact the model performance. In this study, we adopt a raw data preprocessing module focusing on the two following tasks: 1) detecting and removing outliers; 2) imputing missing data points, the number of which may increase after removing outliers.

The first problem is addressed by using the Isolation Forest, an outlier detection technique built on the basis of decision trees [1]. This technique is based on an assumption that outliers are few, different and susceptible to a mechanism called isolation. In comparison with conventional distance and density measures, isolation has been proved to be a much more effective indicator to detect anomalies. In addition, Isolation Forest also requires a small linear time complexity. Further details on the algorithm of Isolation Forest can be found in [1]. Once outliers are reduced, a local polynomial regression

technique is used to reconstruct missing data samples and reduce noises. The preprocessed data is later used for training and testing the prediction model in the following stages.

6.1.2 Model selection

A. Prediction horizon

Several research works have been carried out on determining an optimal horizon of prediction in order to provide predictions accurately and timely, and to ensure the usefulness of the prognostic model. However, to the authors' knowledge, there is no general rule reported for dealing with this issue. We have carried out a review on the horizons selected in recent prediction studies for industrial applications during 2015-2019 [2]–[28] and the result is summarized in Fig. 6.2. The result shows that multi-step ahead prediction has been less studied than single-step ahead prediction, and that most of the works were carried out with horizons ranging from 3 to 6 steps ahead. To demonstrate the effectiveness of the proposed model, a prediction horizon of 15 steps (45 days of operation) ahead is investigated in this study.

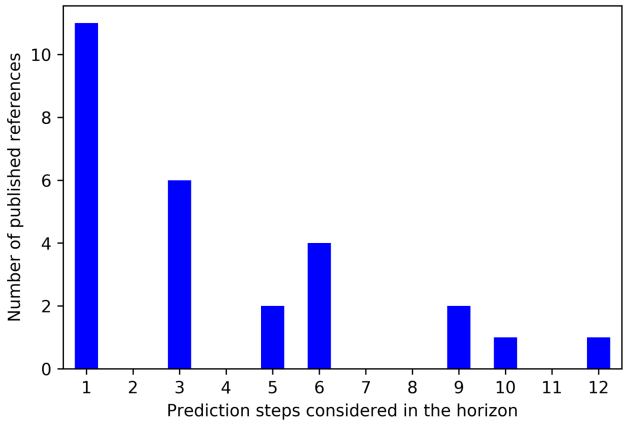


Fig. 6.2. Prediction horizons of recent studies.

B. Hyperparameter optimization

The TPE optimization approach described in Chapter 5, Section 5.3, is employed to automatically optimize the hyperparameters of the proposed prediction model.

6.1.3 Multi-step ahead prediction

In the testing stage, the MIMO prediction strategy introduced in Chapter 5, Section 5.1.2.C, is used to predict the future values. As mentioned in Section 6.1.2, the prediction horizon h is set to 15-step ahead in this study, as shown in Fig. 6.3.

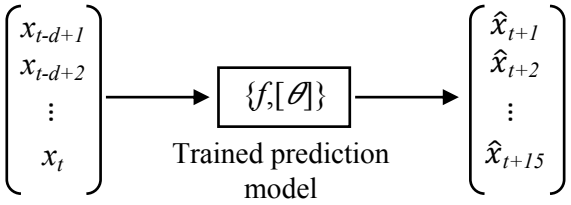


Fig. 6.3. Multi-step ahead prediction procedure.

To further assess the prediction performance, we adopt a MC Dropout technique [29] in order to capture the uncertainty information of the multi-step ahead predictions of the proposed model. It is important to note that the standard LSTM network is not capable to quantify the prediction uncertainty itself. In the MC dropout technique, a dropout probability is applied to all the weight layers in the network, which represents the network weights drawn from a Bernoulli distribution. Thus, the prediction uncertainty can be quantified by running several forward passes through the network. In this study, we perform N_{MC}=100 stochastic forward passes, in which network units of each layer are randomly dropped out, and obtain the mean and confidence interval of the predictions.

6.2 Experimental study

6.2.1 SG data

In this work, the prediction performance of the proposed model is evaluated on the SG data of French NPPs. SGs in pressurized water reactors (PWRs) are heat exchangers which use the heat from the primary reactor coolant to produce steam in the secondary side and, thus, drive the turbine generators. In addition, the SGs act as a safety barrier between the radioactive primary side and the non-radioactive secondary side. Due to their critical role in NPPs, any degradation mechanism in SGs should be monitored and prevented at the early stages of propagation. A widely used method of degradation monitoring is the analysis of the Wide Range Level (WRL) dynamic behavior recorded by control sensors [30], [31].

WRL is one of the condition monitoring variables measured from the NPP SGs. It is estimated from the difference between the pressure measured at two difference heights, i.e. the dome and the bottom of the downcomer, as illustrated in Fig. 6.4 (label 18) [31]. Due to its nature, WRL is very sensitive to the temperature, the flow rate of the feed-water and the circulation ratio of the SG. Usually, WRL variations are monitored during slow transients and during manual control at low power load [31]. Among critical SG degradation mechanisms, clogging is a phenomenon where the flow holes of the tube support plates are partially or completely blocked by deposits, leading to the reduction of the circulation flow rate in the SGs [31]. Clogging in SG is a slow process which may take several years. In [32], it has been shown that the WRL of a SG is closely related to the clogging degradation. Thus,

the predictions of WRL can be converted to the clogging degradation state.

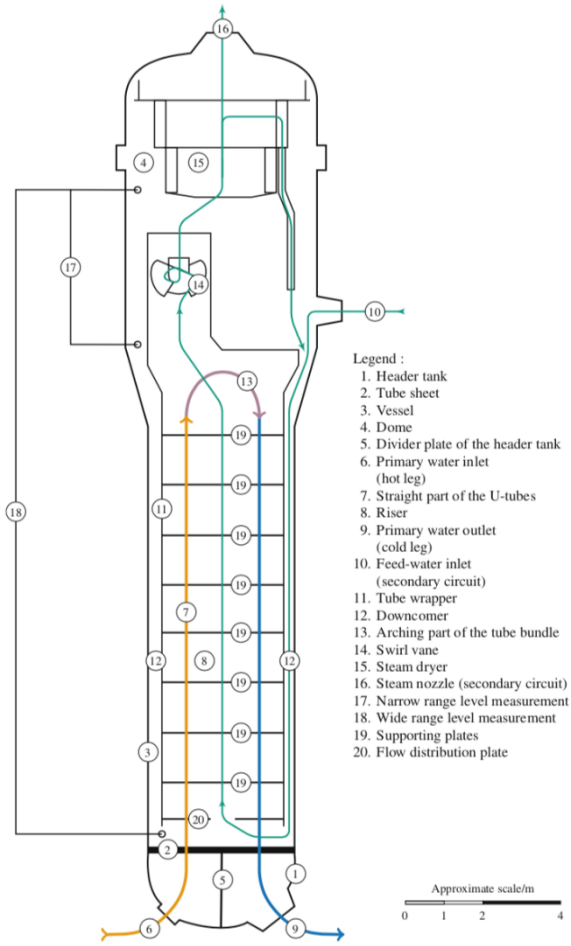
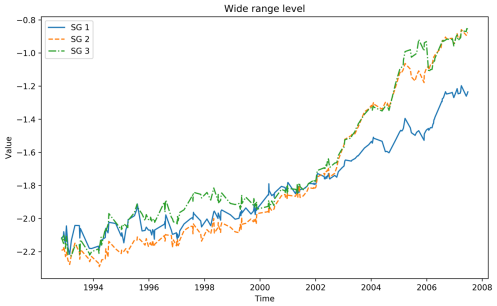


Fig. 6.4. The front-cut schematic of a 51B-model SG [31].

The original SG data employed in this study were collected from six SGs of two different 900-MW NPPs, which are operated by Électricité de France (EDF). Each plant consists of three SGs. The WRL data were recorded during the stationary regimes in which the power demand percentage is stably maintained greater than 90%, at an interval of 3 days from July 1992 to June 2007. Fig. 6.5 shows the temporal evolution of the WRL observations of the two NPPs. The names of the plants are omitted for confidentiality reasons.



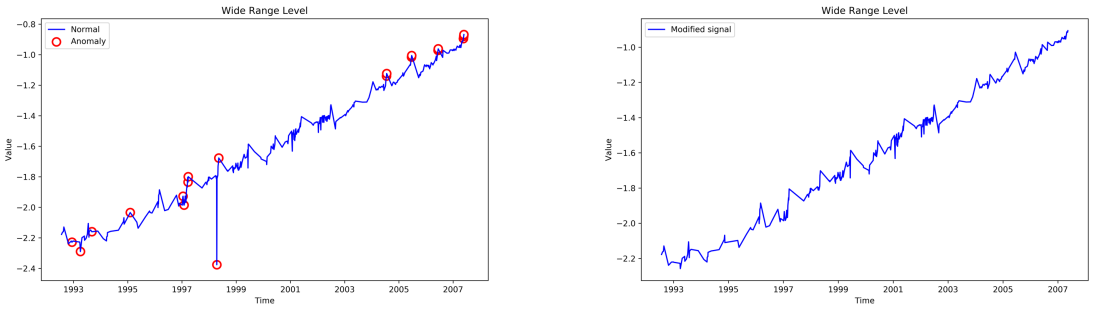
(a) Plant No. 1

(b) Plant No. 2

Fig. 6.5. Raw WRL measurements recorded from control sensors of different NPPs.

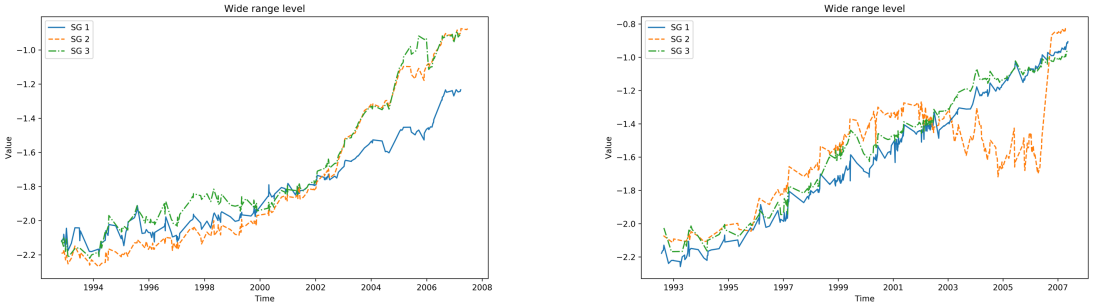
6.2.2 Data preprocessing

Before being used for the model development, the raw SG data are preprocessed by using the Isolation Forest and local regression approaches described in Section 6.1.1. Fig. 6.6 shows the results of applying the Isolation Forest for reducing outliers in the data of SG 1 of plant No. 2. In Fig. 6.6(a), the solid line indicates the normal measurements whereas the detected outliers are highlighted as circled points, which are later eliminated in Fig. 6.6(b). An interesting observation in Fig. 6.6(a) is the anomalous spike between 1997 and 1999. Without the outlier detection step, this sudden spike could highly impact, in a negative manner, on the prediction accuracy. After reducing the outliers, imputations for missing data samples are given. The preprocessed data of all SGs after the preprocessing stage are shown in Fig. 6.7.



(a) Original signal with outliers highlighted (b) Modified signal after outlier removal

Fig. 6.6. Applying the Isolation Forest to the data of SG 1 of plant No. 2.



(a) Plant No. 1 (b) Plant No. 2

Fig. 6.7. The results of the preprocessing stage for all SG data.

6.3 Results and discussion

After the preprocessing stage, each SG data series is divided into a training set and a testing set. The data for the first 11 years, from July 1992 to December 2002, which include a total of 1230 samples

at a 3-day interval, are selected to train the proposed prediction model and the next 5-year data with 510 samples are employed to test the model performance.

Before constructing the proposed model, we employ the False Nearest Neighbor (FNN) algorithm [33] to determine the appropriate embedding dimension *d* of the data series. The main idea of the FNN algorithm is to find the minimum dimension where the distances between the nearest neighbors in the time series do not significantly change in the next higher dimensional embedding. Fig. 6.8 shows the result of applying FNN to the data of SG 1 of plant No. 1. A threshold for identifying the minimum embedding dimension is set to 0. In this Figure, the minimum embedding dimension value is found at 12. We summarize the optimal embedding dimensions identified for all the SGs data series in Table 6.1.

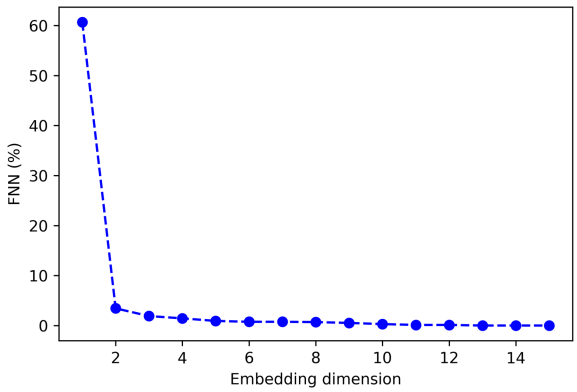


Fig. 6.8. FNN result for SG 1 of plant No. 1.

Table 6.1 Minimum embedding dimensions for all SGs.

Plant	No. 1			No. 2		
SG	1	2	3	1	2	3
Embedding dimension	12	13	9	9	11	6

In this study, we carry out three comparisons to evaluate the performance of the proposed prognostic model. The first comparison is conducted to analyze the viability of TPE in tuning the proposed model during the training stage. As a standard optimization approach, Random Search (RS) is considered for benchmarking purposes. Another comparison is, then, carried out to specifically validate the efficacy of dropout in the proposed prediction framework. In the third comparison, four hybrid prediction models, including single-output support vector regression using recursive strategy (SVR-REC), multi-output support vector regression using MIMO strategy (SVR-MIMO), single-output multilayer perceptron neural network using recursive strategy (MLP-REC) and multi-output multilayer perceptron neural network using MIMO strategy (MLP-MIMO), are employed as the benchmark

models for comparison with the proposed model in multi-step ahead predictions. In this performance evaluation, three prediction accuracy metrics are considered, including Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Scaled Error (MASE). Their definitions are given as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{x}_i - x_i\right)^2}\,, \tag{6.1}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{x}_i - x_i}{x_i}\right| \times 100\%, \tag{6.2}$$

$$MASE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left|\hat{x}_i - x_i\right|}{\frac{1}{N-1}\sum_{j=2}^{N}\left|x_j - x_i\right|}\right), \tag{6.3}$$

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

6.3.1 Automatic hyperparameter optimization

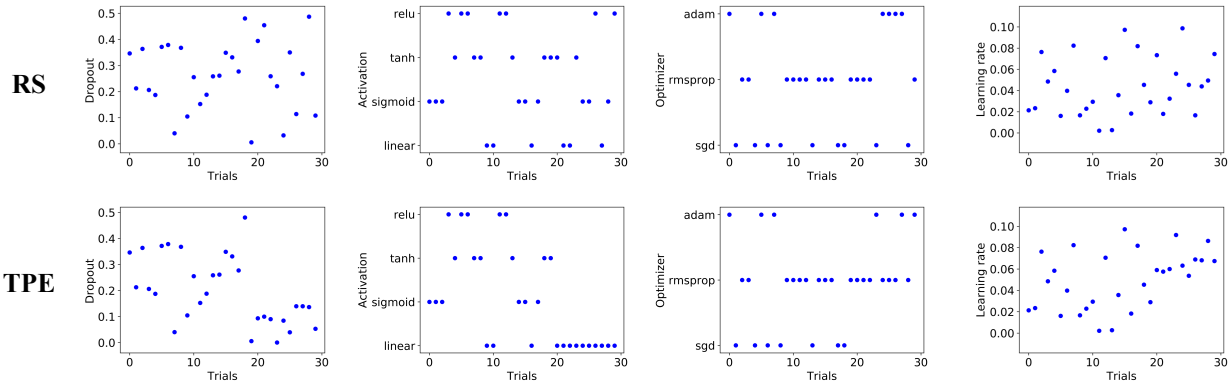
The proposed prediction model is constructed with one LSTM layer with 64 neurons. Four major hyperparameters of the model are to be tuned, including dropout rate, activation function type, optimizer type and learning rate. The details of the hyperparameter search space are shown in Table 6.2. For a fair comparison, the TPE and RS algorithms are evaluated by using the same model configurations and hyperparameter search space. The number of optimization trials is selected as 30 for the two algorithms. In addition, a *k*-fold cross-validation (k = 3 in this study) is adopted to prevent overfitting during training the model. The Mean Square Error (MSE) is used as the objective function for model selection. In other words, at each optimization trial, the hyperparameter configuration with the lowest average prediction error evaluated by cross-validation is chosen. To achieve the training convergence, the number of training epochs is set to 100.

Table 6.2 Hyperparameters of the proposed prediction model.

Hyperparameter	Type of distribution	Value set or Range
Dropout rate	Uniform float	[0, 0.5]
Activation function	Categorical	{Linear, Sigmoid, Tanh, ReLU}
Optimizer	Categorical	{SGD, RMSprop, Adam}
Learning rate	Uniform float	[0.0001, 0.1]

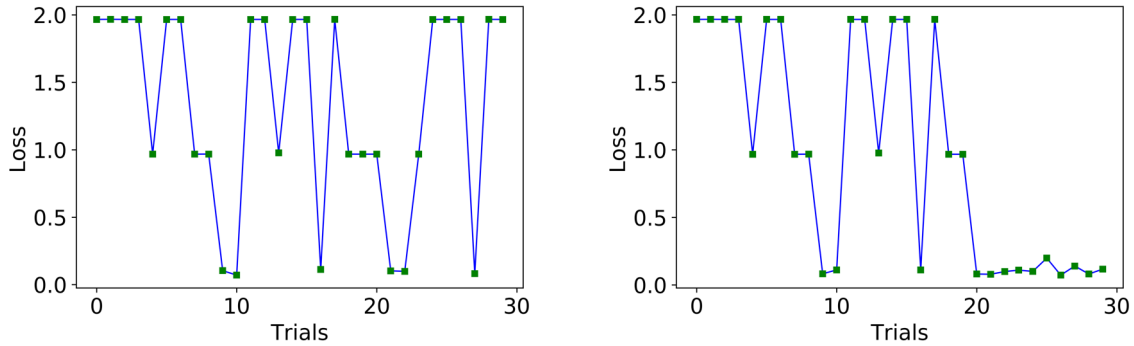
Fig. 6.9 shows the comparison of the TPE and RS hyperparameter searches over 30 trials for SG 1

of plant No. 1. The corresponding training loss is also given in Fig. 6.10. In particular, the TPE algorithm uses the first 20 startup trials for initializing the distributions of the *good* and *bad* hyperparameter sets, as mentioned in Chapter 5, Section 5.3. This initialization process is performed by employing a standard RS. Therefore, in Figs. 6.9 and 6.10, we can observe a similar performance between TPE and RS in both hyperparameter searching and their obtained training losses during the first 20 trials. However, the performance of TPE is quickly improved after the initialization. It much more focuses on the good hyperparameter configurations which was found in the previous trials, leading to faster converge and lower training loss than RS within 30 trials.



(a) Dropout rate (b) Activation function (c) Optimizer (d) Learning rate

Fig. 6.9. Hyperparameters tuning process over 30 trials by TPE (top Figures) and RS (bottom Figures) for SG 1 of plant No.1.



(a) RS (b) TPE

Fig. 6.10. Training loss versus trials of TPE and RS for SG 1 of plant No. 1.

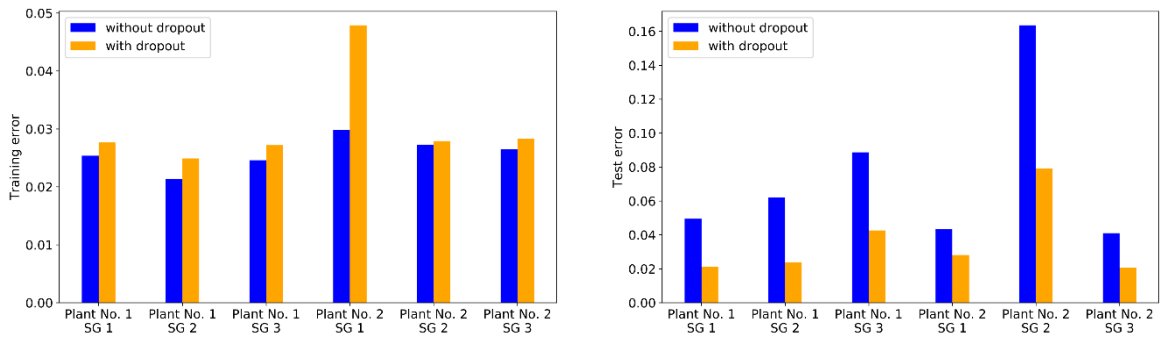
In Table 6.3, we show the performance comparison between TPE and RS, in terms of their obtained best training loss for all SGs. The results obviously show that the optimal configurations found by TPE generally outperform the best ones found by RS in the considered case studies. Thus, the optimal hyperparameter configurations found by TPE are used for prediction in the next stage.

Table 6.3 The best training loss obtained by TPE and RS in hyperparameter tuning for all SGs.

Plant	No. 1			No. 2		
SG	1	2	3	1	2	3
Random search	0.0487	0.0479	0.0307	0.0358	0.0321	0.0343
TPE	0.0440	0.0370	0.0319	0.0350	0.0314	0.0270

6.3.2 Dropout regularization

In this section, a comparison is carried out between the proposed prediction model and a model with the same architecture but trained without dropout. The other hyperparameters are kept identical between the two models, as described in Section 6.3.1. The probability of the used dropout is automatically optimized by TPE. We employ all the six SG datasets to comprehensively evaluate dropout during both the training and test phases in terms of RMSE. The comparative results are shown in Fig. 6.11. The result shows that the prediction model trained without dropout has lower training errors but much higher test errors, which may be an indication of the presence of overfitting. In contrast, the dropout model significantly reduces the overfitting problem with lower test errors for all the datasets. The average error reduction of the dropout model is 51.91%, which strongly indicates the efficacy of dropout in reducing overfitting and improving the prediction performance of the neural network.



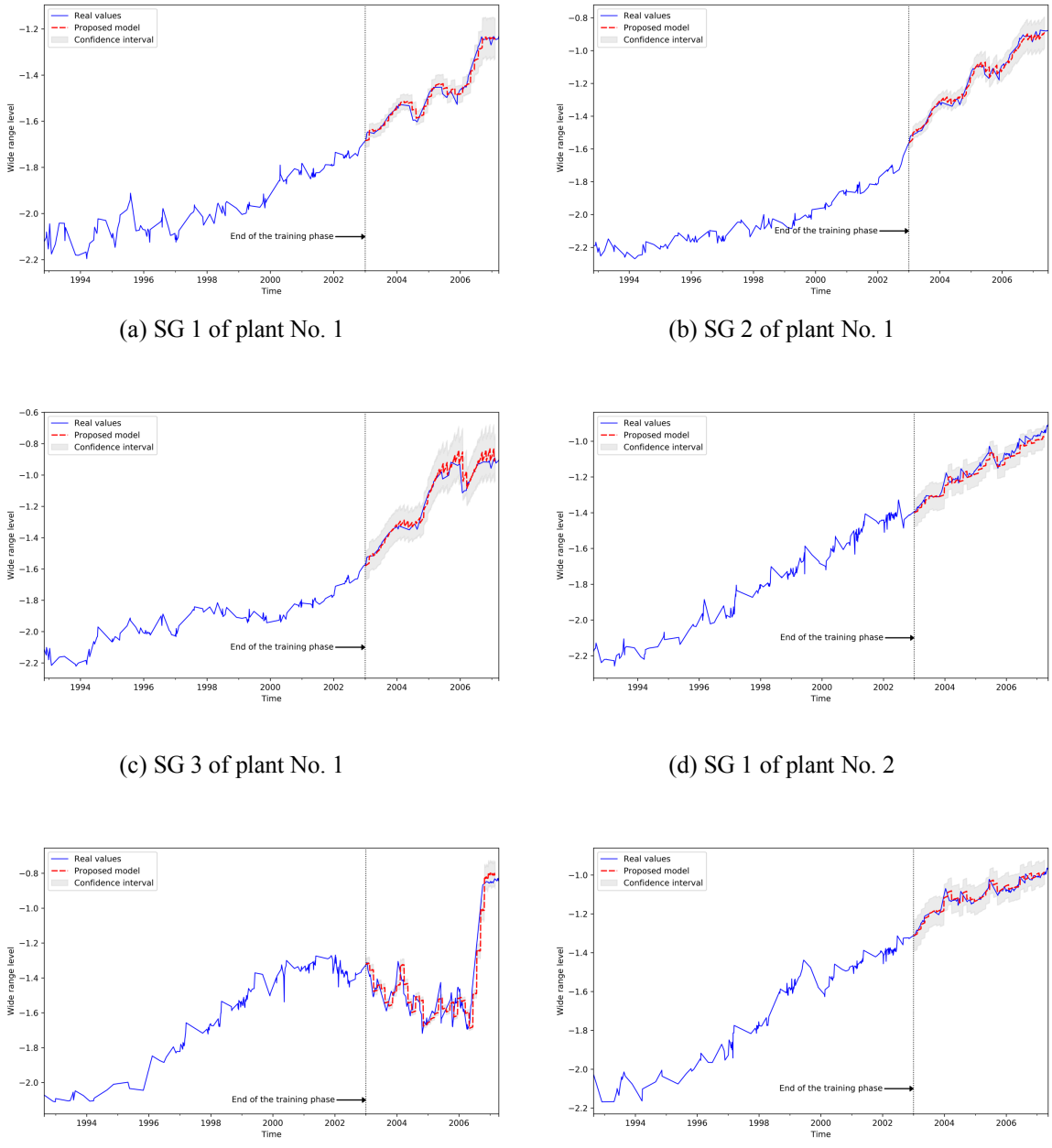
(a) Training phase (b) Test phase

Fig. 6.11. Training and test errors for the network architecture trained without and with dropout.

6.3.3 Performance evaluation

The WRL measurements of the six SGs are used for validating the developed prediction model for multi-step ahead prediction. It is important to remind that the prediction horizon used in this study is 15 steps ahead, which equals 45 operating days of the SGs. After the training is finished, the prediction model is used to continuously predict 15-step ahead in the next 5 years. Fig. 6.12 illustrates the prediction results of the proposed model for all SGs. The predicted values are shown as the dashed line, whereas the solid line depicts the actual observations. The 95% confidence interval of the predictions,

obtained via MC simulations, is depicted as the grey region. The results show that the proposed model is able to keep track with the changes of the WRL data while achieving accurate predictions, which are very close to the actual data for all SGs. Moreover, the 95% confidence bounds of the predictions are narrow and close to the target values, indicating predictions with a high precision. In industrial applications, these results are of crucial importance for accurately estimating the equipment RUL.



(a) SG 1 of plant No. 1

(b) SG 2 of plant No. 1

(c) SG 3 of plant No. 1

(d) SG 1 of plant No. 2

(e) SG 2 of plant No. 2

(f) SG 3 of plant No. 2

Fig. 6.12. Multi-step ahead prediction results by the proposed model for all SGs.

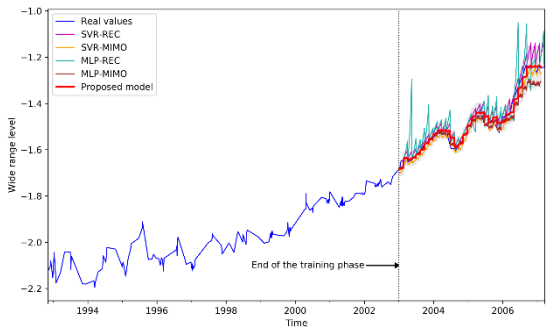
The prediction results obtained by the proposed model are, then, evaluated with respect to the four

benchmark models, i.e. SVR-REC, SVR-MIMO, MLP-REC and MLP-MIMO, in terms of prediction accuracy. For a fair comparison, the hyperparameters of the compared models are optimized by using TPE with 30 trials. The details of the hyperparameter search spaces of the compared models are shown in Table 6.4.

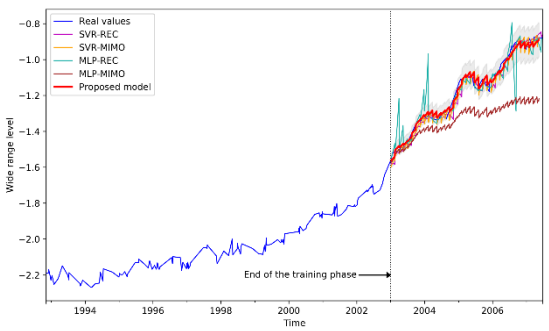
Table 6.4 Hyperparameters of the benchmark models.

Model	Hyperparameter	Value set or Range
SVR (including SVR-REC and SVR-MIMO)	Kernel function	{Linear, RBF, Poly, Sigmoid}
	Degree (of the polynomial kernel function)	[2, 4]
	Regularization parameter (C)	[0.01, 100]
	Kernel coefficient (gamma)	[0.01, 10]
MLP (including MLP-REC and MLP-MIMO)	Hidden layer size	[1, 5]
	Activation function	{Logistic, Tanh, ReLU}
	Optimizer	{LBFGS, SGD, Adam}
	Learning rate	{Constant, Invscaling, Adaptive}
	Regularization parameter (alpha)	[0.0001, 0.01]

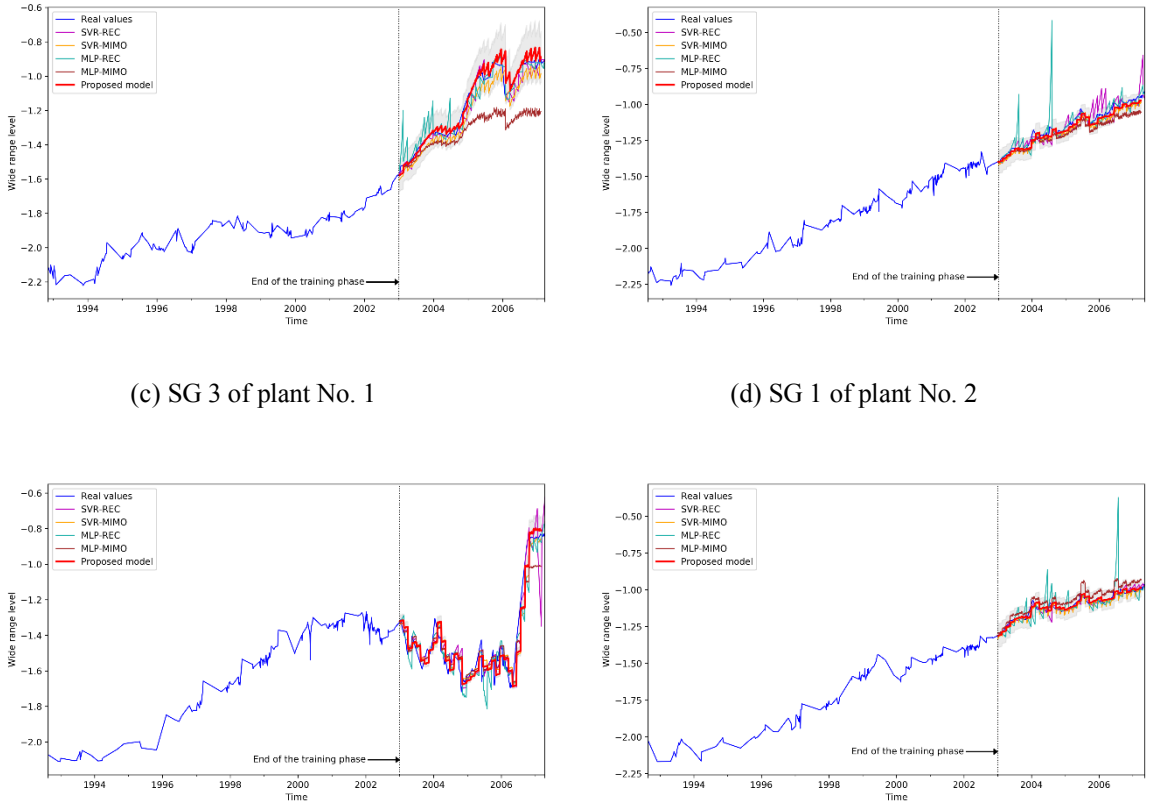
The comparative results of the proposed model and the four benchmark models for multi-step ahead predictions are shown in Fig. 6.13. Table 6.5 summarizes the prediction results in terms of the three accuracy indicators for different SG data. As can be seen in Fig. 6.13 and Table 6.5 (values in bold), the proposed prediction model outperforms the four other benchmark models and achieves higher accuracy for all SGs. The results indicate the accurate and efficient learning of the proposed prediction model for the long-term dependencies of the SG data.



(a) SG 1 of plant No. 1



(b) SG 2 of plant No. 1



(c) SG 3 of plant No. 1

(d) SG 1 of plant No. 2

(e) SG 2 of plant No. 2

(f) SG 3 of plant No. 2

Fig. 6.13. Multi-step ahead predictions using different models for all SGs.

Table 6.5 Comparison of the prediction performance in multi-step ahead predictions for all SGs.

	Method	SG 1			SG 2			SG 3		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
Plant No. 1	SVR-REC	0.0382	2.0775	15.2484	0.0333	2.1970	10.2221	0.0508	3.3173	12.1521
	SVR-MIMO	0.0283	1.6511	12.2085	0.0331	2.3878	10.8281	0.0640	5.1793	19.2863
	MLP-REC	0.0597	2.7824	20.8682	0.0656	3.3349	15.3398	0.0577	3.3597	13.4886
	MLP-MIMO	0.0339	1.7074	11.8887	0.1888	15.1554	62.2885	0.1867	15.2453	52.7662
	Proposed model	**0.0212**	**1.0950**	**8.6166**	**0.0239**	**1.6973**	**5.8214**	**0.0426**	**2.7230**	**4.0846**
Plant No. 2	SVR-REC	0.0572	3.6462	12.8555	0.0906	4.7909	**7.0354**	0.0242	1.5005	**6.1132**
	SVR-MIMO	0.0401	3.1774	11.8732	0.0849	4.7575	7.4570	0.0247	1.7842	7.5984
	MLP-REC	0.0751	3.0309	11.2367	0.0862	4.7819	7.4403	0.0734	3.4535	14.2511
	MLP-MIMO	0.0607	4.8530	17.4741	0.0888	5.4251	7.5992	0.0499	4.1168	17.2689
	Proposed model	**0.0281**	**2.0117**	**8.6455**	**0.0791**	**4.4033**	9.3923	**0.0206**	**1.3992**	7.9604

The average computational time of training the proposed prediction model is 3.2 hours, on a GPGPU node comprising 2 Intel Xeon CPU E5-2695 (24 cores at 2.40 Hz with 32 GB of RAM) and 2

Nvidia Tesla K40m graphic cards (with 12 GB of GRAM). It is important to note that SG data used in this work were recorded at an interval of 3 days. After being trained, the proposed model can be used to perform a 15-step ahead prediction, which is equivalent to 45 operating days ahead of the SGs. Due to this reason, the proposed prediction framework can be applied for a real-time time series prediction of the considered application.

References

[1] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008.

[2] Q. Han, F. Meng, T. Hu, and F. Chu, "Non-parametric hybrid models for wind speed forecasting," *Energy Convers. Manag.*, 2017.

[3] Z. Qu, K. Zhang, W. Mao, J. Wang, C. Liu, and W. Zhang, "Research and application of ensemble forecasting based on a novel multi-objective optimization algorithm for wind-speed forecasting," *Energy Convers. Manag.*, 2017.

[4] S. R. Moreno and L. dos Santos Coelho, "Wind speed forecasting approach based on Singular Spectrum Analysis and Adaptive Neuro Fuzzy Inference System," *Renew. Energy*, 2018.

[5] X. Mi, H. Liu, and Y. Li, "Wind speed prediction model using singular spectrum analysis, empirical mode decomposition and convolutional support vector machine," *Energy Convers. Manag.*, 2019.

[6] T. Niu, J. Wang, K. Zhang, and P. Du, "Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy," *Renew. Energy*, 2018.

[7] J. Chen, G. Q. Zeng, W. Zhou, W. Du, and K. Di Lu, "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization," *Energy Convers. Manag.*, 2018.

[8] C. Tian, Y. Hao, and J. Hu, "A novel wind speed forecasting system based on hybrid data preprocessing and multi-objective optimization," *Appl. Energy*, 2018.

[9] Z. Yang and J. Wang, "A hybrid forecasting approach applied in wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm," *Energy*, vol. 160, pp. 87–100, Oct. 2018.

[10] M. Santhosh, C. Venkaiah, and D. M. Vinod Kumar, "Ensemble empirical mode decomposition based adaptive wavelet neural network method for wind speed prediction," *Energy Convers. Manag.*, 2018.

[11] J. Naik, P. Satapathy, and P. K. Dash, "Short-term wind speed and wind power prediction using hybrid empirical mode decomposition and kernel ridge regression," *Appl. Soft Comput. J.*, 2018.

[12] Z. Yang and J. Wang, "A combination forecasting approach applied in multistep wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm," *Appl. Energy*, vol. 160, pp. 87–100, Oct. 2018.

[13] H. Liu, Z. Duan, F. ze Han, and Y. fei Li, "Big multi-step wind speed forecasting model based on secondary decomposition, ensemble method and error correction algorithm," *Energy Convers. Manag.*, 2018.

[14] J. Hu, J. Wang, and L. Xiao, "A hybrid approach based on the Gaussian process with t-observation model for short-term wind speed forecasts," *Renew. Energy*, 2017.

[15] M. A. Chitsazan, M. Sami Fadali, and A. M. Trzynadlowski, "Wind speed and wind direction forecasting using echo state network with nonlinear functions," *Renew. Energy*, 2019.

[16] D. Y. Hong, T. Y. Ji, M. S. Li, and Q. H. Wu, "Ultra-short-term forecast of wind speed and wind power based on morphological high frequency filter and double similarity search algorithm," *Int. J. Electr. Power Energy Syst.*, 2019.

[17] H. Liu, X. Mi, and Y. Li, "Smart multi-step deep learning model for wind speed forecasting based on

variational mode decomposition, singular spectrum analysis, LSTM network and ELM," *Energy Convers. Manag.*, 2018.

[18] 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.

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

[20] J. Liu and E. Zio, "A SVR-based ensemble approach for drifting data streams with recurring patterns," *Appl. Soft Comput. J.*, 2016.

[21] J. Liu, V. Vitelli, E. Zio, and R. Seraoui, "A Novel Dynamic-Weighted Probabilistic Support Vector Regression-Based Ensemble for Prognostics of Time Series Data," *IEEE Trans. Reliab.*, 2015.

[22] L. Xiao, F. Qian, and W. Shao, "Multi-step wind speed forecasting based on a hybrid forecasting architecture and an improved bat algorithm," *Energy Convers. Manag.*, 2017.

[23] P. Du, J. Wang, Z. Guo, and W. Yang, "Research and application of a novel hybrid forecasting system based on multi-objective optimization for wind speed forecasting," *Energy Convers. Manag.*, 2017.

[24] J. Wang, P. Du, T. Niu, and W. Yang, "A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting," *Appl. Energy*, 2017.

[25] X. wei Mi, H. Liu, and Y. fei Li, "Wind speed forecasting method using wavelet, extreme learning machine and outlier correction algorithm," *Energy Convers. Manag.*, 2017.

[26] D. Wang, H. Luo, O. Grunder, and Y. Lin, "Multi-step ahead wind speed forecasting using an improved wavelet neural network combining variational mode decomposition and phase space reconstruction," *Renew. Energy*, 2017.

[27] W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, and X. Fan, "A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting," *Energy Convers. Manag.*, 2017.

[28] Y. Li, H. Wu, and H. Liu, "Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEWT reconstruction," *Energy Convers. Manag.*, 2018.

[29] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *arXiv e-prints*, p. arXiv:1506.02142, Jun. 2015.

[30] G. Corredera, M. Alves-Vieira, and O. de Bouvier, "Fouling and TSP Blockage of Steam Generators on EDF Fleet : Identified Correlations with Secondary Water Chemistry and planned Remedies," in *International Conference, Water chemistry of nuclear reactor systems; 2008; Berlin, Germany*, 2008.

[31] S. Girard, *Physical and Statistical Models for Steam Generator Clogging Diagnosis*. Cham: Springer International Publishing, 2014.

[32] P. Luca, "Development of Unsupervised and Semi-Supervised Clustering-Based Methods for Degradation Assessment of Nuclear Power Plant Steam Generators," Politecnico Di Milano, 2018.

[33] 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.

Chapter 7 A Multi-Step Ahead Prediction Method for NPP Reactor Coolant Pump Signals

Multi-step ahead prediction can help decision makers improving maintenance planning and minimizing unexpected shutdowns in the nuclear industry. In this Chapter, we develop a hybrid prediction framework based on Ensemble Empirical Mode Decomposition (EEMD) and Long Short-Term Memory (LSTM) neural network. EEMD decomposes time series into a set of components, which allow effectively describing the system dynamics and therefore facilitates the prediction task. Then, LSTM neural network models are developed for predicting the multi-step ahead behavior of the individual EEMD components and the obtained predictions are aggregated to reconstruct the time series. A Tree-structured Parzen Estimator (TPE) algorithm is employed for hyperparameter optimization. The performance of the proposed method is validated by considering different long-term prediction horizons on a practical case study concerning time series data acquired from Reactor Coolant Pumps (RCPs) of Nuclear Power Plants (NPPs). The proposed method shows superior performances with respect to alternative prediction models.

7.1 Hybrid prediction approaches

A problem typically encountered in the development of multi-step ahead prediction models is the data complexity, which means that time series collected from real-world systems can contain 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 the degradation of the prediction performance as the time horizon of the prediction increases [1]. To address this issue, research on hybrid prediction models have been recently carried out to take advantage of the strength of using ensemble of different individual models. For example, Moshkbar-Bakhshayesh and Ghofrani [2] 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 [3] 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. [4] 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. Comprehensive analyses on hybrid approaches for the applications concerning multi-step ahead prediction can be found in [1], [5], [6].

7.2 Signal decomposition methods

This Section presents methods for signal decomposition based on empirical mode decomposition (EMD). Section 7.2.1 and 7.2.2 are dedicated to the original EMD and the EEMD algorithms, respectively.

7.2.1 Empirical Mode Decomposition (EMD)

EMD was proposed by Huang et al. [7] 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) [8]. 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 7.1) [9].

Algorithm 7.1. EMD decomposition process

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] [7]).

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:

$$m_j(t) = \left[U_j(t) + L_j(t) \right] / 2 \tag{7.1}$$

d. Generate the new component $c_{j+1}(t)$ of the next sifting iteration:

$$c_{j+1}(t) = c_j(t) - m_j(t) \tag{7.2}$$

e. Compute the squared difference between two consecutive siftings as follows:

$$SD(j) = \sum_{l=1}^{t} \frac{\left| c_{j+1}(l) - c_j(l) \right|^2}{\left| c_j(l) \right|^2} \tag{7.3}$$

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:

$$r_i(t) = r_i(t) - IMF_i(t) \tag{7.4}$$

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:

$$X_t = \sum_{i=1}^{N_c} IMF_i(t) + r_{N_c}(t) \tag{7.5}$$

7.2.2 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 [10]. To address these problems, EEMD has been developed [11]. 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 [8]. The EEMD algorithm is described in Algorithm 7.2.

Algorithm 7.2. EEMD decomposition process

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:

$$X_t^j = X_t + w_t^j,\ j = 1,2,...,J \tag{7.6}$$

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 $IMF_i^j(t)$:

$$\overline{IMF_i}(t) = \frac{1}{J}\sum_{j=1}^{J} IMF_i^j(t) \tag{7.7}$$

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

$$X_t = \sum_{i=1}^{N_c} \overline{IMF}_i(t) + r_{N_c}(t) \tag{7.8}$$

7.3 Proposed multi-step ahead prediction method

The proposed prediction method is composed of two main parts: decomposition and multi-step ahead prediction (Fig. 7.1). 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.

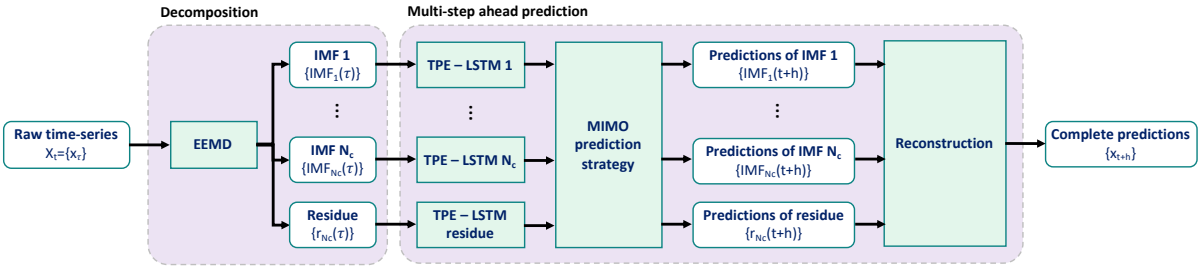
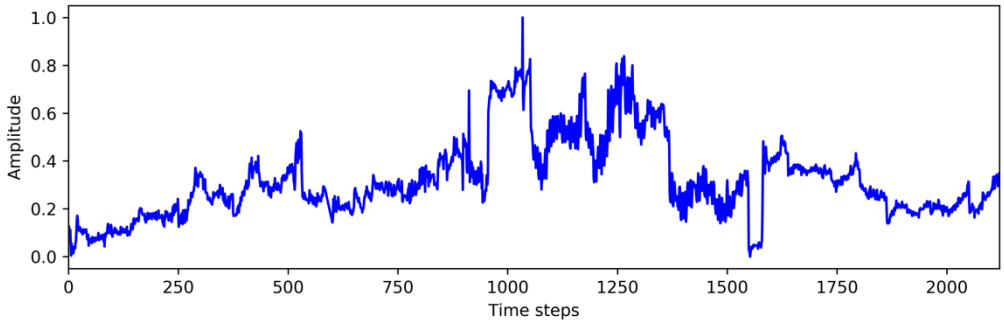


Fig. 7.1. Overview of the proposed multi-step ahead prediction method.

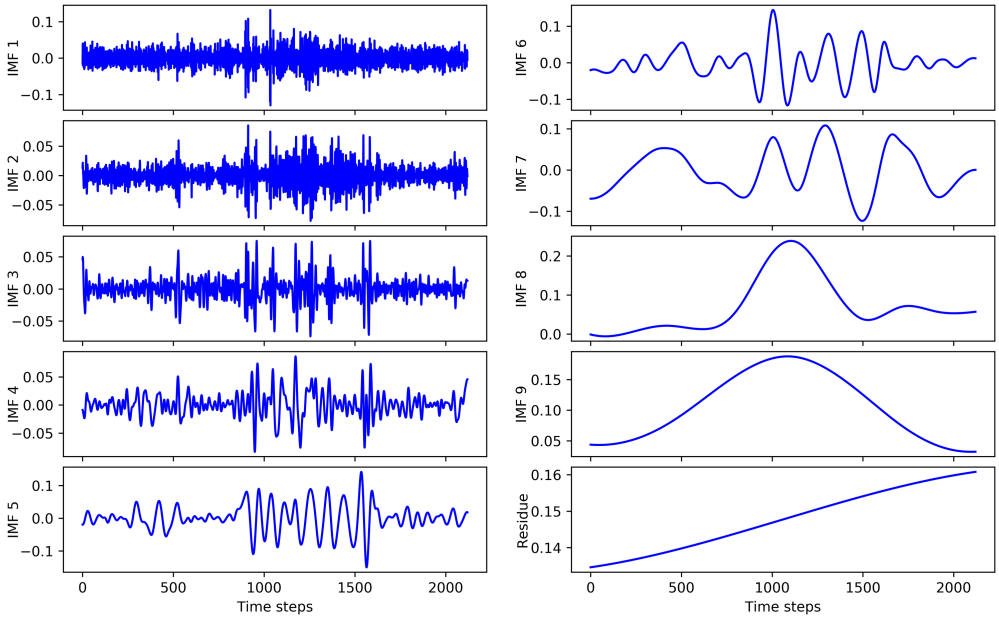
7.3.1 Decomposition of the original time series

EEMD is employed for decomposing the raw time series X_t into separate frequency components $\{IMF_i(t)\}\ (i = 1,2,...,N_c)$. The number of obtained IMFs N_c varies depending on the time series characteristics. Fig. 7.2 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. 7.2(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. 7.2. Time series decomposition by using EEMD.

7.3.2 Multi-step ahead prediction step

In the second stage of the proposed method, we develop a dedicated model for the multi-step ahead prediction 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 Chapter 5, Section 5.3. In the testing phase, the predictions of the components are performed and aggregated to obtain the multi-step ahead prediction of the original time series. The details of the hyperparameter optimization during the training phase and the MIMO prediction strategy are described in the following

Sections.

A. 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 7.1. 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 7.1. Hyperparameters of the LSTM models optimized by the TPE and possible settings of the proposed method considered.

Hyperparameter	Type of distribution	Possible setting
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]

B. MIMO prediction strategy

In general, there are three widely used strategies addressing multi-step ahead prediction: Recursive, Direct and MIMO [12], [13]. Each strategy is characterized by different trade-offs between accuracy and complexity. In this work, we employ the MIMO strategy, since using only one model with multiple outputs offers two main advantages: 1) avoiding error accumulation in long-term predictions; 2) reduction of the training computational cost [13]. The main difficulty to be addressed in developing a MIMO prediction model is the selection of the appropriate model configurations, which in this work is handled by TPE.

The MIMO approach (also known as the Parallel approach) aims at simultaneously predicting multiple future observations by using one single predictor [14]. It is illustrated in Fig. 7.3, where $\{f,[\theta]\}$ denote the predictor and its hyperparameters, respectively, and *d* the embedding dimension which is set by using the False Nearest Neighbor (FNN) approach [15].

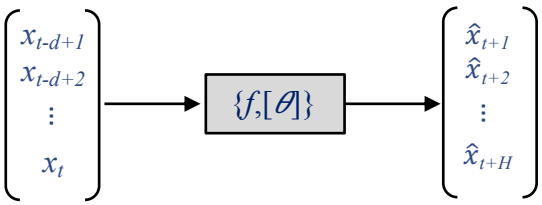
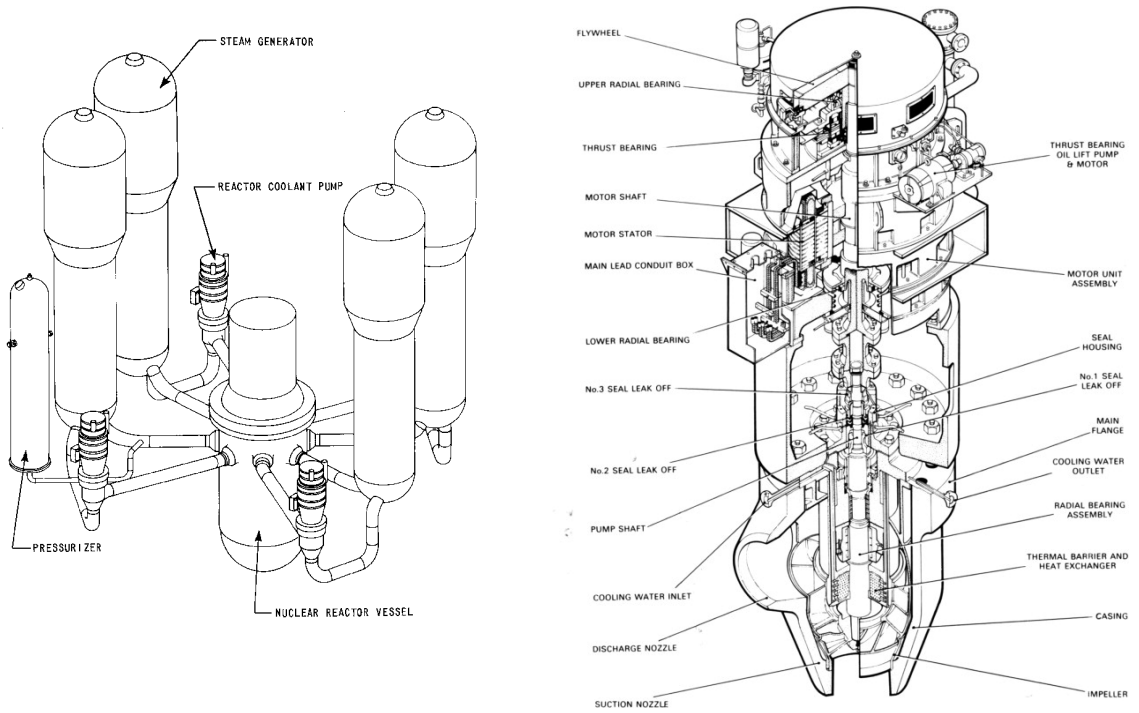


Fig. 7.3. Multi-step ahead prediction model with MIMO strategy.

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

We consider the Reactor Coolant Pumps (RCPs) of a NPP, which 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. 7.4 depicts the structures of the RCS and the RCP of a Pressurized Water Reactor (PWR).



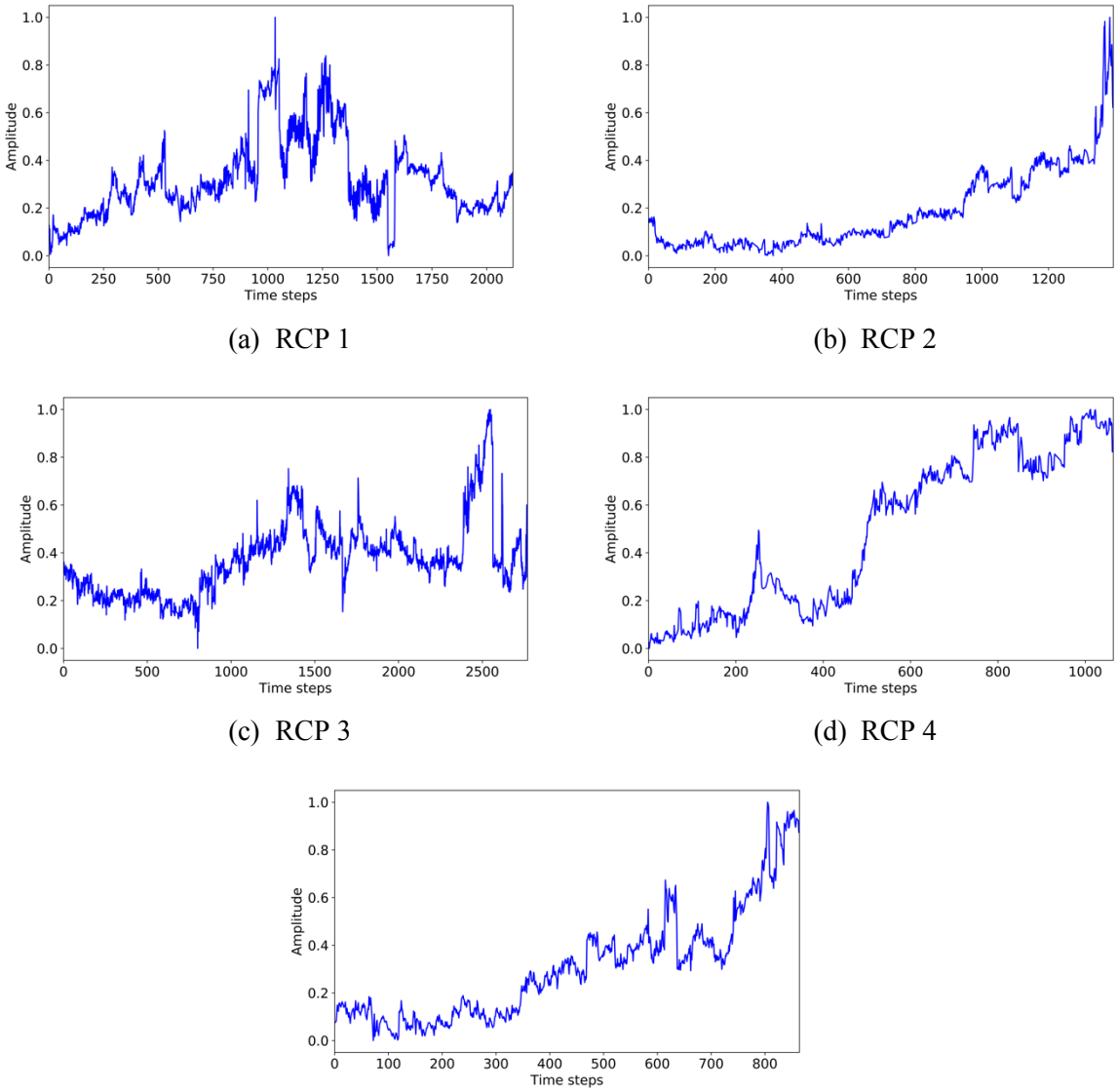
(a) Reactor Coolant System (b) Reactor Coolant Pump

Fig. 7.4. Detailed structures of the PWR Reactor Coolant System (RCS) and Reactor Coolant Pump (RCP). The image has been taken from [16].

One of the most vulnerable components of a RCP is the shaft seal system, which is composed of three mechanical seals located between the electric motor and the impeller, as shown in Fig. 7.4(b). The shaft seal system plays an important role in limiting the leakage of the primary circuit to the ambient environment, which are collected and routed to the seal leakoff system [17]. A failure of the shaft seal system can cause the loss of the reactor primary coolant, which can potentially lead to catastrophic consequences [18]. 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.

We consider five different scenarios of leakage flow from the first seal of the RCPs. The data have

been acquired from real RCPs. The time series are measured at a 4-hour interval, starting from different time instances and for different durations, as shown in Fig. 7.5. The durations of the time series are listed in Table 7.2. For each scenario, the time series is divided into two subsets: the first 70% of the time series is used as training set for developing the prediction models and the latter 30% as test set for evaluating the model performance. For confidentiality reasons, the names of the NPPs are not mentioned and the time series data are normalized from 0 to 1. A different model is developed for each one of the five time series.



(a) RCP 1

(b) RCP 2

(c) RCP 3

(d) RCP 4

(e) RCP 5

Fig. 7.5. The RCP leakage flow time series corresponding to the five RCP leakage scenarios considered in the case study.

Table 7.2. Number of observations available in the five time series.

RCP	1	2	3	4	5
Number of observations	2120	1394	2770	1064	864

7.5 Results and discussion

We carry out the validation of the proposed method with three performance evaluations considering different aspects of the methodology structure. Firstly, the effectiveness of the decomposition algorithm, i.e. EEMD, in improving the prediction performance is evaluated. Secondly, we employ a multivariate prediction model to validate the viability of our hybrid prediction framework, which integrates several univariate models. The final experiment focuses on the performance evaluation of the LSTM network, the central part of our method for multi-step ahead prediction. A prediction model based on Echo State Network (ESN) is employed for benchmarking purposes.

For each experiment, we consider three different prediction horizons to assess the prediction capability of the proposed method, including 6 steps (1 day), 12 steps (2 days) and 18 steps (3 days) ahead. The performance of the prediction models are measured with respect to three accuracy metrics, including RMSE (as stated in Chapter 5, Section 5.3), Mean Absolute Percentage Error (MAPE) and Mean Absolute Scaled Error (MASE). The definitions of MAPE and MASE are given as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{x}_i - x_i}{x_i}\right| \times 100\%, \tag{7.9}$$

$$MASE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\hat{x}_i - x_i|}{\frac{1}{N-1}\sum_{j=2}^{N}|x_j - x_i|}\right), \tag{7.10}$$

where N is the number of test observations, x and $\hat{x}$ are the observed and predicted values, respectively. For the computational point of view, all of the experiments are implemented on 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).

7.5.1 Effectiveness of applying the decomposition technique for multi-step ahead prediction

In order to validate the EEMD, we employ a comparative model which is obtained by removing the EEMD module from the proposed method, as illustrated in Fig. 7.6. In this comparative model, the original time series are directly fed to the LSTM prediction model, with the hyperparameters optimized by TPE, as described in Section 7.3.2.A. It is important to note that the LSTM model is constructed with two LSTM layers consisting of 64 neurons for each layer. The predictions with the three horizons

are performed using the MIMO strategy for all of the time series scenarios.

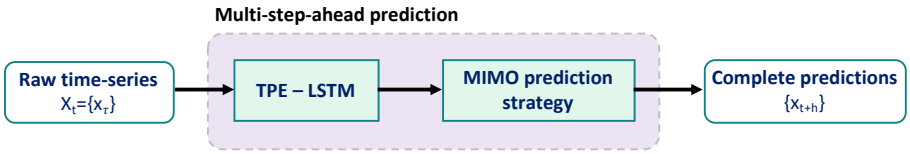
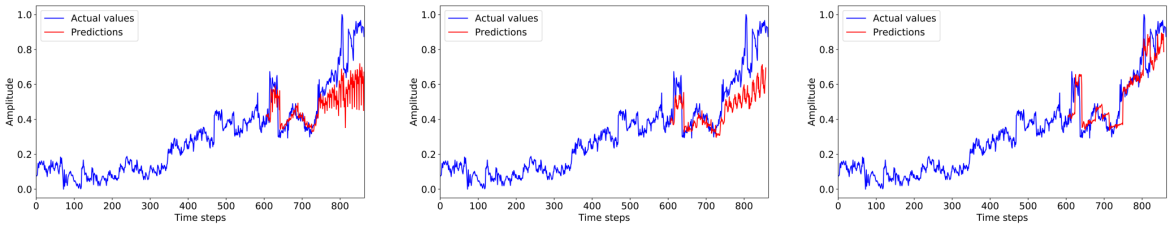


Fig. 7.6. Compared prediction model without using EEMD.

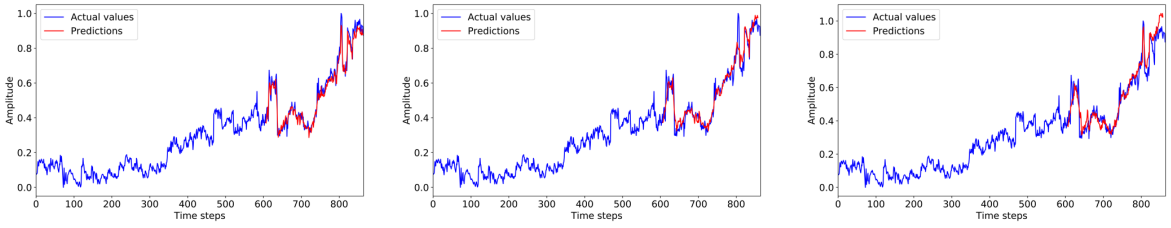
Fig. 7.7 depicts the prediction results for different horizons obtained by the comparative model (Fig. 7.7(a) – (c)) and the proposed model (Fig. 7.7(d) – (f)) for the RCP 5 time series. In each sub-figure, the predicted values are shown as the red solid line, whereas the blue line depicts the actual observations. Fig. 7.7 shows that the predictions of the proposed method are highly accurate and close to the actual values, whereas those of the comparative model are not so accurate in all of the three cases of the prediction horizon. The overall comparison results for all of the time series scenarios are summarized in Table 7.3. The more accurate results (the lower values of the metrics) are highlighted in bold. The results show that the proposed method outperforms the comparative model with a significant improvement in the prediction accuracy, implying the important contribution of the EEMD module to the proposed method for dealing with multi-step ahead predictions.



(a) Model without decomposition – 6 steps ahead

(b) Model without decomposition – 12 steps ahead

(c) Model without decomposition – 18 steps ahead



(d) Proposed model – 6 steps ahead

(e) Proposed model – 12 steps ahead

(f) Proposed model – 18 steps ahead

Fig. 7.7. Results of the multi-step ahead predictions for RCP 5.

Table 7.3. Comparison results of the proposed model and the model without using the EEMD decomposition.

Data scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Without decomposition	0.0405	13.8939	1.6168	0.0608	30.1245	2.3199	0.0667	30.4937	2.6540
	Proposed model	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Without decomposition	0.0776	11.6117	3.5690	0.0897	18.9838	5.2261	0.0893	16.2510	4.5966
	Proposed model	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Without decomposition	0.0627	7.9651	1.7586	0.0868	11.1782	2.5560	0.1081	14.2730	3.6001
	Proposed model	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Without decomposition	0.0568	5.4109	3.1283	0.0730	6.9583	4.1817	0.0891	8.3991	4.9783
	Proposed model	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Without decomposition	0.1583	16.7301	4.1357	0.1651	18.9645	4.5333	0.0988	12.9969	2.5915
	Proposed model	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**

7.5.2 Univariate model versus multivariate model

This experiment focuses on the evaluation of the use of the proposed hybrid framework, in which several univariate prediction models are developed for all of the decomposed components obtained from EEMD, as mentioned in Section 7.3.2. We employ a prediction model based on a multivariate LSTM network for comparison purposes, as illustrated in Fig. 7.8. Specifically, the multivariate LSTM model uses all of the decomposed IMFs as the model inputs and performs predictions using the MIMO strategy. The TPE algorithm is used to select the optimal model hyperparameters.

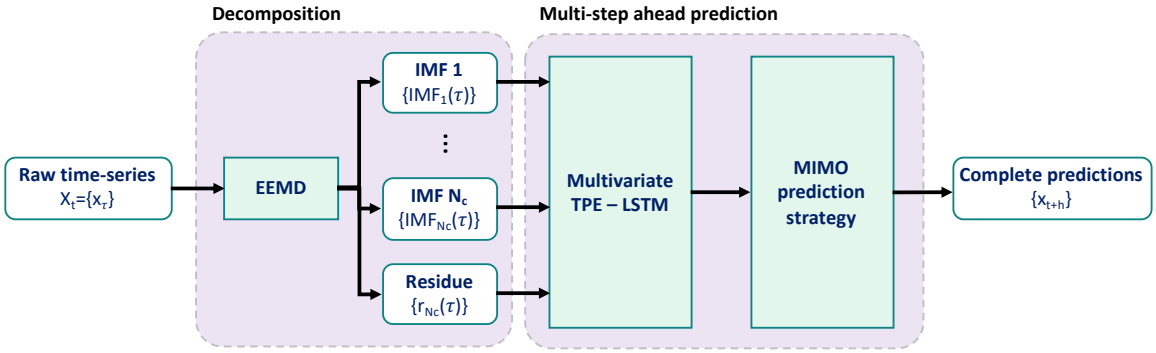
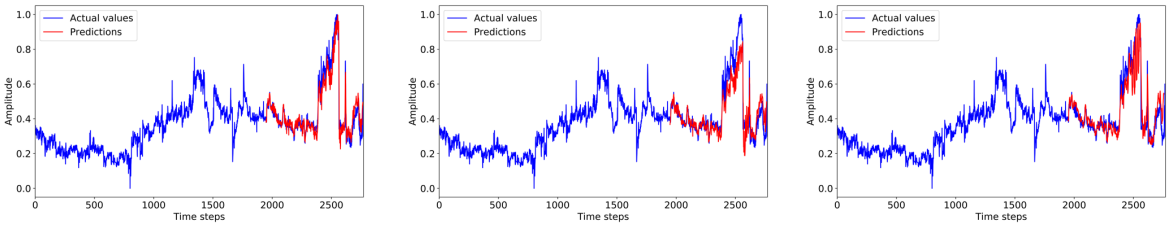


Fig. 7.8. Compared prediction model using a multivariate LSTM network.

The prediction results obtained by the proposed and comparative models for the RCP 3 and 4 scenarios are illustrated in Figs. 7.9 and 7.10, respectively. Table 7.4 summarizes the performance comparison of the two models for all of the data scenarios. As can be seen in Fig. 7.10, the prediction results of the multivariate LSTM model seem quite inaccurate, particularly in Fig. 7.10(c) with a lot of noisy spikes in the predictions. This can be explained by the fact that the data trend of the RCP 4 time series is complicated, strongly nonlinear and nonstationary, making it difficult for a single model to achieve good predictions. On the contrary, the prediction results of the proposed method, Figs. 7.10(d)

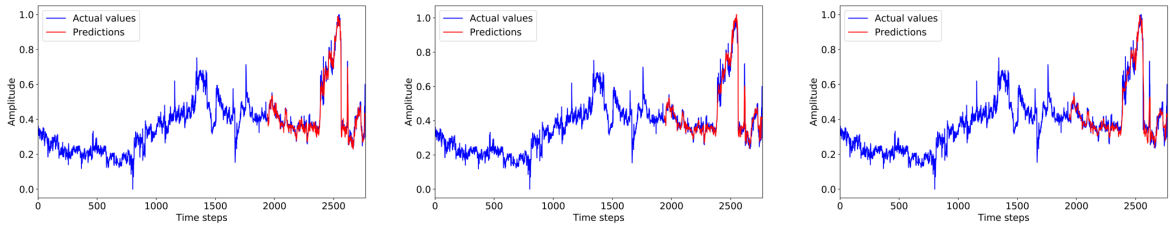
– (f), are remarkably accurate. In Table 7.4, it appears that the hybrid framework provides better results for the whole tests on the five data scenarios than the multivariate model. This indicates that using the hybrid framework integrating univariate prediction models can achieve more accurate multi-step ahead predictions.



(a) Multivariate LSTM model – 6 steps ahead

(b) Multivariate LSTM model – 12 steps ahead

(c) Multivariate LSTM model – 18 steps ahead

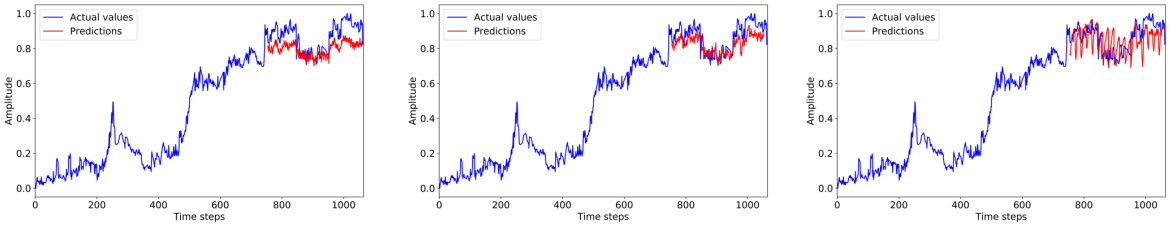


(d) Proposed model – 6 steps ahead

(e) Proposed model – 12 steps ahead

(f) Proposed model – 18 steps ahead

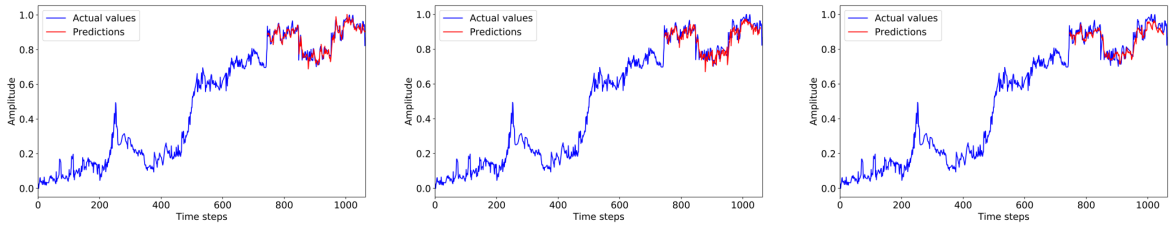
Fig. 7.9. Results of the multi-step ahead predictions for RCP 3.



(a) Multivariate LSTM model – 6 steps ahead

(b) Multivariate LSTM model – 12 steps ahead

(c) Multivariate LSTM model – 18 steps ahead



(d) Proposed model – 6 steps ahead

(e) Proposed model – 12 steps ahead

(f) Proposed model – 18 steps ahead

Fig. 7.10. Results of the multi-step ahead predictions for RCP 4.

Table 7.4. Comparison results of the proposed model and the multivariate LSTM model.

Data scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Multivariate LSTM	0.0249	9.3767	1.3214	0.0386	20.3446	1.9242	0.0455	17.8378	2.4970
	Proposed model	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Multivariate LSTM	0.0709	9.2070	2.8689	0.0483	6.3916	1.9585	0.0916	20.2789	6.2618
	Proposed model	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Multivariate LSTM	0.0500	8.1171	1.8032	0.0747	10.8237	2.6785	0.0760	10.5087	2.5134
	Proposed model	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Multivariate LSTM	0.0851	8.1530	5.0001	0.0607	5.6678	3.4487	0.0819	7.7580	4.5046
	Proposed model	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Multivariate LSTM	0.1375	18.8527	3.8056	0.3038	27.1751	6.1985	0.1340	18.1811	3.6363
	Proposed model	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**

7.5.3 Performance evaluation of the LSTM prediction models

LSTM models play a fundamental role in our proposed method for learning complex data mappings, especially long-term dependency, and performing multi-step ahead predictions with the supports of the TPE optimization and the MIMO prediction strategy. In this Section, we validate the prediction performance of the LSTM models via a comparison with another widely used RNN called Echo State Network (ESN).

ESN is a RNN with a sparsely connected hidden layer [19]. 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 [19], [20].

The compared prediction model is developed by replacing the LSTM models with the ESN models, and the rest of the framework is kept unchanged, as illustrated in Fig. 7.11. In this framework, the TPE is used to optimize the two major hyperparameters of the ESN models, including the number of reservoirs and the spectral radius, as described in Table 7.5.

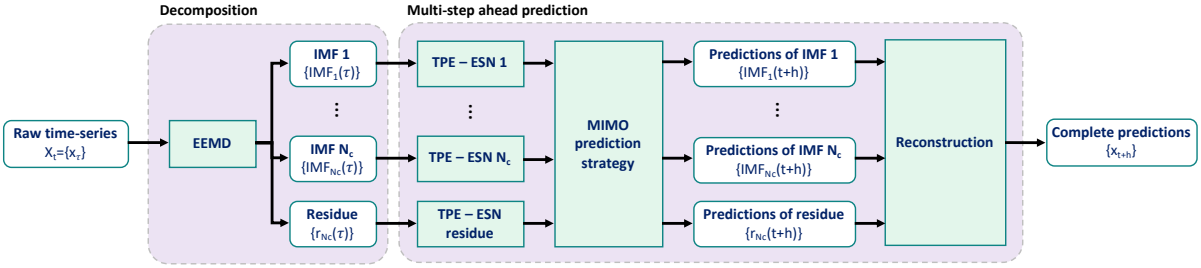
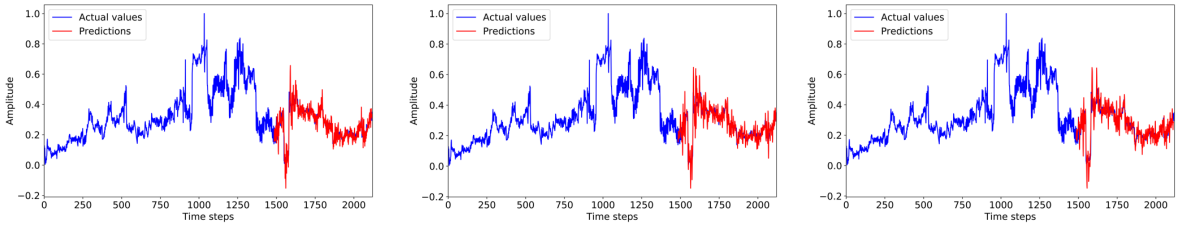


Fig. 7.11. Compared prediction model using the ESN RNNs.

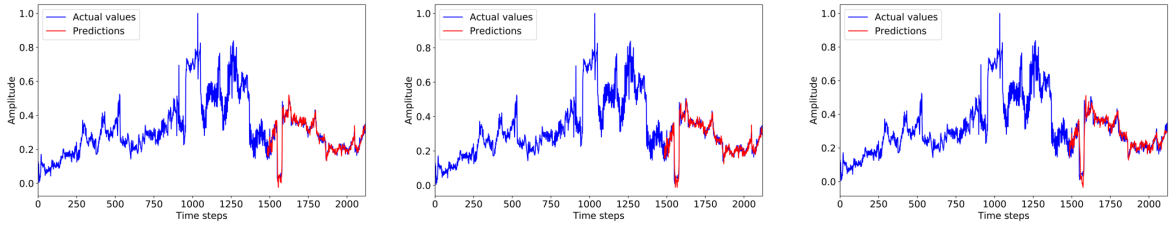
Table 7.5. Hyperparameters of the ESN model.

Hyperparameter	Type of distribution	Possible setting
Number of reservoirs	Uniform integer	[20, 500]
Spectral radius	Uniform float	[0.01, 1]

Figs. 7.12 and 7.13 show the results of the multi-step ahead predictions obtained by the ESN framework and the proposed framework for the RCP 1 and 2 scenarios, respectively. We summarize the overall performance comparison in Table 7.6. According to these results, the prediction framework using LSTMs consistently outperforms the ESN-based framework, achieving a greater accuracy for multi-step ahead predictions. Thus, LSTM is a more suitable choice for the development of a multi-step ahead prediction framework.

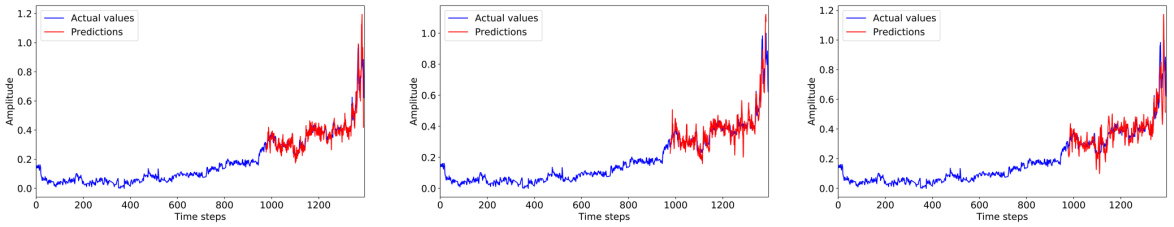


(a) ESN model – 6 steps ahead (b) ESN model – 12 steps ahead (c) ESN model – 18 steps ahead

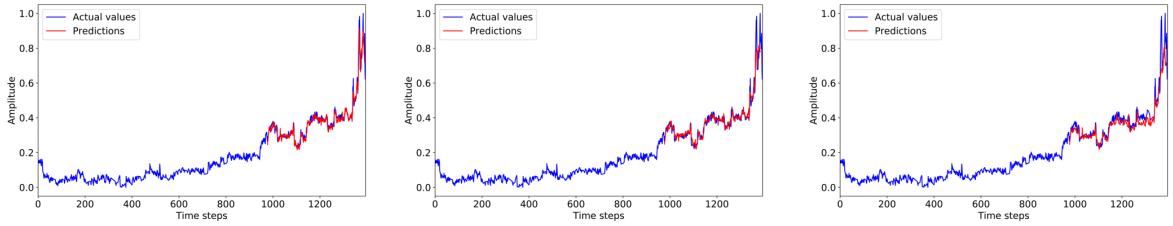


(d) Proposed model – 6 steps ahead (e) Proposed model – 12 steps ahead (f) Proposed model – 18 steps ahead

Fig. 7.12. Results of the multi-step ahead predictions for RCP 1.



(a) ESN model – 6 steps ahead (b) ESN model – 12 steps ahead (c) ESN model – 18 steps ahead



(d) Proposed model – 6 steps ahead (e) Proposed model – 12 steps ahead (f) Proposed model – 18 steps ahead

Fig. 7.13. Results of multi-step ahead predictions for RCP 2.

Table 7.6. Comparison results of the proposed model and the ESN-based prediction model.

Data scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	ESN model	0.0450	18.2183	2.4145	0.0521	20.1742	2.7480	0.0544	21.2928	2.9608
	Proposed model	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	ESN model	0.0496	7.7376	2.2372	0.0511	9.4669	2.6751	0.0672	11.1388	3.1238
	Proposed model	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	ESN model	0.0647	11.3833	2.3444	0.0616	11.1641	2.3267	0.0750	12.7170	2.5916
	Proposed model	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	ESN model	0.0419	3.7228	2.0821	0.0480	4.5109	2.5750	0.0675	6.0974	3.5221
	Proposed model	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	ESN model	0.0380	5.1331	1.0158	0.0578	8.0459	1.5163	0.0835	13.3672	2.3941
	Proposed model	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Z. Shao, F. Chao, S. L. Yang, and K. Le Zhou, "A review of the decomposition methodology for extracting and identifying the fluctuation characteristics in electricity demand forecasting," *Renewable and Sustainable Energy Reviews*. 2017.

[7] 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.

[8] 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.

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

[10] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, 2009.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

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

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

[19] H. Jaeger, "Echo state network," *Scholarpedia*, 2007.

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

Chapter 8 Conclusion

8.1 Original contributions and limitations

The underlying objective of the research work carried out in the Ph.D. Thesis is to develop prognostic methods tailored for different types of components and different information sources available for the model development, and to evaluate their effectiveness on case studies concerning prognostic problems of industrial applications. Specific focus was on two prognostic problems: model-based prognostics for fatigue crack growth prediction and data-driven prognostics for multi-step ahead predictions of the real time series data of NPP SGs and RCPs.

8.1.1 Model-based prognostic methods for fatigue crack growth prediction

The Thesis has presented two model-based prognostic methods for addressing the problems of selecting proper degradation models and using them for predicting the component degradation evolution and the RUL when new observations are collected. The proposed model-based methods have been developed based on the integration of filtering approaches, i.e. recursive Bayesian filtering and PF, and two novel ensemble strategies in which the weights of the individual models in the ensemble are computed based on their prediction performance at previous time steps. The proposed methods have been applied to case studies of fatigue crack growth simulated under different operating conditions and different degradation trajectories. The results have shown that the proposed models are able to accurately predict the degradation states and the RUL under various operating conditions, and outperform individual models in various prognostic performance indicators.

However, both proposed methods have some limitations. First, the performance of the proposed methods should be validated with respect to a real case study of fatigue crack growth and other ensemble approaches to assess the effectiveness of the proposed weighted ensemble strategies. Also, the prediction performance of the proposed ensemble models depends on the individual models considered in the ensemble. More advanced degradation models, such as the FKM Directive of fatigue crack growth [1], can be further investigated to improve the proposed methods.

8.1.2 Multi-step ahead prediction methods of NPP time series data

Multi-step ahead prediction is another strong focus of the Thesis, and two prediction methods based on LSTM deep neural network have been proposed, developed and applied to real case studies concerning different NPP components. The two proposed methods have not only addressed the problem of multi-step ahead prediction of time series data, but also addressed several practical issues, including anomaly detection, automatic hyperparameter optimization and prediction uncertainty quantification.

The first multi-step ahead prediction method has been evaluated with a case study concerning the real WRL measurements of six NPP SGs, which had been collected over a period of 16 consecutive years, whereas the second method has been applied to predict the time series data of different NPP RCPs, with three long-term prediction horizons considered. The results have showed the promising performance of the two proposed methods in adaptively estimating the optimal settings for their network architectures and capturing the underlying long-term dependencies inherent in the given data, for achieving accurate predictions over long-term horizons and outperforming conventional prediction approaches.

A limitation of the two proposed multi-step ahead prediction methods is the lack of degradation-to-failure time information, which would enable the RUL estimation for effective predictive maintenance. Another limitation is the input data for model development, which consider only univariate time series measurements. In fact, sufficient information and data for performing multivariate predictions are not provided, such as the information of the interdependency between measured variables and degradations, the interdependency within the variables, and the maintenance reports of the NPP components. Further investigations of the effectiveness of the proposed methods with multivariate time series measurements are needed.

8.2 Future research and development perspectives

The ultimate goal of the research and developments carried out in this Ph.D. Thesis is to contribute to the reliability, availability, maintainability and safety (RAMS) improvement of industrial components and systems by predictive modelling. However, there are still methodological and technical issues that must be dealt with to provide more effective prognostic systems. Suggestions for future research and development include:

1) For the model-based prognostic methods:
   - Validate the proposed methods with a real case study of fatigue crack growth in industrial components and systems.
   - Perform further comparisons with respect to other ensemble approaches, and integrate advanced PoF models of fatigue crack growth.
2) For the multi-step ahead prediction methods:
   - Develop a multivariate time series prediction method to utilize the great amount of condition monitoring data in NPPs for prognostic purposes.
   - Integrate the proposed methods within a RUL estimation step for PHM and predictive maintenance.
   - Develop a generalized long-term prediction method for different NPP components by using transfer learning [2].
   - Apply the proposed methods to long-term anomaly prediction for anticipating and

reasoning out abnormal phenomena and failures which can occur during plant condition monitoring.

- Provide a reliable prediction uncertainty quantification method for confident decision-making.

References

[1] "FKM-Guideline. Fracture Mechanics Proof of Strength for Engineering Components," *VDMA Publ., 1st Ed. (2001), 2nd Ed. (2004), 3rd Ed.*

[2] L. Yang, S. Hanneke, and J. Carbonell, "A theory of transfer learning with applications to active learning," *Mach. Learn.*, 2013.

Part II

Publications

LIST OF WORKS

1. International Journal Papers

[J4] Hoang-Phuong Nguyen, Piero Baraldi, and Enrico Zio. (2020). A method based on Ensemble Empirical Mode Decomposition and Long Short-Term Memory neural network for multi-step ahead predictions of nuclear power plant signals. Submitted to Applied Energy in February, 2020.

[J3] Hoang-Phuong Nguyen, Jie Liu, and Enrico Zio. (2020). 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. *Applied Soft Computing*, vol. 89, pp. 106116. doi.org/10.1016/j.asoc.2020.106116

[J2] Hoang-Phuong Nguyen, Jie Liu, and Enrico Zio. (2019). Ensemble of models for fatigue crack growth prognostics, *IEEE Access, Special Section on Advances in Prognostics and System Health Management*, vol. 7, pp. 49527 - 49537. doi:10.1109/access.2019.2910611

[J1] Hoang-Phuong Nguyen, Jie Liu, Enrico Zio. (2018). Dynamic-weighted ensemble for fatigue crack degradation state prediction, *Engineering Fracture Mechanics*, vol. 194, p.212-223. doi:10.1016/j.engfracmech.2018.03.013

2. Papers on Proceedings of International Conferences

[C3] Hoang-Phuong Nguyen, Piero Baraldi, Jie Liu, Enrico Zio. (2019). Multi-step-ahead Prognostics Using LSTM Recurrent Neural Network. *The 2019 4th International Conference on System Reliability and Safety (ICSRS2019)*, Rome, Italy, November 20-22. (Best Presentation Award)

[C2] Hoang-Phuong Nguyen, William Fauriat, Jie Liu, Enrico Zio. (2018). A data-driven approach for predicting the remaining useful life of steam generators. *The 2018 3rd International Conference on System Reliability and Safety (ICSRS2018)*, Barcelona, Spain, November 24-26. doi:10.1109/ICSRS.2018.8688716

[C1] Hoang-Phuong Nguyen, Jie Liu, Enrico Zio. (2017). Model ensemble-based prognostic framework for fatigue crack growth prediction. *The 2017 2nd International Conference on System Reliability and Safety (ICSRS2017)*, Milan, Italy, December 20-22. doi:10.1109/ICSRS.2017.8272843.

Paper 1:

Hoang-Phuong Nguyen, Jie Liu, Enrico Zio. (2018). Dynamic-weighted ensemble for fatigue crack degradation state prediction, *Engineering Fracture Mechanics*, vol. 194, p.212-223.

Dynamic-Weighted Ensemble For Fatigue Crack Degradation State Prediction

Hoang-Phuong Nguyen[a], Jie Liu[b,c], Enrico Zio[a,c,d,*]

[a] *Chair on System Science and the Energetic Challenge, EDF Foundation, CentraleSupélec, Université Paris-Saclay, 9 rue Joliot-Curie, 91192 Gif-sur- Yvette, France*

[b] *School of Reliability and Systems Engineering, Beihang University, 37 Xueyuan Road, Haidian, Beijing, China*

[c] *Sino-French Risk Science and Engineering (RISE) Laboratory, Beihang University, 37 Xueyuan Road, Haidian, Beijing, China*

[d] *Dipartimento di Energia, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy*

Abstract

This paper proposes a prognostic framework for online prediction of fatigue crack growth in industrial equipment. The key contribution is the combination of a recursive Bayesian technique and a dynamic-weighted ensemble methodology to integrate multiple stochastic degradation models. To evaluate the effectiveness of the proposed framework, a case study concerning fatigue crack growth under time-varying operation conditions is carried out. The results indicate that the proposed prognostic framework outperforms single crack growth models in terms of prediction accuracy under evolving operating conditions.

Keywords: Prognostics and Health Management (PHM); Recursive Bayesian; Dynamic ensemble; Multiple stochastic degradation; Fatigue crack growth

1. Introduction

Cracks are among the most common degradations in equipment of several major industries, including manufacturing [1, 2], construction [3, 4], aerospace [5-7], automotive [8, 9], energy [10, 11], etc. A study conducted by the American Society of Civil Engineers (ASCE) [4] has revealed that more than 80% of the collapses of American bridges in steel were caused by fatigue and fracture in structural elements. In [5], it has been shown that in aerospace industry, cracks develop in most critical components of rotorcrafts, such as the main rotor blade, the major cabin frame cap splice, and the tail boom. These unexpected degradations increase the operation risk and can cause severe economic losses in case of breakdowns [12-15]. Thus, for the past several decades, the development of reliable prognostic systems to accurately analyze and estimate the crack propagation in an equipment has attracted the attention of industrial practitioners and researchers.

Some prognostic models have been developed using historical degradation data from a population of similar equipment, whereas the real-time condition monitoring data of the specific equipment were

not considered [16-19]. These historical information, however, may not be always available in practical industrial systems, especially for newly produced equipment or expensive components where the data acquisition costs too much [20]. More importantly, different practical operational conditions, such as load, temperature, and speed, could significantly impact on the rate of the degradation processes, which makes each specific system present a particular degradation trajectory [21]. Therefore, it is important to include the condition monitoring data of the targeted equipment. To address this issue, Cadini et al. [14] introduced a failure prognostic method for fatigue crack growth prediction using a stochastic crack growth model and a Bayesian technique to dynamically update the degradation state from a sequence of monitored measurements. In this sense, recursive Bayesian algorithms are potentially suitable for model-based prognostic frameworks. Indeed, the prior distribution of the degradation states can be combined with the likelihood of the monitored measurements for updating the posterior distribution of the states adaptively when new measurements are available. In [22], Boris et al. presented a prognostic method based on a Bayesian technique to dynamically update the stress intensive range of the physical degradation model at each load cycle until failure, using the condition monitoring measurements. In another study, a comprehensive architecture for both fault detection and isolation (FDI), and failure prognosis for a UH-60 planetary carrier plate was carried out by exploiting a non-linear degradation model and a Bayesian variant, to effectively detect abnormal conditions and predict online the crack depth evolution of the equipment [23].

In practice, the performance of online prognostic models for fatigue crack growth heavily depends on the adopted physics-of-failure model and it is very important to figure out an appropriate modelling framework for a specific degradation process under time-varying operation conditions. To address this issue, numerous fatigue crack growth models have been extensively studied [24-30]. In [31], a comparison of stochastic fatigue crack growth models including the Markov chain model, the Yang's power law model, and a polynomial model were carried out. The results showed that each degradation model has its own range of applicability, and only fits a certain particular degradation process. To the knowledge of the authors, there is no general consensus on a comprehensive prognostic model for fatigue crack growth under different degradation processes. Recently, in the applications of Lithium-ion battery prognostics, hybrid and multi-degradation model ensembles have gained interest because of higher accuracy and better generalization capability than individual degradation models [32, 33]. The basic idea behind these empirical frameworks is to find a set of diverse degradation models which cover different situations so that they complement each other. In [33], an interacting multiple model particle filter (IMMPF) was introduced to combine the estimations from three battery capacity degradation models. The study concluded that the interacting multiple model can achieve higher robustness in terms of smaller estimation errors and more stable performance than a single model.

In this paper, a prognostic framework for fatigue crack growth is proposed by integrating a recursive Bayesian technique and a dynamic ensemble. The degradation state of the component is estimated based on the condition monitoring data collected until the current load cycle, and short-term

degradation state prediction is performed to anticipate and proactively prevent sudden breakdowns of the component in a near future. The key contribution of the work is the dynamic ensemble which combines different crack evolution models with dynamic weights. The dynamic weights are computed based on the historical estimation error for a predefined number of the latest load cycles. To the authors' knowledge, this ensemble framework has been here developed and applied for the first time for a prognostic problem of fatigue crack growth. To validate the performance of the proposed framework, a case study concerning fatigue crack growth with evolving operation conditions is carried out and the results are compared with those obtained by applying single degradation models.

The rest of this paper is organized as follows. Section 2 introduces the degradation models for fatigue crack growth and details the proposed prognostic framework. Section 3 describes the illustrative case study of fatigue crack growth with different load conditions. Finally, Section 4 concludes the study.

2. Proposed prognostic framework

Fig. 1 illustrates an overall flow diagram of the proposed prognostic framework, which consists of the following two main stages: current degradation state estimation based on a recursive Bayesian technique and dynamic-weighted ensemble for prediction. More details about the proposed prognostic framework are described in the following sections.

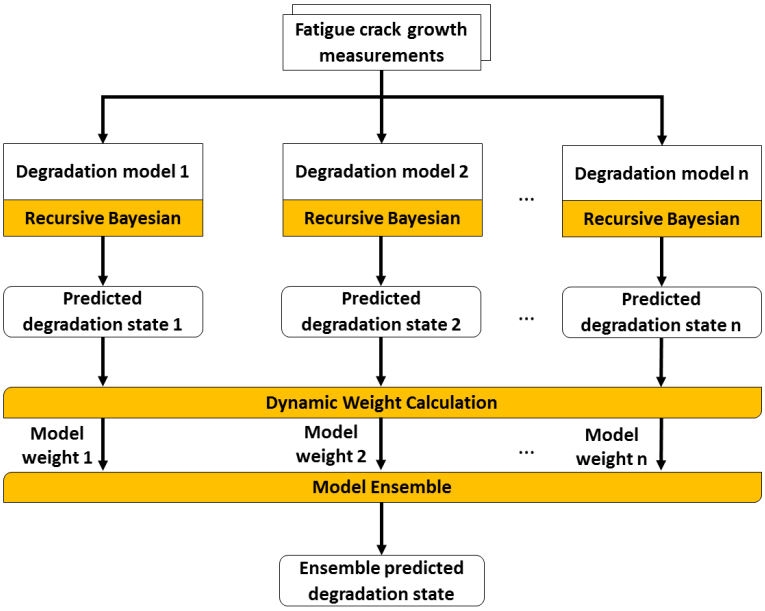


Fig. 1. Flow diagram of the proposed prognostic framework.

2.1. Recursive Bayesian technique for state estimation and parameter identification

2.1.1. State estimation with fixed parameters

In this section, we first describe the technique for estimating the degradation state at time t $\{x_t, t \in \mathrm{N}\}$, given the available condition monitoring measurements $\{z_t, t \in \mathrm{N}\}$. We consider a state-space model describing the evolution of the degradation state x_t and the measurement z_t at the inspection time t, given by:

$$x_t = f_t(x_{t-1}, \omega_{t-1}), \tag{1}$$

$$z_t = g_t(x_t, v_t) \tag{2}$$

where f_t denotes the state transition function and g_t is the measurement function; ω_{t-1} and v_t are independent identically distributed (i.i.d.) state noise and measurement noise sequences, respectively. The current degradation state x_t depends only on the previous state x_{t-1}, and not on the states at previous times. This represents a first-order Markov process with independent degradation increments. The Bayesian technique can be used for estimating the system state x_t at time t by constructing the posterior probability density function (pdf) $p(x_t \mid z_{1:t})$ based on the state transition function and the monitored measurements. More specifically, the system state is recursively estimated by performing the following two steps, namely *prediction* and *update*.

The prediction step involves using both the previous state estimation x_{t-1} and the state transition model in Eq. (1) to estimate the *prior* distribution of the current state x_t via the Chapman-Kolmogorov equation, as follows:

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}, z_{1:t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \tag{3}$$

where $p(x_t \mid x_{t-1})$ is the conditional probability distribution defined by the state model in Eq. (1). The initial distribution $p(x_0 \mid z_0) = p(x_0)$ is assumed to be available and is known as the prior. In the update step, the new measurement z_t is used for obtaining the *posterior* state pdf $p(x_t \mid z_{1:t})$ by applying Bayes' theorem, as follows:

$$p(x_t \mid z_{1:t}) = \frac{p(x_t \mid z_{1:t-1}) p(z_t \mid x_t)}{p(z_t \mid z_{1:t-1})} \tag{4}$$

where $p(z_t \mid x_t)$ is the likelihood function defined by the measurement model in Eq. (2) and $p(z_t \mid z_{1:t-1})$ is a normalizing constant which is defined by:

$$p(z_t \mid z_{1:t-1}) = \int p(x_t \mid z_{1:t-1}) p(z_t \mid x_t) dx_t \tag{5}$$

2.1.2. Sequential parameter identification

A main advantage of the proposed prognostic framework lies also in the capability of identifying time-varying parameters from the sequential data, as detailed in this section. Let us add the time-varying parameter θ_t to the state vector at the inspection time t, as follows:

$$y_t = \begin{bmatrix} x_t \\ \theta_t \end{bmatrix} \tag{6}$$

For this new state vector, the state-space system model in Eqs. (1) and (2) can be reformulated as follows:

$$y_t = F(y_{t-1}, \omega_{t-1}), \tag{7}$$

$$z_t = G_t(y_t, v_t) \tag{8}$$

where

$$F(y, \omega) = \begin{bmatrix} f(x, \omega) \\ \theta \end{bmatrix} \tag{9}$$

$$G_t(y, v) = g_t(x, v) \tag{10}$$

Based on the modified model, the posterior joint distribution $p(y_t \mid z_{1:t}) = p(x_t, \theta_t \mid z_{1:t})$ can be obtained by employing the recursive Bayesian algorithm discussed in the previous section. In this circumstance, the initial distribution is assumed independent of θ, that is, $p(y_0) = p(x_0, \theta_0) = p(x_0)$. Finally, the marginal posterior distributions of the system state x_t and the time-varying parameter θ_t can be obtained by:

$$p(x_t \mid z_{1:t}) = \int p(y_t \mid z_{1:t}) d\theta_t = \int p(x_t, \theta_t \mid z_{1:t}) d\theta_t \tag{11}$$

$$p(\theta_t \mid z_{1:t}) = \int p(y_t \mid z_{1:t}) dx_t = \int p(x_t, \theta_t \mid z_{1:t}) dx_t \tag{12}$$

2.1.3. Short-term state prediction

We consider the degradation state estimation not only at the current load cycle t, i.e. state regression, but also the prediction at fixed prediction horizons $T = t + \tau$, i.e. short-term prediction. In the following case study, three scenarios are considered: 1) degradation state regression at the current time $T = t$; 2) short-term state prediction at time $T = t + 100$; and 3) short-term state prediction at time $T = t + 300$.

2.2. Fatigue crack growth models

For applying the Bayesian recursive technique for state estimation to the fatigue crack growth process, four stochastic degradation models of fatigue crack growth are considered, including Paris-Erdogan, polynomial, global function-based, and curve fitting technique-based models.

2.2.1. Paris-Erdogan model

One well known degradation model for fatigue crack growth is the Paris-Erdogan law [34], which represents the dependence between the crack growth rate dx/dN and the Irwin's stress intensity factor ΔK [35] as follows:

$$\frac{dx}{dN} = C(\Delta K)^m \tag{13}$$

where x is the crack length, C and m are material constants, and N is the number of fatigue load cycles. In this study, the Paris-Erdogan model is employed for considering an infinite plate with a center crack subjected to a sinusoidally applied stress σ, where the geometric factor is equal to 1 and the stress intensity factor ΔK is defined as follows [35]:

$$\Delta K = \Delta\sigma\sqrt{\pi x} \tag{14}$$

where $\Delta\sigma$ is the cyclic stress amplitude.

To take into consideration the statistical variability of the crack growth rate in practice, Myötyri et al. [36] introduced a stochastic variant of the Paris-Erdogan model by using a process intrinsic stochasticity, given by:

$$\frac{dx}{dN} = e^{\omega} C(\Delta K)^m \tag{15}$$

where $\omega \sim N(0,\sigma_\omega^2)$ is an additional white Gaussian noise. For Δt sufficiently small, the state-space model in Eq. (15) can be discretized as follows:

$$x_t = x_{t-1} + e^{\omega} C(\Delta K)^m \Delta t \tag{16}$$

which represents a non-linear Markov process with independent evolution of the degradation state x.

2.2.2. Polynomial model

A mismatch of crack growth models based on power function with the median crack growth curve has been discovered [31, 37]. To address this issue, a crack growth model based on polynomial function was proposed as follows [31]:

$$\frac{dx}{dN} = e^{\omega}(p_0 + p_1 x + p_2 x^2) \tag{17}$$

where $p_i, i = 0,...,2$ are polynomial constants. The stress intensity factor ΔK is not considered in this model [31]. In other studies, the polynomial model was shown to yield the best fit in the linear least square stage of the degradation process [32, 33].

The Markov process representation for a polynomial function-based crack growth model can be defined as follows:

$$x_t = x_{t-1} + e^{\omega}(p_0 + p_1 x + p_2 x^2)\Delta t \tag{18}$$

2.2.3. Global function

Despite the fact that the Paris-Erdogan model and polynomial model show satisfactory performance in the description of the fatigue crack growth process, the effects of equipment geometry on the degradation process have not been taken into consideration [30]. To tackle this shortcoming, Hossien et al. [30] introduced a global function by further reformulating the model for stress intensity range in Eq. (14), using a geometric factor defined as follows:

$$\Delta K = h(x)\Delta\sigma\sqrt{\pi x} \tag{19}$$

where *h(x)* is the geometric factor. In this work, a case study considering a center-cracked plate under uniform tension is considered for validating the global function-based crack growth model, in which the geometric factor *h(x)* is defined by [38]:

$$h(x) = 1 + 0.128\left(\frac{x}{w}\right) - 0.288\left(\frac{x}{w}\right)^2 + 1.523\left(\frac{x}{w}\right)^3 \tag{20}$$

where *w* is the specimen width.

The global function-based crack growth model can be further discretized as follows:

$$x_t = x_{t-1} + e^{\alpha} C\left[1 + 0.128\left(\frac{x}{w}\right) - 0.288\left(\frac{x}{w}\right)^2 + 1.523\left(\frac{x}{w}\right)^3\right]^m (\Delta\sigma\sqrt{\pi x})^m \Delta t \tag{21}$$

2.2.4. Curve fitting function

Still in [30], the authors presented another empirical crack growth model based on a curve fitting technique, given by:

$$\frac{dx}{dN} = e^{\alpha}\left(\frac{1}{C_1 x^m + C_2}\right) \tag{22}$$

where C_1, C_2, m denote the model constants, whose values need to be estimated. According to [30], the stress intensity factor ΔK is not considered in the model.

The authors showed that the crack growth model based on the curve fitting function had capabilities of outperforming the conventional models, such as the power function and the polynomial function, in terms of higher prediction accuracy and lower computation cost. The discretized Markov process representation for a curve fitting function-based crack growth model can be defined as follows:

$$x_t = x_{t-1} + e^{\alpha}\left(\frac{1}{C_1 x^m + C_2}\right)(\Delta K)^m \Delta t \tag{23}$$

2.3. Dynamic-weighted ensemble for prediction

Although various stochastic crack growth models have been studied, it is still difficult to develop a unique accurate model for specific degradation processes, particularly for the ones under time-varying

operation conditions. In this section, a dynamic-weighted ensemble is presented to find automatically the best combinations of multiple crack growth models with respect to their estimation performances. The proposed algorithm consists of the following three steps:

a) Step 1: At the load cycle time t, when the new measurement is available, the time-varying model parameters and the estimated degradation states given by each model are updated by using the recursive Bayesian algorithm described in Section 2.1.

b) Step 2: The estimated degradation state of each individual model is weighted by their estimation errors for the last cycles as follows:

$$w_t^i = \frac{(\phi_t^i)^{-2}}{\sum_i (\phi_t^i)^{-2}} \tag{24}$$

where w_t^i and ϕ_t^i are the dynamic weight and estimation error coefficient of the *i*th model at time *t*, respectively. The estimation error coefficient ϕ_t^i is defined as follows:

$$\phi_t^i = \frac{1}{\delta} \sum_{k=t-\delta}^{t} (z_k - \hat{x}_k^i)^2 \tag{25}$$

where δ is the time horizon (δ = 50 load cycles in the case study that follows) and $\hat{x}_k^i$ is the estimated degradation state of the *i*th model at time *k*. Consequently, the highest weight is given to the model in the ensemble with lowest error at the present time t, and vice versa, the smallest weight in constructing the ensemble is assigned to the least accurate model at time *t*.

c) Step 3: Once the dynamic weights for all models at the current time *t* are calculated, the predicted degradation state of the ensemble is computed as follows:

$$\tilde{x}_T = \sum_i^{N_M} (\hat{x}_T^i \times w_t^i) \tag{26}$$

where $\tilde{x}_T$ is the ensemble predicted state at the load cycle *T* and N_M is the number of degradation models (N_M = 4 in this study). Note that the weights are for the current load cycle as no new measurements are available for any future load cycles.

3. Results and discussion

3.1. Illustrative case study of fatigue crack growth

Numerical simulations of fatigue crack growth have been carried out with an initial crack length x_0 of 10^{-4} mm and a test frequency of 1 Hz. The total number of fatigue load cycles is N = 2000 cycles. To explore the capabilities of the proposed approach under time-varying degradation processes, the fatigue lifetime of the simulated crack growth is split into four continuous and equivalent periods, where the crack length is generated according to different crack growth models as follows:

a) In the first 500-load-cycle period, the Paris-Erdogan model is employed to simulate the crack propagation process as described in Eq. (16). In this regard, the Paris-Erdogan model can provide a linear relationship between $\log(dx/dN)$ and $\log(\Delta K)$, in other words, the stress intensity factor (SIF) range of simulated data is compatible to the Region II (Paris region).
b) In the following period, the polynomial crack growth model in Eq. (18) is used to generate the crack length.
c) The global function-based crack growth model in Eq. (21) is, then, employed for the third period.
d) Finally, in the last period, from the cycle 1501 to 2000, the curve fitting function-based crack growth model in Eq. (23) is utilized.

The model parameters in this work are first initialized based on empirical knowledge, as detailed in Table 1. Subsequently, a Bayesian-based parameter identification approach, as presented in Section 2.1.2, is applied to adaptively update the parameters according to the real-time information from measured data at each load cycle.

Table 1
Detailed settings of model characteristics of fatigue crack growth case studies.

State noise variance	$\sigma_\omega^2 = 0.49$
Measurement noise variance	$\sigma_v^2 = 0.16$
Paris-Erdogan model	$C = 0.1$, $m = 1.3$
Polynomial model	$p_0 = 1.4 \times 10^{-3}, p_1 = 1.5 \times 10^{-3}, p_2 = 1 \times 10^{-5}$
Global function-based model	$C = 0.005$, $m = 0.245$
Curve fitting function-based model	$w = 1$ mm, $C_1 = 250$, $C_2 = 0.3$, $m = -0.7$

For the purpose of extensively validating the effectiveness of the proposed approach for drifting degradation processes, two crack growth profiles under different conditions of load ratio, $R = 0.1$ and $R = 0.15$, are artificially integrated to expand the case study of the simulated crack growth, as illustrated in Fig. 2. Specifically, the load ratio R, or the stress ratio, measures the general influence of the mean stress on the fatigue crack growth behavior, and is defined by the ratio of the minimum to the maximum stresses experienced during a cycle. As the ratio R increases, the fatigue crack growth rate curve is shifted toward higher dx/dN [38].

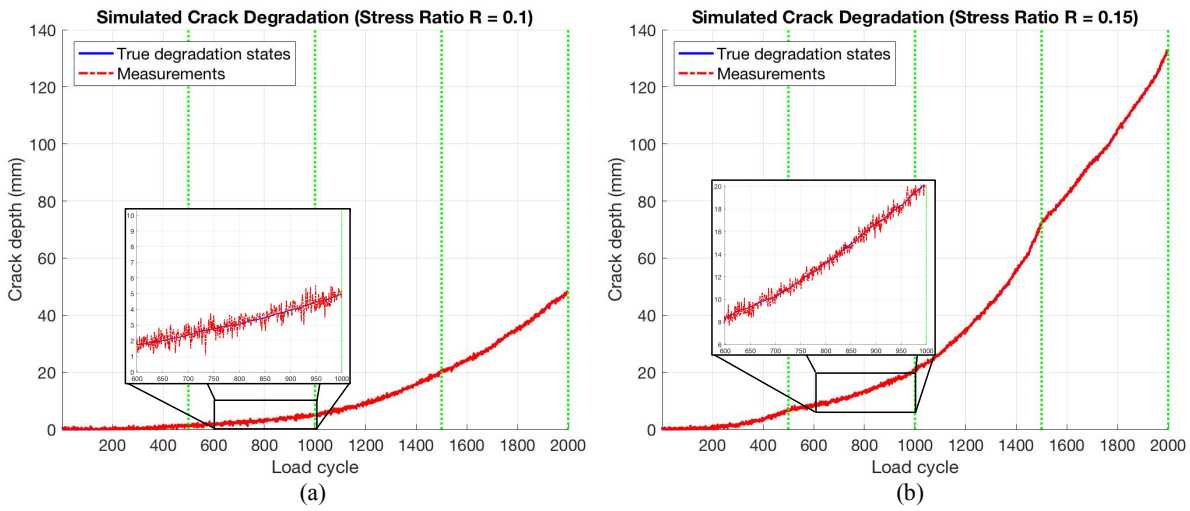


Fig. 2. Simulated crack depth evolution profiles with different load ratios (a) R = 0.1 and (b) R = 0.15.

In order to validate the SIF ranges of the generated crack growth data, an investigation is carried out on the first 500-load-cycle period, where the crack size is very small, and the results are illustrated in Figs. 3 and 4. The state noise is also considered in this investigation. As expected, the results clearly show a linear relationship between $\log(dx/dN)$ and $\log(\Delta K)$ in all cases of stress ratio conditions, which means that the simulated data completely correspond to the Region II (Paris region).

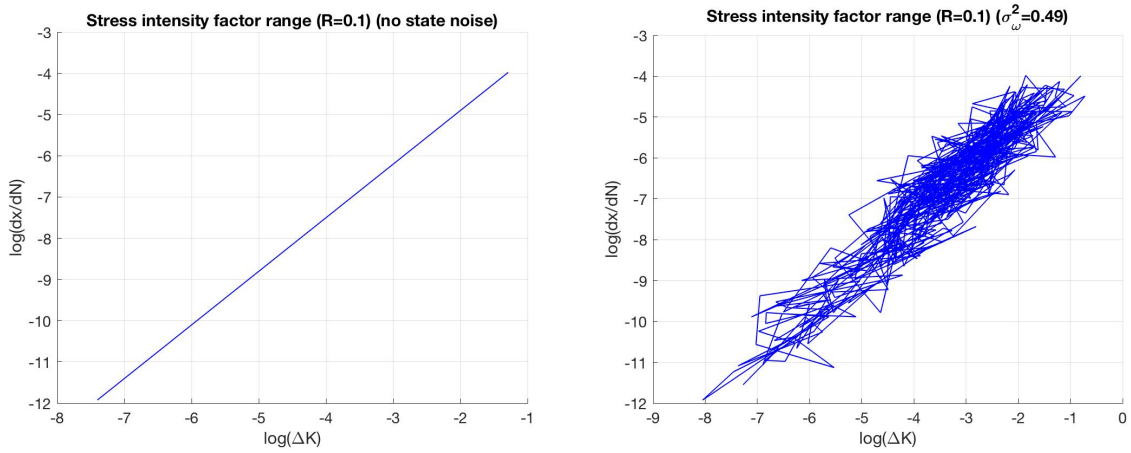


Fig. 3. Stress intensity factor range, ΔK log scale, with R = 0.1. Without state noise considered (left) and with state noise $\sigma_\omega^2 = 0.49$ (right).

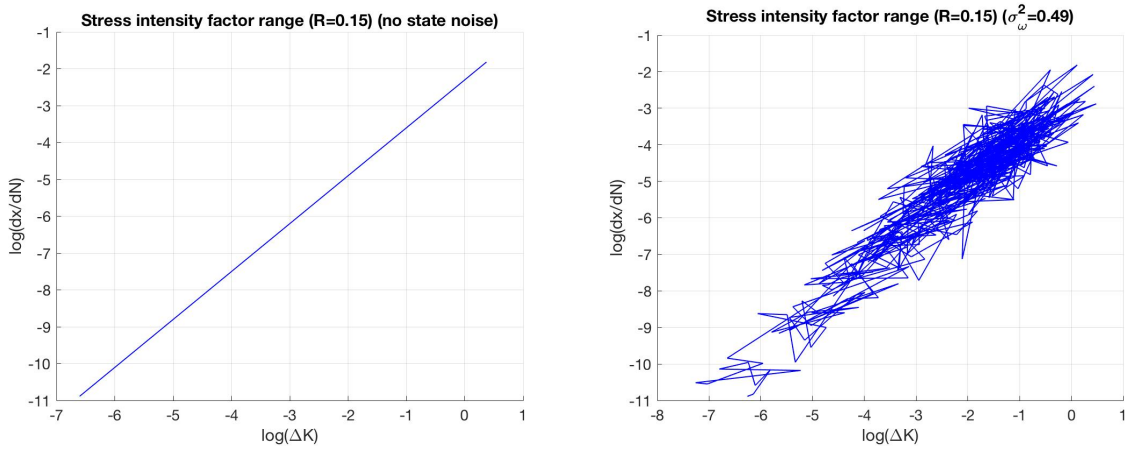


Fig. 4. Stress intensity factor range, ΔK log scale, with R = 0.15. Without state noise considered (left) and with state noise $\sigma_\omega^2 = 0.49$ (right).

3.2. Performance evaluation

The proposed prognostic framework based on recursive Bayesian technique and dynamic-weighted ensemble is applied to determine the best combination of multiple crack growth models in terms of degradation prediction performance. More specifically, when a new monitored measurement is available, the degradation states and crack growth model parameters are estimated online via the Bayesian technique. The four stochastic crack growth models described in Section 2.2 are considered. The weight for each individual model is updated at the current load cycle and the ensemble is obtained by integrating the individual degradation models. The mean square error (MSE) is considered as the performance evaluation index to indicate prediction accuracy:

$$MSE_i = \frac{1}{N}\sum_{t=1}^{N}(x_t - \hat{x}_t^i)^2 \tag{27}$$

where x_t and $\hat{x}_t^i$ denote the true degradation state and the prediction of the *i*th crack growth model at time *t*, respectively.

The degradation state estimations at different load ratios are shown in Figs. 5 and 6. As expected, the estimated degradation states among different models are similar for the current time, as shown in Figs. 5(a) and 6(a), but obviously separated for short-term state predictions. This indicates the effectiveness of the prediction error-based dynamic weights in reflecting the performance of different degradation models. An interesting observation in Figs. 5 and 6 is that the polynomial model can exhibit satisfactory performance when the fatigue crack depth is small, but its performance is rapidly degraded when the crack becomes longer. This is mainly because of the fact that the polynomial function only fits in the linear least square stage of the degradation process, as mentioned in Section 2.2.2. In other words, the polynomial model is able to achieve satisfactory performance only for linear and deterministic fatigue crack growth processes. In contrast, by combining dynamically different

degradation models, the proposed ensemble approach can achieve superior performance to the individual models in predicting the degradation states of fatigue crack growth. Table 2 shows that the proposed ensemble framework outperforms the individual crack growth models, yielding a prediction accuracy of 2.07 and 33.14 in terms of MSE for short-term degradation state prediction at time t+300 under the load ratios R = 0.1 and R = 0.15, respectively.

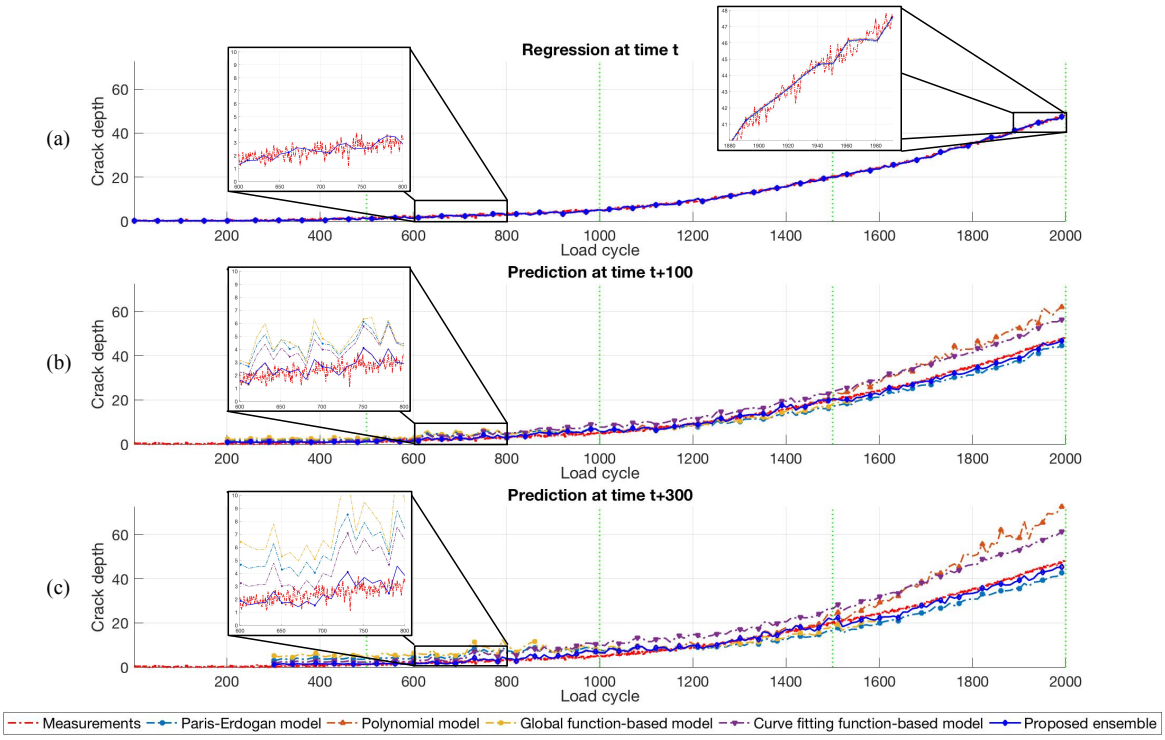


Fig. 5. Estimation of degradation state at load ratio R = 0.1 in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

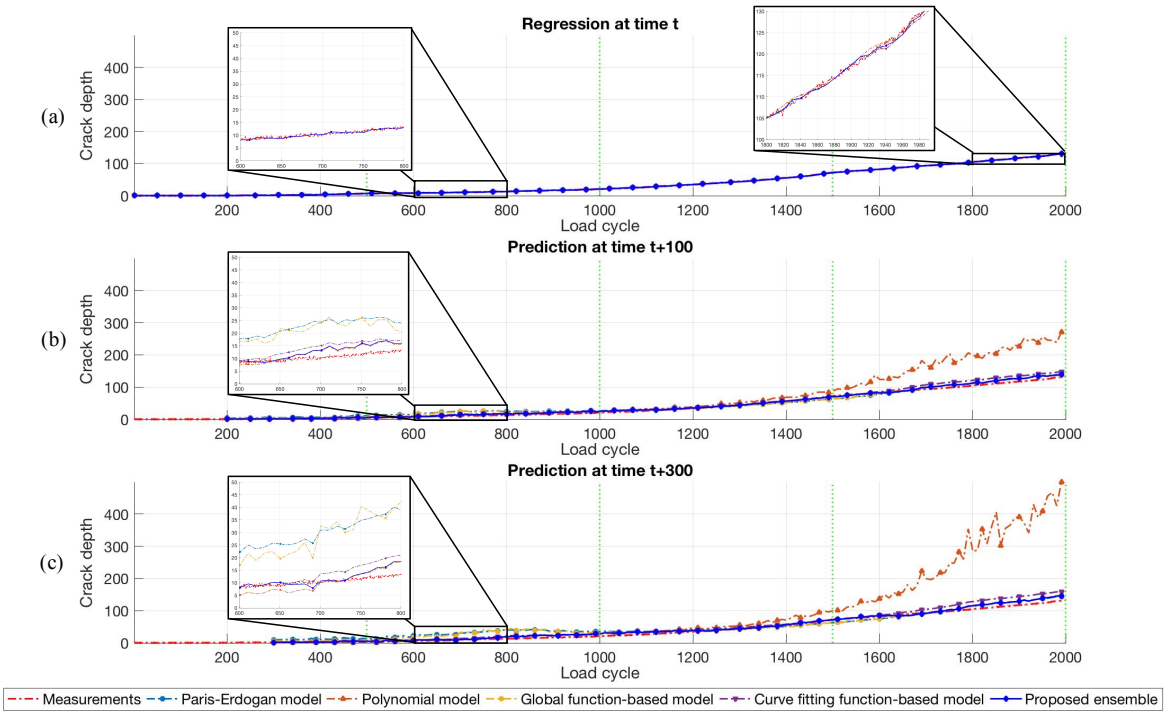


Fig. 6. Estimation of degradation state at load ratio R = 0.15 in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

Table 2
MSE results of the degradation state regression and short-term prediction at different load conditions.

	R = 0.1			R = 0.15		
	t	t+100	t+300	t	t+100	t+300
Paris-Erdogan model	0.10	1.09	12.79	0.15	10.35	151.04
Polynomial function-based model	0.10	4.51	60.90	0.15	278.25	9764.72
Global function-based model	0.10	0.69	11.92	0.15	8.94	140.21
Curve fitting function-based model	0.10	3.54	42.03	0.15	14.33	119.09
Proposed ensemble	0.10	0.38	2.07	0.12	5.23	33.14

The impact of uncertainty on the performance of the proposed framework in case of unknown initial state of degradation has also been investigated. In this case, the monitored data are assumed unavailable from time 1 to t_0 (t_0 = 500 in this study), and the true degradation state of the system x_{t_0} is also assumed unknown. The performance of the proposed approach under different load ratio conditions is shown in Figs. 7 and 8. The dashed line with marker are the predicted degradation states of the proposed approach while the dotted lines are the 95% confidence intervals. The results in Figs. 7 and 8 show that the proposed framework can yield accurate state predictions even without knowledge of the initial

degradation state. In Fig. 8(c), some abnormal spikes in the confidence intervals can be observed. Note that these abnormal spikes exist only in the case study in which the degradation state prediction is performed at time t+300 with no available measurement until the time t_0 = 500. For the relative shorter prediction horizon, no spike is observed. This can be explained by the fact that for a long-term prediction time t+300, the performances of individual models in the ensemble can be unexpectedly degraded due to the propagation of uncertainty. As can be seen in Figs. 9 and 10 below, in the last 400 load cycles, the performance diversity between the polynomial model and the others is clearly observed, and, furthermore, the variance of the polynomial model also rapidly increases, resulting in unsatisfactory performance in the estimation of the confidence intervals.

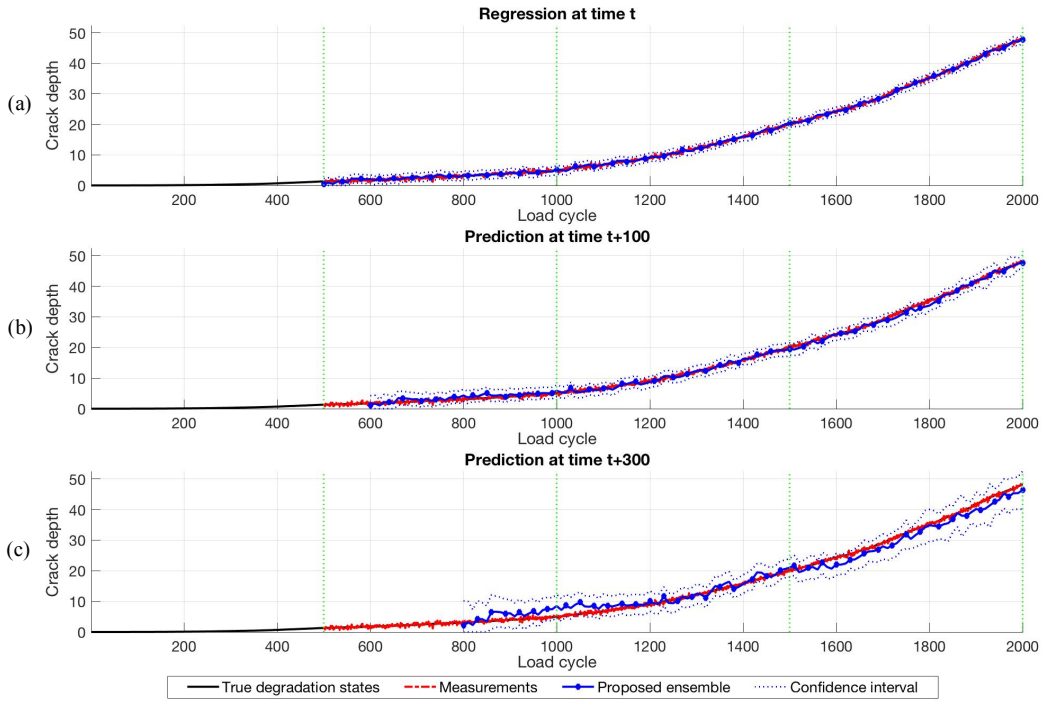


Fig. 7. Estimation of degradation state with measurements that are not available until the time t_0 = 500 at load ratio R = 0.1, in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

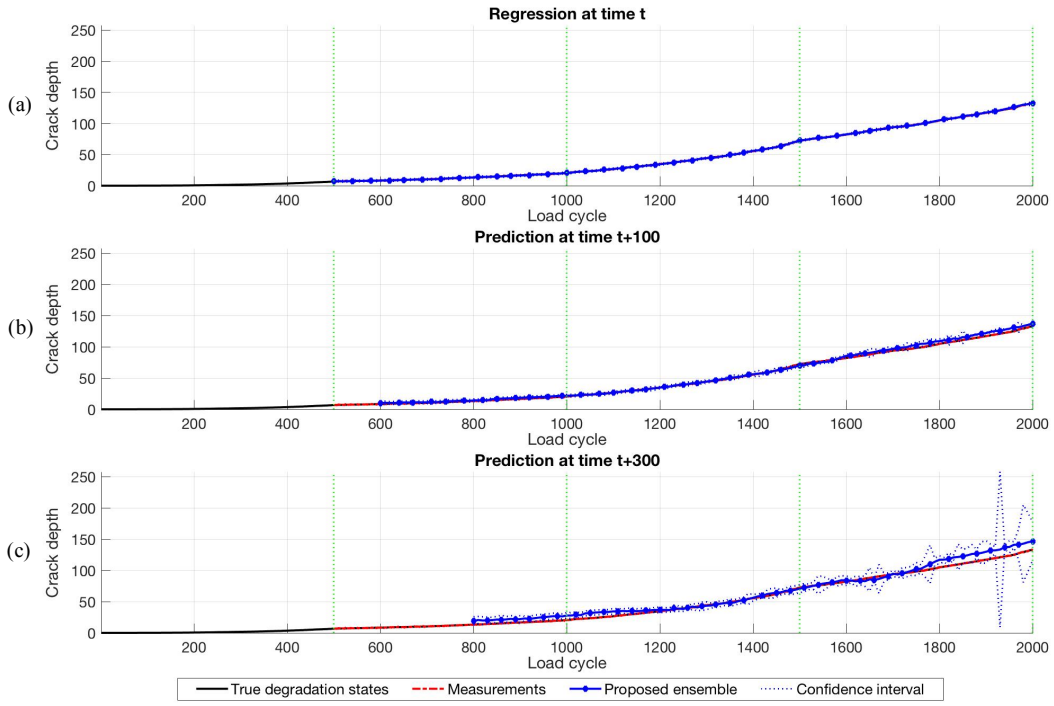


Fig. 8. Estimation of degradation state with measurements that are not available until the time t_0 = 500 at load ratio R = 0.15, in three scenarios: (a) Regression at time t; (b) Prediction at time t+100; and (c) Prediction at time t+300.

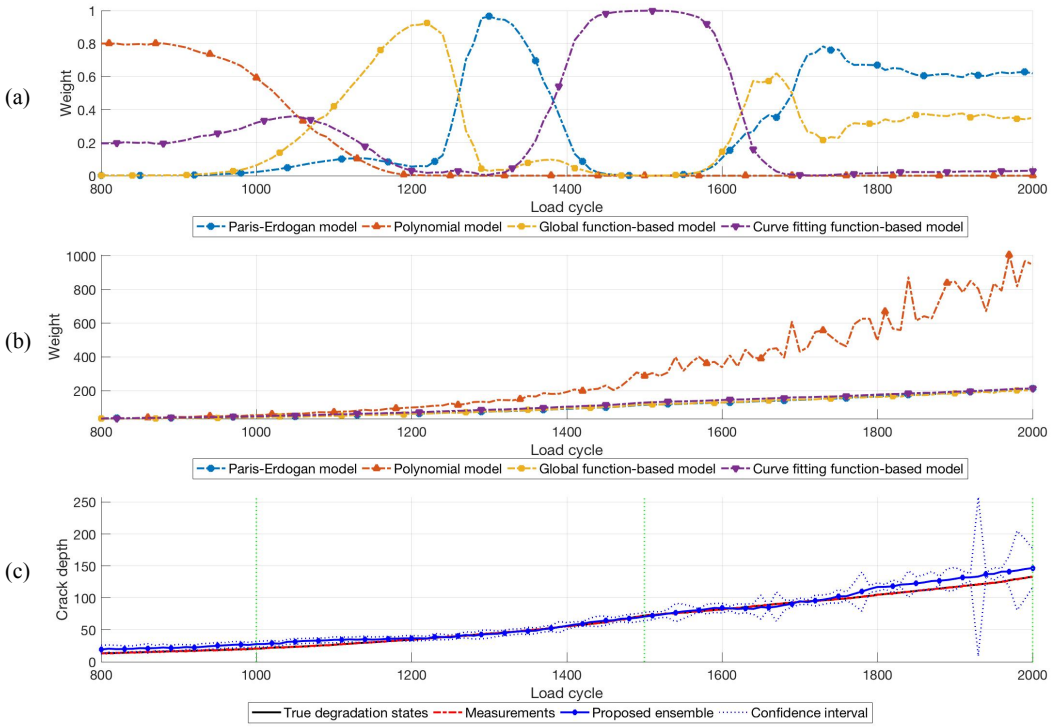


Fig. 9. Degradation state prediction at time t+300 with measurements that are not available until the time t_0 = 500 at load ratio R = 0.15: (a) Ensemble weights of individual models; (b) Degradation state prediction of individual models; (c) Degradation state prediction of the proposed ensemble.

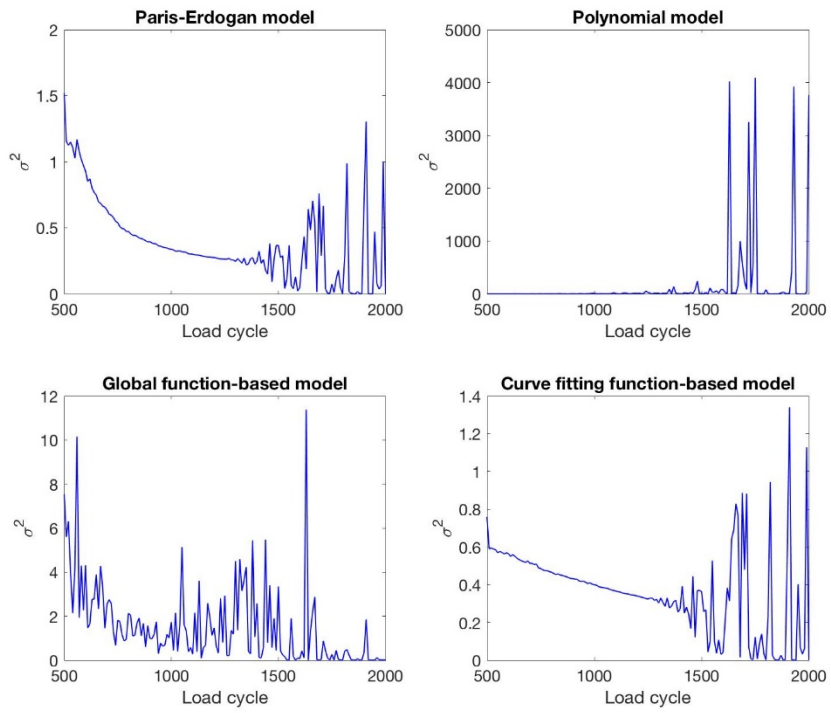


Fig. 10. Variances of degradation state prediction at time t+300 with measurements that are not available until the time t_0 = 500 at load ratio R = 0.15.

4. Conclusions

This paper proposes a prognostic framework for predicting the degradation states of fatigue crack growth under different load conditions. Although several degradation models have been investigated for fatigue crack growth prognosis in the literature, there is no general consensus on a comprehensive crack growth model over different degradation processes. To address this issue, a dynamic-weighted ensemble of multiple degradation models is presented. The key idea of the proposed approach is to utilize a dynamic weight vector, which is updated at each load cycle when the new measurements become available, for evaluating individual degradation models performance with respect to their estimation errors on previous cycles. Short-term predictions of crack growth are also provided to anticipate and proactively prevent sudden breakdowns of the component in a near future. Simulation results show that the proposed prognostic framework can yield a satisfactory performance under evolving operating conditions, and outperforms individual models for fatigue crack growth in terms of prediction accuracy.

The performance of the proposed dynamic-weighted ensemble framework depends also on the specific degradation models used in the ensemble. More advanced degradation models of fatigue crack growth (e.g. those of the FKM Directive [39]) can be accommodated in the framework. Moreover, not only state-dependent but also age-dependent models can be used for predicting the degradation process of fatigue crack growth [40]. Note that the predetermined initial values of the model parameters may

affect the prediction performance: their values should be initialized based on the available measurements of the equipment. To address this issue, research on dynamically initializing the model parameters, e.g. by using the Maximum Likelihood Estimation (MLE) [41], can be considered in future work.

References

[1] Eckmann, S., & Schweizer, C. (2017). Characterization of fatigue crack growth, damage mechanisms and damage evolution of the nickel-based superalloys MAR-M247 CC (HIP) and CM-247 LC under thermomechanical fatigue loading using in situ optical microscopy. *International Journal Of Fatigue*, 99, 235-241.

[2] Kamaya, M. (2017). Fatigue crack tolerance design for stainless steel by crack growth analysis. *Engineering Fracture Mechanics*, 177, 14-32.

[3] Lee, S., Zi, G., Mun, S., Kong, J., & Choi, J. (2015). Probabilistic prognosis of fatigue crack growth for asphalt concretes. *Engineering Fracture Mechanics*, 141, 212-229.

[4] Zhao, Z., Haldar, A., & Breen, F. (1994). Fatigue-Reliability Evaluation of Steel Bridges. *Journal Of Structural Engineering*, 120(5), 1608-1623.

[5] Haile, M., Riddick, J., & Assefa, A. (2016). Robust Particle Filters for Fatigue Crack Growth Estimation in Rotorcraft Structures. *IEEE Transactions On Reliability*, 65(3), 1438-1448.

[6] Haile, M., Chen, T., Shiao, M., & Le, D. (2011). Crack Growth Behavior in Preloaded Metallic Nested-Angle Plates Under Flight Load Spectrum. *Proceedings of the 2011 Annual Conference on Experimental and Applied Mechanics*, 6, 3–11.

[7] Urban, M. (2003). Analysis of the fatigue life of riveted sheet metal helicopter airframe joints. *International Journal Of Fatigue*, 25(9-11), 1013-1026.

[8] Yang, H., Doquet, V., & Zhang, Z. (2017). Fatigue crack growth in two TWIP steels with different stacking fault energies. *International Journal Of Fatigue*, 98, 247-258.

[9] Mayén, J., Abúndez, A., Pereyra, I., Colín, J., Blanco, A., & Serna, S. (2017). Comparative analysis of the fatigue short crack growth on Al 6061-T6 alloy by the exponential crack growth equation and a proposed empirical model. *Engineering Fracture Mechanics*, 177, 203-217.

[10] Yang, G., Pointeau, V., Tevissen, E., & Chagnes, A. (2017). A review on clogging of recirculating steam generators in Pressurized-Water Reactors. *Progress In Nuclear Energy*, 97, 182-196.

[11] Prusek, T., Moleiro, E., Oukacine, F., Adobes, A., Jaeger, M., & Grandotto, M. (2013). Deposit models for tube support plate flow blockage in Steam Generators. *Nuclear Engineering And Design*, 262, 418-428.

[12] Compare, M., Bellani, L., & Zio, E. (2017). Availability Model of a PHM-Equipped Component. *IEEE Transactions On Reliability*, 66(2), 487-501.

[13] Amiri, M., Modarres, M., & Droguett, E. (2015). AE entropy for detection of fatigue crack initiation and growth. *IEEE Conference on Prognostics and Health Management (PHM) 2015*, Austin, TX, 1-8.

[14] Cadini, F., Zio, E., & Avram, D. (2009). Model-based Monte Carlo state estimation for condition-based component replacement. *Reliability Engineering & System Safety*, 94(3), 752-758.

[15] Amiri, M., & Modarres, M. (2014). Short fatigue crack initiation and growth modeling in aluminum 7075-T6. *Proceedings Of The Institution Of Mechanical Engineers, Part C: Journal Of Mechanical Engineering Science*, 229(7), 1206-1214.

[16] Xu, J., Wang, Y. and Xu, L. (2014). PHM-Oriented Integrated Fusion Prognostics for Aircraft Engines Based on Sensor Data. *IEEE Sensors Journal*, 14(4), pp.1124-1132.

[17] Tse, P. and Atherton, D. (1999). Prediction of Machine Deterioration Using Vibration Based Fault Trends and Recurrent Neural Networks. *Journal of Vibration and Acoustics*, 121(3), p.355.

[18] Tobon-Mejia, D., Medjaher, K., Zerhouni, N. and Tripot, G. (2012). A Data-Driven Failure Prognostics Method Based on Mixture of Gaussians Hidden Markov Models. *IEEE Transactions on Reliability*, 61(2), pp.491-503.

[19] Huang, R., Xi, L., Li, X., Richard Liu, C., Qiu, H. and Lee, J. (2007). Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, 21(1), pp.193-207.

[20] Wang, Z., Wang, W., Hu, C., Si, X., & Li, J. (2014). A real-time prognostic method for the drift errors in the inertial navigation system by a nonlinear random-coefficient regression model. *Acta Astronautica*, 103, 45-54.

[21] Gebraeel, N., & Jing Pan. (2008). Prognostic Degradation Models for Computing and Updating Residual Life Distributions in a Time-Varying Environment. *IEEE Transactions On Reliability*, 57(4), 539-550.

[22] Zárate, B., Caicedo, J., Yu, J., & Ziehl, P. (2012). Bayesian model updating and prognosis of fatigue crack growth. *Engineering Structures*, 45, 53-61.

[23] Orchard, M. (2007). A Particle Filtering-based Framework for Online Fault Diagnosis and Failure Prognosis. *Ph.D. Thesis*, Department of Electrical and Computer Engineering, Georgia Institute of Technology.

[24] Bogdanoff, J., & Kozin, F. (1985). *Probabilistic models of cumulative damage*. New York: Wiley.

[25] Sobczyk, K., & Spencer, B. (1992). *Random fatigue*. Boston: Academic Press.

[26] Wu, W. (1986). On the Markov approximation of fatigue crack growth. *Probabilistic Engineering Mechanics*, 1(4), 224-233.

[27] Rocha, M., & Schuëller, G. (1996). A probabilistic criterion for evaluating the goodness of fatigue crack growth models. *Engineering Fracture Mechanics*, 53(5), 707-731.

[28] Yang, J., & Manning, S. (1990). Stochastic crack growth analysis methodologies for metallic structures. *Engineering Fracture Mechanics*, 37(5), 1105-1124.

[29] Yang, J., & Manning, S. (1996). A simple second order approximation for stochastic crack growth analysis. *Engineering Fracture Mechanics*, 53(5), 677-686.

[30] Hossein, S., Mohammad, P., & Saeid, K. (2016). Assessment of stochastic fatigue failures based on deterministic functions. *13th International conference on Probabilistic Safety Assessment and Management (PSAM 13)*, Seoul, Korea.

[31] Wu, W., & Ni, C. (2004). Probabilistic models of fatigue crack propagation and their experimental verification. *Probabilistic Engineering Mechanics*, 19(3), 247-257.

[32] Xing, Y., Ma, E., Tsui, K., & Pecht, M. (2013). An ensemble model for predicting the remaining useful performance of lithium-ion batteries. *Microelectronics Reliability*, 53(6), 811-820.

[33] Su, X., Wang, S., Pecht, M., Zhao, L., & Ye, Z. (2017). Interacting multiple model particle filter for prognostics of lithium-ion batteries. *Microelectronics Reliability*, 70, 59-69.

[34] Paris, P., & Erdogan, F. (1963). A Critical Analysis of Crack Propagation Laws. *Journal Of Basic Engineering*, 85(4), 528.

[35] Irwin, G. (1957). Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate. *Journal of Applied Mechanics*, 24, 361-364.

[36] Myötyri, E., Pulkkinen, U., & Simola, K. (2006). Application of stochastic filtering for lifetime prediction. *Reliability Engineering & System Safety*, 91(2), 200-208.

[37] Ni, C. (2014). Formulation of a Polynomial Stochastic Fatigue Crack Growth Model. *Advanced Materials Research*, 909, 467-471.

[38] Gdoutos, E. (2005). *Fracture Mechanics*. Dordrecht: Springer.

[39] FKM-Guideline. Fracture Mechanics Proof of Strength for Engineering Components. VDMA Publ., 1st Edition (2001, in German), 2nd Edition (2004, in German and English), 3rd Edition (2005, in German).

[40] Li, N., Lei, Y., Guo, L., Yan, T. and Lin, J. (2017). Remaining Useful Life Prediction Based on a General Expression of Stochastic Process Models. *IEEE Transactions on Industrial Electronics*, 64(7), pp.5709-5718.

[41] Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S. and Dybala, J. (2016). A Model-Based Method for Remaining Useful Life Prediction of Machinery. *IEEE Transactions on Reliability*, 65(3), pp.1314-1326.

Paper 2:

Hoang-Phuong Nguyen, Jie Liu, and Enrico Zio. (2019). Ensemble of models for fatigue crack growth prognostics, *IEEE Access, Special Section on Advances in Prognostics and System Health Management*, vol. 7, pp. 49527 - 49537.

Ensemble of models for fatigue crack growth prognostics

Hoang-Phuong Nguyen[a], Jie Liu[b,*], Enrico Zio[c,d,e]

[a] *Chair on System Science and the Energetic Challenge, EDF Foundation, CentraleSupélec, Université Paris-Saclay, 9 rue Joliot-Curie, 91192 Gif-sur- Yvette, France*

[b] *School of Reliability and Systems Engineering, Beihang University, 37 Xueyuan Road, Haidian, Beijing, China*

[c] *Dipartimento di Energia, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy*

[d] *ParisTech/PSL Université Paris, Centre de Recherche sur les Risques et les Crises (CRC), 06904 Sophia Antipolis, France*

[e] *Department of Nuclear Engineering, Kyung Hee University, Seoul 02447, South Korea*

Abstract

Various models of fatigue crack growth in different scenarios have been proposed in the literature. Here, in this paper, we propose a general prognostic framework for tracking crack evolution in equipment undergoing fatigue and predicting the Remaining Useful Life (RUL). The main contribution of this work is to integrate Particle Filtering (PF) and a new ensemble model which combines diverse physical degradation models with respect to their accuracy performance in previous time steps, in order to maximize the overall prediction capability. To validate the effectiveness of the proposed framework, a case study concerning multiple fatigue crack growth degradations is extensively investigated.

Keywords: Prognostics and Health Management; Fatigue crack growth; Dynamic ensemble; Multiple stochastic degradation; Particle Filter; Remaining Useful Life;

Nomenclature

Abbreviations

BWWV	Best-Worst Weighted Vote
EOP	End-Of-Process
IMMPF	Interacting Multiple Model Particle Filter
MAPE	Mean Absolute Percentage Error
MC	Monte Carlo
MSE	Mean Square Error
PBM	Physics-Based Model
PDF	Probability Density Function
PF	Particle Filtering
PPI	Prognostic Performance Indicator

RMSE	Root Mean Square Error
RUL	Remaining Useful Life
SMC	Sequential Monte Carlo
SME	Sample Mean Error
SMeE	Sample Median Error
TWEB	Timeliness Weighted Error Bias

Symbols

a	constant of polynomial crack growth model
b	constant of curve fitting model
C	material constant
d	width of the specimen undergoing fatigue crack (mm)
f	state transition function
g	measurement function
$h(x)$	geometric factor
m	material constant
N	number of fatigue load cycles (cycle)
N_M	number of degradation models
N_P	number of particles
N_S	number of units under test
p	probability distribution
q	importance sampling distribution
RUL_t	actual RUL at time t (cycle)
$\widehat{RUL}_t^i$	estimated RUL of the ith degradation model at time t (cycle)
$\widehat{RUL}_t$	estimated RUL of the ensemble at time t (cycle)
t	time (cycle)
T_t^i	estimated failure time of the ith degradation model at time t (cycle)
$w_{est}^{i,t}$	previous estimation accuracy-based output weight of the ith degradation model in the ensemble at time t
$w_{pre}^{i,t}$	previous prediction accuracy-based output weight of the ith degradation model in the ensemble at time t
$w_{overall}^{i,t}$	overall output weight of the ith degradation model in the ensemble at time t
$\widetilde{w}_{overall}^{i,t}$	normalized overall output weight of the ith degradation model in the ensemble at time t
x	degradation state (mm)

x_{th}	failure threshold (mm)
$\hat{x}_t$	estimated degradation state of the ensemble at time *t* (mm)
$\hat{x}_t^i$	estimated degradation state of the *i*th degradation model at time *t* (mm)
$\tilde{x}_{t_p:t}^i$	predicted degradation state of the *i*th degradation model at time *t* with measurements that are available up to time t_p ($t_p < t$) (mm)
z	measurement (mm)
α	geometric coefficient of fatigue crack
δ_{est}	time horizon for previous estimates considered (cycles)
δ_{pre}	time horizon for previous predictions considered (cycles)
ΔK	stress intensity factor (MPa√m)
$\Delta\sigma$	cyclic stress amplitude (MPa)
Δt	time interval (cycle)
ε	weight coefficient of individual degradation model
υ	measurement noise
ω	state noise

1. Introduction

The rapid development of technology and computer science is bringing opportunities for industrial systems to evolve smarter and faster, but also more complex. In this fast-changing environment, unanticipated risks and failures which may cause large-scale breakdowns with significant losses in both production and economics, have also increased [1]. To cope with this challenging situation, the development of reliability and health management strategies for preventing components and systems from such unexpected failures are urgently required. Specifically, these strategies aim to monitor health conditions of engineering components, predict their Remaining Useful Lives (RULs) and, ultimately, enable optimal maintenance decisions before the breakdown of the components [2], [3]. In practice, the reliability of equipment usually starts decreasing due to gradual degradation, e.g., delamination [4], fatigue crack [5]–[8], corrosion [9], [10], etc., under periodic cyclic loads and eventually leading to failures. Fatigue crack growth is one of the most frequent degradations leading to components and systems failures in several major industries, including energy [6], [11], automotive [7], aerospace [8], etc. Therefore, the demand of prognostic systems for dealing with fatigue crack growth has recently increased.

To address this issue, Physics-Based Models (PBMs), which utilize the physical knowledge of the degradation for constructing a quantitative analytical model of the equipment behavior, have gained significant attention for fatigue crack growth prognostics [12]–[14]. In [13], a failure prognostic scheme for fatigue crack growth prediction was introduced, which employed a stochastic crack growth model

and a Bayesian technique to timely update the equipment degradation state from a sequence of monitored measurements. Other Bayesian-based prognostic approach was presented to estimate the stress intensive range of the degradation model in an online manner [14]. The capability of Bayes theorem was fully exploited for updating knowledge about the current degradation state of the target equipment and the unknown parameters in physical models, when a new measurement becomes available.

Among Bayesian-based prognostic techniques, a sequential Monte Carlo (SMC) method, known as Particle Filtering (PF) method, has become very popular due to its capability of effectively handling non-linear systems and non-Gaussian noises. The key idea behind this method is to represent the posterior distribution of the equipment state by a random set of weighted samples, also called *particles*, and then compute the estimated state based on the particles and their associated weights. This methodology has been widely adopted for state estimation and prediction of crack growth [15]–[17], Lithium-ion batteries [18], [19], PEM fuel cells [20], bearings [21], etc.

On the other hand, the performance of model-based prognostic frameworks for fatigue crack growth largely depends on the choice of the adopted physics-of-failure model [22], [23]. Numerous research on modelling fatigue crack growth have been extensively investigated and developed [5], [24]–[26]. In [24], a comprehensive comparison of stochastic models for fatigue crack growth, including the Markov chain model, the Yang's power law-based model, and a polynomial model, was carried out. The results indicated that each degradation model has its own specific range of applicability, that is, each model is only appropriate to certain degradation processes under certain conditions. To the best knowledge of the authors, there is no general consensus on a prognostic model for fatigue crack growth under different degradation processes. Recently, hybrid and multi-degradation model ensembles have attracted the attention of industrial practitioners and researchers due to their superiority over individual degradation models in terms of higher accuracy and better generalization capability [19], [27]. The fundamental idea of these empirical frameworks is to exploit the diversity of different degradation models, which can offer complementary information about the degradation states to be estimated. In an application of Lithium-ion battery prognostics, an Interacting Multiple Model Particle Filter (IMMPF) has been presented to combine the estimations from three different battery capacity degradation models [27]. The results experimentally indicated that the ensemble approach can yield a promising performance in terms of smaller estimation errors and more accurate predictions than single models.

In this paper, an ensemble-based prognostic approach is presented for predicting the evolution to failure and the RUL of an equipment undergoing fatigue crack growth. To maximize the diversity property of the proposed framework, four stochastic degradation models of fatigue crack growth are considered in this work. Moreover, PF is used to track the crack propagation process with nonlinear and non-Gaussian characteristics and eventually to predict the RUL of the equipment before breakdowns. To further enhance the performance of the proposed framework, a dynamic weighted ensemble strategy is proposed in this paper, based on the previous accuracy performance in degradation

state estimation and RUL prediction of each single model in the ensemble. Finally, a set of prognostic performance indicators (PPIs) is employed to validate the prediction capability of the proposed framework.

The rest of this paper is organized as follows. Section 2 introduces the degradation models for fatigue crack and details the proposed prognostic framework. Section 3 describes the illustrative case study and the experimental results of the proposed framework in comparison with individual models are shown in Section 4. Finally, Section 5 concludes the study.

2. Ensemble-based framework for fatigue crack prognostics

This section presents the proposed ensemble-based framework for fatigue crack prognostics. Three key issues are addressed: how to select the degradation models for the ensemble; how to use the degradation models for estimating the degradation states and predicting the RUL of the equipment; how to combine the outputs of the individual models for achieving maximum accuracy. Fig. 1 illustrates the flowchart of the proposed prognostic model; more details are given in the following sections.

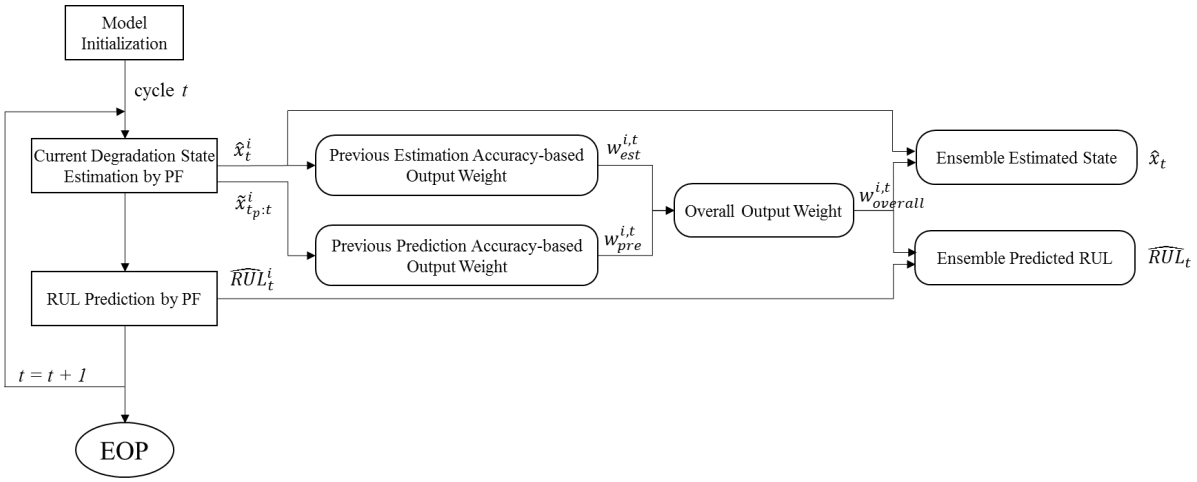


Fig. 1. Flow diagram of the proposed prognostic framework.

2.1. Degradation models for fatigue crack

Diversity is an important aspect to consider in the design of an ensemble modeling framework. To address this issue, four stochastic fatigue crack degradation models are selected for exploiting their diversity in the ensemble: Paris-Erdogan, polynomial, global function-based, and curve fitting models.

2.1.1. Paris-Erdogan model

The popular Paris-Erdogan model describes the dynamic evolution of the crack depth x as a function of the load cycle number N as follows [28]:

$$\frac{dx}{dN} = C(\Delta K)^m \tag{1}$$

where C and m are constants related to the material properties, and ΔK is the Irwin's stress intensity factor defined by [29]:

$$\Delta K = \Delta\sigma\sqrt{\pi x} \tag{2}$$

where $\Delta\sigma$ is the cyclic stress amplitude. In practice, the statistical variability of the crack growth rate can be addressed by modifying Eq. (1) with an intrinsic process stochasticity [30]:

$$\frac{dx}{dN} = e^{\omega} C(\Delta K)^m \tag{3}$$

where $\omega \sim N(0,\sigma_\omega^2)$ is a white Gaussian noise. For a sufficiently small Δt, the Markov chain state-space model of the degradation state x in Eq. (3) can be discretized as follows:

$$x_t = x_{t-1} + e^{\omega} C(\Delta K)^m \Delta t \tag{4}$$

2.1.2. Polynomial model

The polynomial models were first introduced for fatigue crack growth in order to solve the mismatch between the traditional power function-based models, i.e. Paris-Erdogan, and the practical median crack growth curves [24], [31]:

$$\frac{dx}{dN} = e^{\omega}(a_0 + a_1 x + a_2 x^2) \tag{5}$$

where $a_i, i = 0,...,2$ are the second-degree polynomial parameters. Indeed, various works showed that the polynomial models are able to yield the best fit of the linear stage of a degradation process, compared to conventional models [19], [31]. Specifically, the Markov process representation for a polynomial crack growth model can be given as follows:

$$x_t = x_{t-1} + e^{\omega}(a_0 + a_1 x + a_2 x^2)\Delta t \tag{6}$$

2.1.3. Global function

Considering again the Paris-Erdogan model Eq. (4) and the fact that fatigue crack growth generally depends not only on material properties but also on equipment geometry, a so-called global function was introduced by reformulating the stress intensity factor [32]:

$$\Delta K = h(x)\Delta\sigma\sqrt{\pi x} \tag{7}$$

where $h(x)$ denotes the geometric factor of fatigue crack, defined by:

$$h(x) = \alpha_0 + \alpha_1 \frac{x}{d} + \alpha_2 \left(\frac{x}{d}\right)^2 + \alpha_3 \left(\frac{x}{d}\right)^3 \tag{8}$$

where $\alpha_i, i = 0,...,3$ and d are geometric coefficients and the width of the specimen, respectively. The global function-based model for fatigue crack growth can be, then, written as follows:

$$x_t = x_{t-1} + e^{\omega} C(h(x)\Delta\sigma\sqrt{\pi x})^m \Delta t \tag{9}$$

2.1.4. Curve fitting function

In [32], an empirical crack growth model based on a curve fitting function was presented, which was shown to outperform the conventional models, such as Paris-Erdogan and polynomial models, in terms of higher prediction accuracy and lower computational cost:

$$\frac{dx}{dN} = e^{\omega}\left(\frac{1}{b_1 x^m + b_2}\right) \tag{10}$$

where b_1, b_2 are model constants. The discretized Markov process representation for the model can be given as follows:

$$x_t = x_{t-1} + e^{\omega}\left(\frac{1}{b_1 x^m + b_2}\right)(\Delta K)^m \Delta t \tag{11}$$

2.2. Degradation state estimation and RUL prediction by PF

In this work, PF is employed to estimate the current degradation state of the equipment and to predict its future evolution until failure. The key idea of PF is based on Bayesian filtering and Monte Carlo (MC) simulation [33]. The basics of the method are recalled in the following sections.

2.2.1. Current degradation state estimation

PF assumes that the state model can be represented as a first-order Markov process, where the current degradation state x_t at time t depends only on its previous state x_{t-1}. The dynamic system process can be described by the following equations:

$$x_t = f_t(x_{t-1}, \omega_{t-1}) \tag{12}$$

$$z_t = g_t(x_t, v_t) \tag{13}$$

where z_t denotes the measurement, ω_t is the state noise sequence, and v_t is the measurement noise sequence at the inspection time $t\ \{t \in \mathrm{N}\}$.

In a Bayesian framework, the system state x_t can be estimated by constructing its posterior probability density function (pdf), $p(x_t \mid z_{1:t})$, via two consecutive steps, namely prediction and update. In the prediction step, the previous state estimation x_{t-1} and the state transition model f_t are utilized

to obtain the prior distribution of the system state x_t at current time t via the Chapman-Kolmogorov equation:

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}, z_{1:t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1}$$

$$= \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \tag{14}$$

where $p(x_t \mid x_{t-1})$ is the conditional probability distribution and is defined by the state model in Eq. (12). As a new measurement z_t is collected, the required *posterior* distribution of the current state x_t can, then, be obtained by *updating* the prior distribution via Bayes theorem as follows:

$$p(x_t \mid z_{1:t}) = \frac{p(x_t \mid z_{1:t-1}) p(z_t \mid x_t)}{p(z_t \mid z_{1:t-1})} \tag{15}$$

where $p(z_t \mid x_t)$ is the likelihood function defined by the measurement model in Eq. (13) and $p(z_t \mid z_{1:t-1})$ is a normalizing constant given by:

$$p(z_t \mid z_{1:t-1}) = \int p(x_t \mid z_{1:t-1}) p(z_t \mid x_t) dx_t \tag{16}$$

However, there is usually no analytical solution to Eqs. (14) and (15) [19]. To address this issue, PF utilizes MC simulation to approximate the true probability distribution with a set of weighted random particles $\{x_t^i, w_t^i, i = 1,.., N_P\}$, where N_P is the total number of particles. In fact, these particles evolve statistically independently of each other, according to the probabilistic state model Eq. (12). In this regard, the posterior distribution at time t can be approximated as:

$$p(x_t \mid z_{1:t}) \approx \sum_{i=1}^{N_P} w_t^i \delta\left(x_t - x_t^i\right) \tag{17}$$

where $\delta(.)$ is the Dirac Delta function, often used to represent a discrete distribution as a continuous probability density function $p(x)$:

$$p(x) = \sum_{i=1}^{n} p_i \delta\left(x_t - x_i\right) \tag{18}$$

where $x = \{x_1, ..., x_n\}$ is a discrete distribution with corresponding probabilities $\{p_1, ..., p_n\}$. In particular, the particle x_t^i is sampled from the importance sampling distribution $q(x_t \mid z_{1:t})$ and its associated weight w_t^i is given by:

$$w_t^i = \frac{p(z_{1:t} \mid x_t^i) p(x_t^i)}{q(x_t^i \mid z_{1:t})} \tag{19}$$

By setting $q(x_t \mid z_{1:t}) = p(x_t \mid x_{t-1})$ defined in Eq. (12), the particle weight w_t^i can be updated with a new collected measurement z_t as follows:

$$w_t^i = w_{t-1}^i p(z_t \mid x_t^i) \tag{20}$$

where $p(z_t \mid x_t^i)$ is the likelihood of measurement z_t given the particle x_t^i. Note that the weights are normalized as $\sum_i w_t^i = 1$.

2.2.2. Future degradation evolution prediction

Once the posterior distribution $p(x_t \mid z_{1:t})$ of the current degradation state is estimated, it is possible to predict the future degradation evolution and the RUL of the equipment. However, note that there is no available information for estimating the likelihoods of the future degradation states, because future measurements $z_{t+l}, l = 1,...,T-t$, where T is the time horizon of interest for the analysis, have not been collected yet. The only available information is the dynamic state model Eq. (12). Then, the *l*-step ahead posterior distribution $p(x_{t+l} \mid z_{1:t})$ can be written as follows:

$$p(x_{t+l} \mid z_{1:t}) = \int \dots \int \prod_{j=t+1}^{t+l} p(x_j \mid x_{j-1}) p(x_t \mid z_{1:t}) \prod_{j=t}^{t+l-1} dx_j \tag{21}$$

The numerical evaluation of the integrals in Eq. (21) requires significant computational effort. In this paper, an approach presented in [34] is adopted with the assumption that the particle weights do not change from time t to time $t+l$, i.e., $w_t^i = w_{t+1}^i = ... = w_{t+l}^i$. Accordingly, the predicted distribution at time $t+l$ is given by:

$$p(x_{t+l} \mid z_{1:t}) \approx \sum_{i=1}^{N_P} w_t^i \delta(x_{t+l} - x_{t+l}^i) \tag{22}$$

where the particle x_{t+l}^i is obtained by iteratively applying the state model Eq. (12) to the corresponding particle of the current state x_t^i.

Finally, the RUL associated to each particle at the present time t can be calculated with reference to the earliest time that the degradation state exceeds the failure threshold x_{th}:

$$\widehat{RUL}_t^i = \left\{ (T_t^i - 1 - t) \middle| g(x_{T_t^i - 1}, p_t^i, v_t) < x_{th}, g(x_{T_t^i}, p_t^i, v_t) \geq x_{th} \right\} \tag{23}$$

where T_t^i is obtained by simulating the particle evolution via the state model Eq. (12). The predicted RUL distribution is, then, given by:

$$p(RUL \mid z_{1:t}, x_i < x_{th}) \approx \sum_{i=1}^{N_P} w_t^i \delta(R\hat{U}L_t - R\hat{U}L_t^i) \tag{24}$$

More details can be found in [35], [36].

2.3. Selective ensemble based on previous estimation and prediction accuracies

With respect to the way of calculating the weights of the models in an ensemble, the existing ensemble methods can generally be divided into three categories: (a) simple vote ensemble [37], where all individual models outputs are given the same weight coefficients in the voting strategy; in this scheme, majority vote is the most popularly used rule; (b) weighted ensemble [27], which combines individual models with different weight coefficients: each individual is assumed to have a different contribution to the performance of the ensemble model; (c) selective ensemble [38], which includes only an optimal subset of models. This latter method has recently attracted increasing interest, due to its capability of significantly reducing the bias and variance in the ensemble estimation [38].

In this section, we present a selective ensemble approach for prognostics of fatigue crack growth based on a best-worst weighted vote (BWWV) strategy [39]. A novel ensemble weight constructed by using both previous estimation and prediction accuracies of each individual model in the population is proposed.

2.3.1. Previous estimation accuracy based output weight calculation

Suppose that we have a sequence of measurements collected until the current time t, $\{z_j, j = 1,...,t\}$. The degradation states described by the individual models, $\{\hat{x}_j^i, i = 1,...,N_M, j = 1,...,t\}$, where N_M is the number of individual models in the population ($N_M = 4$ in this study), can be estimated by using the PF described in Section 2.2. A weight coefficient of the *i*th model, based on the Root Mean Square Error (RMSE) of its previous estimates with respect to the corresponding measurements, can be calculated as follows:

$$\varepsilon_t^i = \sqrt{\frac{1}{\delta_{est}} \sum_{k=t-\delta_{est}}^{t} (z_k - \hat{x}_k^i)^2} \tag{25}$$

where δ_{est} is the time horizon of previous estimates considered ($\delta_{est} = 50$ load cycles in the case study that follows). The previous estimation accuracy-based output weight of each single model is, then, obtained based on the BWWV as follows:

$$w_{est}^{i,t} = 1 - \frac{\varepsilon_t^i - \varepsilon_t^{\min}}{\varepsilon_t^{\max} - \varepsilon_t^{\min}} \tag{26}$$

where $\varepsilon_t^{\min} = \min_i \{\varepsilon_t^i\}$ and $\varepsilon_t^{\max} = \max_i \{\varepsilon_t^i\}$. By using the BWWV strategy, a maximum weight, $w_{est}^t = 1$, is assigned to the model in the ensemble with highest accuracy at the present time *t*, and a null weight, $w_{est}^t = 0$, is given to the model with least accuracy, which is equivalent to removing the model from the ensemble for the estimation at time *t*.

2.3.2. Previous prediction accuracy-based output weight calculation

Due to the fact that there is no available information from observations to predict the future equipment RUL, the prediction accuracy of each model in the ensemble for the previous time steps is used to calculate the corresponding output weight. We first identify a time instant t_p before the present time t in the time horizon, where $t = t_p + \delta_{pre}$ ($\delta_{pre} = 100$ load cycles in the following case study), as illustrated in Fig. 2. The state prediction $\tilde{x}_{t_p:t}$ (the dashed line) of one model at time t_p is obtained by iteratively applying the system model to the estimated state $\tilde{x}_{t_p}$, which is set to z_{t_p} in this study. We can now calculate the weight coefficient of the *i*th model, based on the RMSE of its predictions for degradation states between time t_p and t, with respect to the measurements:

$$\varepsilon_t^i = \sqrt{\frac{1}{\delta_{pre}} \sum_{k=t_p}^{t} (z_k - \tilde{x}_k^i)^2} \tag{27}$$

Subsequently, the previous prediction accuracy-based output weight of each single model is computed as:

$$w_{pre}^{i,t} = 1 - \frac{\varepsilon_t^i - \varepsilon_t^{\min}}{\varepsilon_t^{\max} - \varepsilon_t^{\min}} \tag{28}$$

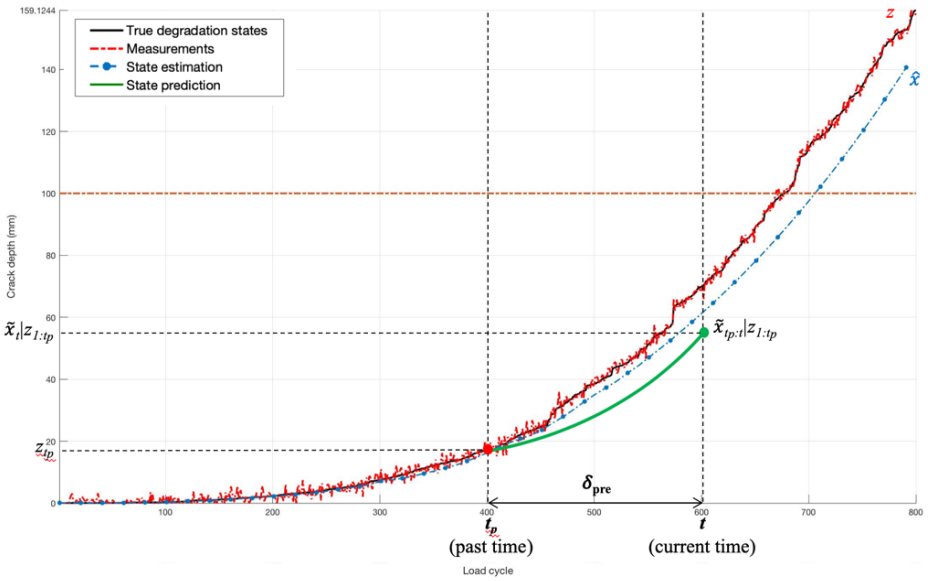


Fig. 2. Sketch of the previous prediction accuracy-based output weight calculation approach.

2.3.3. Output weight calculation

Finally, the complete output weight of the *i*th model in the ensemble at time t is calculated as an average of the previous estimation accuracy-based and the previous prediction accuracy-based weights:

$$w_{overall}^{i,t} = \frac{w_{est}^{i,t} + w_{pre}^{i,t}}{2} \tag{29}$$

The output weight is, then, normalized as:

$$\tilde{w}_{overall}^{i,t} = \frac{w_{overall}^{i,t}}{\sum_{i}^{N_M} w_{overall}^{i,t}} \tag{30}$$

Once the output weights for all models are updated, a weighted-sum strategy is used to obtain the degradation state estimation and the RUL prediction of the ensemble as follows:

$$\hat{x}_t = \sum_{i=1}^{N_M} \hat{x}_t^i \times \tilde{w}_{overall}^{i,t} \tag{31}$$

$$\widehat{RUL}_t = \sum_{i=1}^{N_M} \widehat{RUL}_t^i \times \tilde{w}_{overall}^{i,t} \tag{32}$$

where $\hat{x}_t$ and $\widehat{RUL}_t$ are the degradation state estimation and the RUL prediction of the proposed ensemble at the present time t, respectively; $\widehat{RUL}_t^i$ is the RUL prediction of the *i*th model in the ensemble.

3. Case study

A case study of fatigue crack growth is carried out in this work to demonstrate the effectiveness of the proposed method, including crack depth measurements of 100 simulated degradation trajectories, as shown in Fig. 3. The common Paris-Erdogan model in Eq. (4) is adopted for describing the evolution of the crack depth with the parameters predefined as follows:

- The model constants are $C = 0.1$ and $m = 1.3$;
- The state and measurement noise variances are $\sigma_\omega^2 = 1.10$ and $\sigma_v^2 = 2.25$, respectively.

The crack depths, with a 10^{-4} mm initial length, are recorded every load cycle. The failure threshold is $x_{th} = 100$ mm. And the fatigue simulation for each degradation trajectory is performed with a total 800 load cycles.

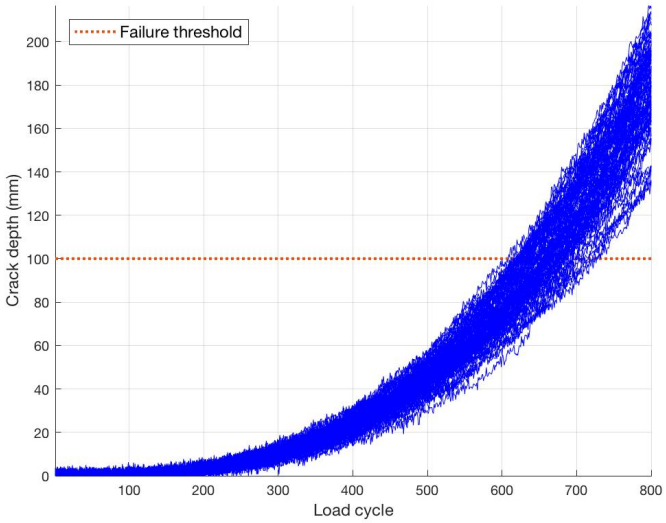


Fig. 3. 100 fatigue crack growth degradation trajectories.

4. Performance evaluation

In this section, the robustness of the proposed ensemble-based prognostic framework is exploited for tracking a fatigue crack growth trajectory and, then, predicting the equipment RUL. The results are compared with four models of fatigue crack growth to validate the improved performance in terms of degradation state estimation and RUL prediction. To evaluate the prognostic framework, five widely used PPIs are considered: a) Timeliness Weighted Error Bias (TWEB); b) Sample Mean Error (SME); c) Mean Absolute Percentage Error (MAPE); d) Mean Square Error (MSE); e) Sample Median Error (SMeE). Details of their definitions are given in Appendix.

When a new measurement is collected, the estimation of the current degradation state for each individual model is also timely updated by using PF as described in Section 2.2. Fig. 4 illustrates the estimation results of four single models over the lifetime of the considered degradation trajectory. The first degradation trajectory from the simulated crack depth dataset described in Section 3 is taken. Each model shows a distinctive characteristic in different stages of the degradation evolution of the fatigue crack, which is perfectly suitable for diversity in the proposed ensemble.

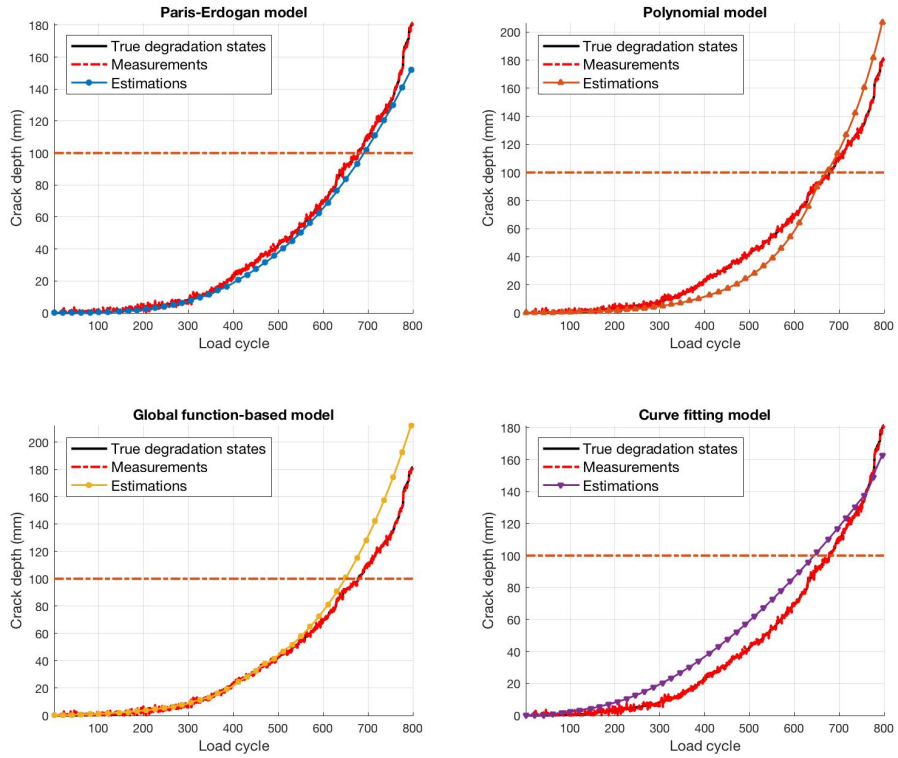


Fig. 4. Degradation state estimation for the considered degradation trajectory using individual models.

Based on the estimations of the individual models, the output weights can be determined and used to update the results of the state estimation and RUL prediction by the proposed ensemble, as shown in Figs. 5 and 6, respectively. As can be seen in Figs. 5 and 6, the individual fatigue crack growth models do not perform very well in the RUL prediction throughout the time horizon considered because of their low accuracy in estimating the current degradation state. In contrast, the proposed approach has a performance which is superior to any individual model throughout the entire life of the equipment, yielding a RUL prediction close to the true RUL.

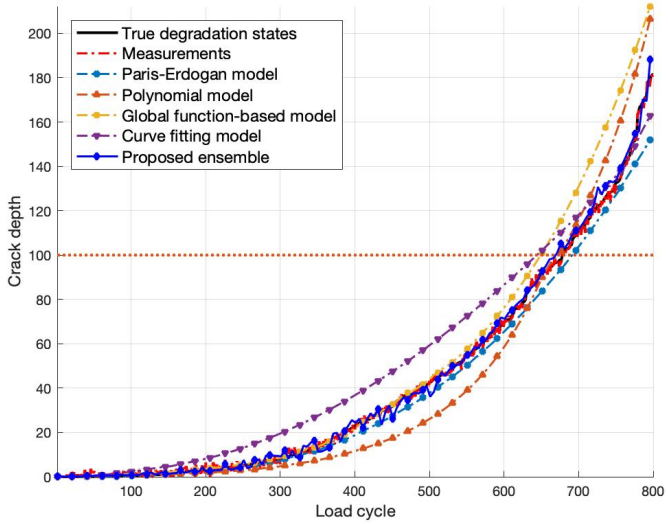


Fig. 5. Degradation state estimation for the considered degradation trajectory using the proposed ensemble.

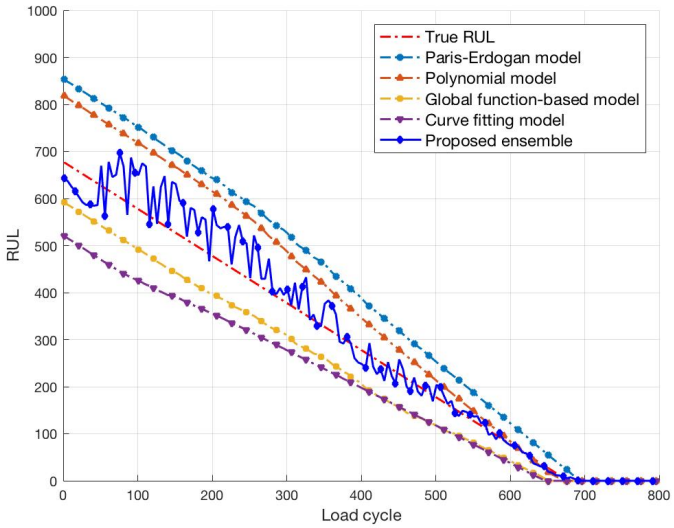


Fig. 6. RUL prediction for the considered degradation trajectory using the proposed ensemble.

To further investigate the performance of the proposed method, four different randomly chosen scenarios are considered, whose results are depicted in Figs. 7 and 8. As shown in these figures, the proposed ensemble method consistently exhibits satisfactory performance in estimating the equipment crack growth trend and accurately predicting the RUL. This is due to the proposed prognostic approach which benefits from the diverse accuracy of the individual models by a weighting scheme that can adaptively select the best set of models. Furthermore, in Fig. 8, the confidence intervals show that the RUL prediction accuracy of the proposed method is improved with more available data.

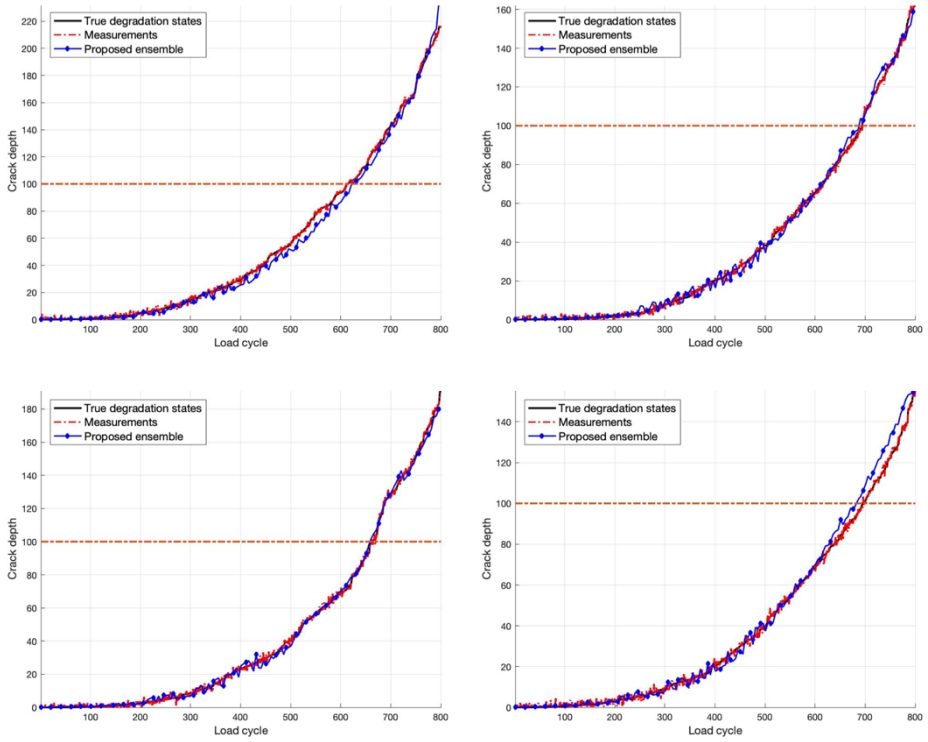


Fig. 7. Degradation state estimation using the proposed ensemble with different available measurements.

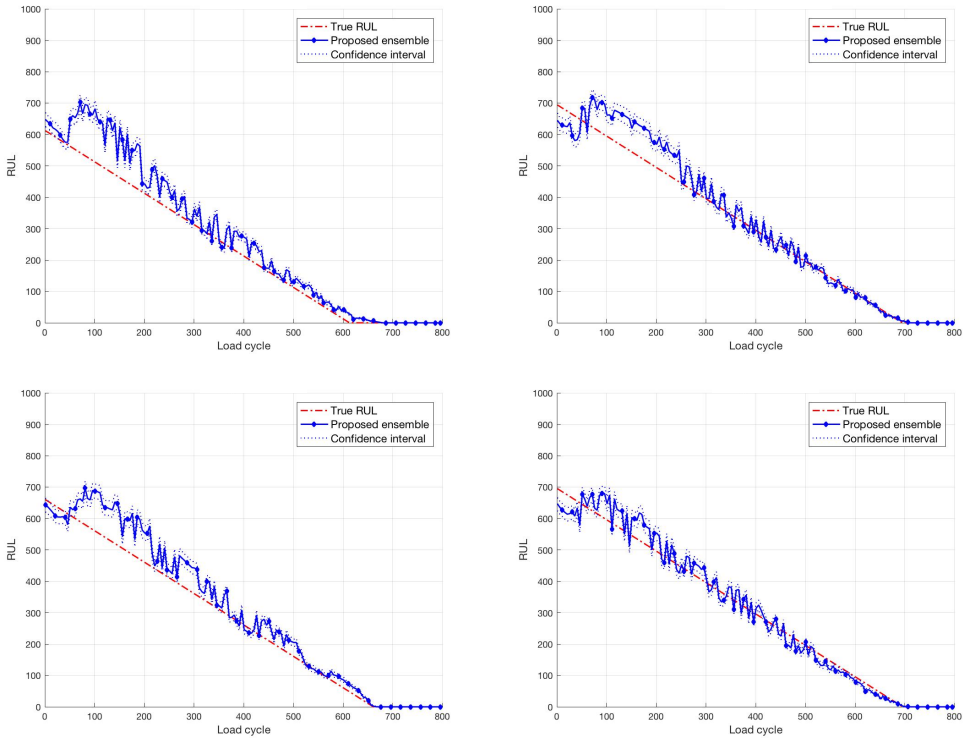


Fig. 8. RUL prediction using the proposed ensemble with different available measurements.

Tables 1 and 2 present the average performances in terms of degradation state estimation and RUL

prediction, which have been calculated based on 100 crack depth growth scenarios. The results clearly show that the proposed prognostic approach consistently outperforms the individual models for all of the prognostic metrics.

Table 1
Performance comparison in terms of MSE of degradation state estimations.

	Paris-Erdogan	Polynomial	Global function	Curve fitting	Proposed ensemble
Avg. MSE (std)	117.72 (102.68)	166.30 (80.39)	138.64 (74.91)	102.90 (69.38)	8.85 (5.04)

Table 2
Performance comparison in terms of PPIs of RUL predictions.

	TWEB	SME	MAPE	MSE	SMeE
Paris-Erdogan	0.09	115.25	0.62	18.28×10^3	114.63
Polynomial	0.07	85.68	0.37	11.56×10^3	85.43
Global function	0.02	45.79	0.20	3.11×10^3	45.86
Curve fitting	0.03	65.18	0.23	7.01×10^3	64.18
Proposed ensemble	0.01	29.41	0.16	3.03×10^3	31.81

5. Conclusions

In this paper, a prognostic modelling framework for fatigue crack growth is proposed. The main original contribution of the work is to combine the PF and a new adaptive ensemble approach, which integrates models of diverse accuracies in previous estimations and predictions for maximizing the generalized prediction performance. The proposed framework is, then, applied to track the degradation evolution and predict the equipment RUL. Various prognostic metrics are employed to evaluate the prediction performance. The results indicate that the proposed ensemble-based prognostic framework outperforms conventional models and is a powerful tool for prognostics of fatigue crack growth.

A limitation of the study is the lack of a real application for validation. Even though several simulation tests were performed to prove the effectiveness of the proposed approach in terms of different PPIs, a real case study of fatigue crack growth is still needed. Further research on addressing this issue with practical applications of fatigue crack can be considered in future work.

Appendix

Detailed definitions of the PPIs

Formula	Description

6. Timeliness weighted error bias (TWEB) $TWEB = \frac{1}{N_S}\sum_{j=1}^{N_S}\phi\left(\sum_{t=1}^{T_i}\gamma_{j,t}\frac{\hat{RUL}_{j,t}-RUL_{j,t}}{T_j}\right)$ $\phi(y)=\begin{cases}\exp\left(\frac{\lvert y\rvert}{e_1}\right)-1, & for\ y<0\\ \exp\left(\frac{\lvert y\rvert}{e_2}\right)-1, & for\ y\geq 0\end{cases}$ $e_1>e_2>0$	Measure the weighted prediction error over the lifetime T_j by using a penalty function $\phi(y)$ and a weighting function $\gamma_{j,t}$. $\gamma_{j,t}$ is defined as a Gaussian kernel function with a mean value T_j and a standard deviation $0.5T_j$. The optimal value for TWEB is 0, which indicates that the predicted RUL is centered on the true one. Higher values of TWEB indicate a great discrepancy between the predicted RUL and the true one.
7. Sample mean error (SME) $SME = \left\lvert\frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{T_j}\sum_{t=1}^{T_j}\left(\hat{RUL}_{j,t}-RUL_{j,t}\right)\right\rvert$	Calculate the average errors of all sample points during the lifetime T_j. The optimal value for SME is 0, which indicates that the average errors of all samples is 0, that is, the predicted RUL is centered on the true one. Higher values of SME indicate a great discrepancy between the predicted RUL and the true one.
8. Mean absolute percentage error (MAPE) $MAPE = \frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{T_j}\sum_{t=1}^{T_j}\left\lvert\frac{\hat{RUL}_{j,t}-RUL_{j,t}}{RUL_{j,t}}\right\rvert$	Measure the average absolute percentage error of all samples throughout the lifetime T_j. The optimal value for MAPE is 0, which indicates a negligible error for all samples during their lifetime. Higher values of MAPE indicate a great discrepancy between the predicted RUL and the true one.
9. Mean square error (MSE) $MSE = \frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{T_j}\sum_{t=1}^{T_j}\left(\hat{RUL}_{j,t}-RUL_{j,t}\right)^2$	Take into account the average quadratic error of the predicted RUL of all samples during the lifetime T_j. The optimal value for MSE is 0, which indicates that the predicted RUL is equal to the true one for all samples. Higher values of MSE indicate high errors in the predicted RUL.
10. Sample median error (SMeE) $SME = \left\lvert\underset{j=1,\ldots,N_S}{median}\left(\frac{1}{T_j}\sum_{t=1}^{T_j}\left(\hat{RUL}_{j,t}-RUL_{j,t}\right)\right)\right\rvert$	Exploit the absolute median of average errors of all samples over the lifetime T_j. The optimal value for SMeE is 0, which indicates that the median error of all samples is zero. Higher values of SMeE indicate that most predicted RULs are wrong.

References

[1] E. Zio, "Challenges in the vulnerability and risk analysis of critical infrastructures," *Reliab. Eng. Syst. Saf.*, vol. 152, pp. 137–150, 2016.

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

[3] E. Zio and M. Compare, "Evaluating maintenance policies by quantitative modeling and analysis," *Reliab. Eng. Syst. Saf.*, vol. 109, pp. 53–65, Jan. 2013.

[4] H. Nykyforchyn, O. Zvirko, O. Tsyrulnyk, and N. Kret, "Analysis and mechanical properties characterization of operated gas main elbow with hydrogen assisted large-scale delamination," *Engineering Failure Analysis*, 2017.

[5] J. N. Yang and S. D. Manning, "Stochastic crack growth analysis methodologies for metallic structures," *Eng. Fract. Mech.*, vol. 37, no. 5, pp. 1105–1124, Jan. 1990.

[6] T. Prusek, E. Moleiro, F. Oukacine, A. Adobes, M. Jaeger, and M. Grandotto, "Deposit models for tube support plate flow blockage in Steam Generators," *Nucl. Eng. Des.*, vol. 262, pp. 418–428, Sep. 2013.

[7] H. K. Yang, V. Doquet, and Z. F. Zhang, "Fatigue crack growth in two TWIP steels with different stacking fault energies," *Int. J. Fatigue*, vol. 98, pp. 247–258, 2017.

[8] M. A. Haile, J. C. Riddick, and A. H. Assefa, "Robust Particle Filters for Fatigue Crack Growth Estimation in Rotorcraft Structures," *IEEE Transactions on Reliability*, vol. 65, no. 3. pp. 1438–1448, 2016.

[9] F. Li, Y. Qu, and J. Wang, "Bond life degradation of steel strand and concrete under combined corrosion and fatigue," *Eng. Fail. Anal.*, vol. 80, pp. 186–196, 2017.

[10] A. Vazdirvanidis, G. Pantazopoulos, and A. Rikos, "Corrosion investigation of stainless steel water pump components," *Engineering Failure Analysis*, 2016.

[11] G. Yang, V. Pointeau, E. Tevissen, and A. Chagnes, "A review on clogging of recirculating steam generators

in Pressurized-Water Reactors," *Prog. Nucl. Energy*, vol. 97, pp. 182–196, 2017.

[12] M. E. Orchard and G. J. Vachtsevanos, "A particle-filtering approach for on-line fault diagnosis and failure prognosis," *Trans. Inst. Meas. Control*, vol. 31, no. 3–4, pp. 221–246, 2009.

[13] F. Cadini, E. Zio, and D. Avram, "Model-based Monte Carlo state estimation for condition-based component replacement," *Reliab. Eng. Syst. Saf.*, vol. 94, no. 3, pp. 752–758, Mar. 2009.

[14] B. A. Zárate, J. M. Caicedo, J. Yu, and P. Ziehl, "Bayesian model updating and prognosis of fatigue crack growth," *Eng. Struct.*, vol. 45, pp. 53–61, Dec. 2012.

[15] P. Baraldi, M. Compare, S. Sauco, and E. Zio, "Ensemble neural network-based particle filtering for prognostics," *Mech. Syst. Signal Process.*, vol. 41, no. 1–2, pp. 288–300, 2013.

[16] M. E. Orchard, F. A. Tobar, and G. J. Vachtsevanos, "Outer Feedback Correction Loops in Particle Filtering-based Prognostic Algorithms: Statistical Performance Comparison," *Stud. Informatics Control*, vol. 18, no. 4, pp. 295–304, 2009.

[17] L. Tang, J. Decastro, G. Kacprzynski, K. Goebel, and G. Vachtsevanos, "Filtering and prediction techniques for model-based prognosis and uncertainty management," in *2010 Prognostics and System Health Management Conference, PHM '10*, 2010.

[18] Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht, "Remaining useful life prediction of lithium-ion battery with unscented particle filter technique," *Microelectron. Reliab.*, vol. 53, no. 6, pp. 805–810, 2013.

[19] Y. Xing, E. W. M. Ma, K.-L. Tsui, and M. Pecht, "An ensemble model for predicting the remaining useful performance of lithium-ion batteries," *Microelectron. Reliab.*, vol. 53, no. 6, pp. 811–820, Jun. 2013.

[20] M. Jouin, R. Gouriveau, D. Hissel, M. C. Péra, and N. Zerhouni, "Prognostics of PEM fuel cell in a particle filtering framework," *Int. J. Hydrogen Energy*, vol. 39, no. 1, pp. 481–494, 2014.

[21] B. Zhang, C. Sconyers, C. Byington, R. Patrick, M. E. Orchard, and G. Vachtsevanos, "A probabilistic fault detection approach: Application to bearing fault detection," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 2011–2018, 2011.

[22] J. Bogdanoff and F. Kozin, "Probabilistic models of cumulative damage," *Wiley, New York*, Oct. 1985.

[23] W.-F. Wu, "On the Markov approximation of fatigue crack growth," *Probabilistic Eng. Mech.*, vol. 1, no. 4, pp. 224–233, Dec. 1986.

[24] W. F. Wu and C. C. Ni, "Probabilistic models of fatigue crack propagation and their experimental verification," *Probabilistic Eng. Mech.*, vol. 19, no. 3, pp. 247–257, Jul. 2004.

[25] K. Sobczyk and B. F. (Billie F. Spencer, *Random fatigue : from data to theory*. Academic Press, 1992.

[26] M. M. Rocha and G. I. Schuëller, "A probabilistic criterion for evaluating the goodness of fatigue crack growth models," *Eng. Fract. Mech.*, vol. 53, no. 5, pp. 707–731, Mar. 1996.

[27] X. Su, S. Wang, M. Pecht, L. Zhao, and Z. Ye, "Interacting multiple model particle filter for prognostics of lithium-ion batteries," *Microelectron. Reliab.*, vol. 70, pp. 59–69, 2017.

[28] P. Paris and F. Erdogan, "A Critical Analysis of Crack Propagation Laws," *J. Basic Eng.*, vol. 85, no. 4, p. 528, Dec. 1963.

[29] G. R. Irwin, "Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate," *J. Appl. Mech.*, 1957.

[30] E. Myötyri, U. Pulkkinen, and K. Simola, "Application of stochastic filtering for lifetime prediction," *Reliab. Eng. Syst. Saf.*, vol. 91, no. 2, pp. 200–208, Feb. 2006.

[31] C. C. Ni, "Formulation of a Polynomial Stochastic Fatigue Crack Growth Model," *Adv. Mater. Res.*, vol. 909, pp. 467–471, 2014.

[32] H. Salimi, M. Pourgol-Mohammad, and S. Kiad, "Assessment of Stochastic Fatigue Failures Based On Deterministic Functions," in *13th International conference on Probabilistic Safety Assessment and Management (PSAM 13)*, 2016.

[33] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[34] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian

filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[35] P. Baraldi, F. Cadini, F. Mangili, and E. Zio, "Model-based and data-driven prognostics under different available information," *Probabilistic Eng. Mech.*, vol. 32, pp. 66–79, 2013.

[36] E. Zio and G. Peloni, "Particle filtering prognostic estimation of the remaining useful life of nonlinear components," *Reliab. Eng. Syst. Saf.*, vol. 96, no. 3, pp. 403–409, 2011.

[37] D. W. Opitz and J. W. Shavlik, "Actively Searching for an Effective Neural Network Ensemble," *Conn. Sci.*, vol. 8, no. 3–4, pp. 337–354, Dec. 1996.

[38] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, no. 1–2, pp. 239–263, May 2002.

[39] F. Moreno-Seco, J. Iñesta, P. De León, and L. Micó, "Comparison of classifier fusion methods for classification in pattern recognition tasks," *Lect. Notes Comput. Sci.*, vol. 4109, pp. 705–713, 2006.

Paper 3:

Hoang-Phuong Nguyen, Jie Liu, and Enrico Zio. (2020). 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. *Applied Soft Computing*, vol. 89, pp. 106116.

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

Hoang-Phuong Nguyen[a], Jie Liu[b], Enrico Zio[c,d,e,*]

[a] *Chair on System Science and the Energetic Challenge, EDF Foundation, CentraleSupélec, Université Paris-Saclay, 9 rue Joliot-Curie, 91192 Gif-sur- Yvette, France*

[b] *School of Reliability and Systems Engineering, Beihang University, 37 Xueyuan Road, Haidian, Beijing, China*

[c] *MINES ParisTech / PSL Université Paris, Centre de Recherche sur les Risques et les Crises (CRC), 1 rue Claude Daunesse, 06904 Sophia Antipolis, France*

[d] *Dipartimento di Energia, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy*

[e] *Eminent Scholar, Department of Nuclear Engineering, Kyung Hee University, 26 Kyungheedae-ro, Hoegi-dong, Dongdaemun-gu, Seoul, South Korea*

Abstract

Developing an accurate and reliable multi-step ahead prediction model is a key problem in many Prognostics and Health Management (PHM) applications. Inevitably, the further one attempts to predict into the future, the harder it is to achieve an accurate and stable prediction due to increasing uncertainty and error accumulation. In this paper, we address this problem by proposing a prediction model based on Long Short-Term Memory (LSTM), a deep neural network developed for dealing with the long-term dependencies in time-series data. Our proposed prediction model also tackles two additional issues. Firstly, the hyperparameters of the proposed model are automatically tuned by a Bayesian optimization algorithm, called Tree-structured Parzen Estimator (TPE). Secondly, the proposed model allows assessing the uncertainty on the prediction. To validate the performance of the proposed model, a case study considering steam generator data acquired from different French nuclear power plants (NPPs) is carried out. Alternative prediction models are also considered for comparison purposes.

Keywords: Prognostics and health management; Time-series forecasting; Multi-step ahead prediction; Long short-term memory; Nuclear power plant prognostics; Steam generator

Nomenclature

Abbreviations

AI	Artificial intelligence

ANN	Artificial neural network
ARMA	Autoregressive moving average
ARIMA	Autoregressive integrated moving average
BO	Bayesian optimization
CNN	Convolutional neural network
DBN	Deep belief network
EDF	Électricité de France
EI	Expected improvement
FNN	False nearest neighbor
FIS	Fuzzy interference system
LSTM	Long short-term memory
MAPE	Mean absolute percentage error
MASE	Mean absolute scaled error
MC	Monte Carlo
MIMO	Multi-input multi-output
MLP-MIMO	Multi-output multilayer perceptron neural network using MIMO strategy
MLP-REC	Single-input multilayer perceptron neural network using recursive strategy
MSE	Mean square error
NF	Neuro-fuzzy
NPP	Nuclear power plant
PHM	Prognostics and health management
PWR	Pressurized water reactor
RBM	Restricted Boltzmann machine
RMSE	Root mean square error
RNN	Recurrent neural network
RS	Random search
RUL	Remaining useful life
SG	Steam generator
SVM	Support vector machine
SVR-MIMO	Multi-Input Multi-Output support vector regression using MIMO strategy
SVR-REC	Single-input support vector regression using recursive strategy
TPE	Tree-structured Parzen estimator
WRL	Wide range level

Symbols

C_t	output of the LSTM cell state at time t
$\tilde{C}_t$	potential values of the LSTM cell state at time t
d	embedding dimension
f	prediction model
f_R	one-step ahead prediction model
$f_{D,h}$	direct prediction model for the horizon time t+h
f_{MIMO}	MIMO prediction model
f_t	output of the LSTM forget gate at time t
h_t	output of the repeating network module of a LSTM at time t
H	prediction horizon
i_t	output of the LSTM input gate at time t
L	number of hidden layers in the LSTM network
max_iter	number of optimization iterations
N_{init}	number of TPE startup iterations
N_{MC}	number of MC dropout realizations
o_t	output of the LSTM output gate at time t
Pr_{bad}	probability of being in the bad group in the TPE algorithm
Pr_{good}	probability of being in the good group in the TPE algorithm
$r^{(l)}$	vector of independent Bernoulli random variables at layer l of the LSTM network
t	time t
x_t	observed value at time t
$\hat{x}_t$	predicted value at time t
$y^{(l)}$	output vector of the hidden layer l of the LSTM network
$\tilde{y}^{(l)}$	thinned output vector of the hidden layer l obtained by using dropout
$z^{(l)}$	input vector of the hidden layer l of the LSTM network
(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
σ	a network layer in the repeating network module of a LSTM
θ	hyperparameter set

θ^* selected hyperparameter set

1. Introduction

In recent years, prognostics and health management (PHM) has attracted increasing attention from academic researchers and practitioners in different industrial sectors. The primary characteristic of PHM is that it can enable estimation and prediction of the health state of components and systems, by making use of past, present and future knowledge, information and data on their operations, and this capability can be used to identify malfunctions and anticipate failure patterns [1]. This allows estimating the remaining useful life (RUL) of components and systems, and scheduling the maintenance interventions for the most opportune and convenient instances. By so doing, the availability and reliability of the assets can be maximized, with reduced unscheduled shutdowns and maintenance costs.

Developing models for efficient PHM is a challenging task, with several issues to be addressed. Among them, determining an appropriate horizon for the prediction, i.e. how far into the future the model should predict and with what accuracy, is crucial and application dependent, in the sense that it depends on the use that is made of the prediction, typically for taking some decisions [2]. For instance, the selected horizon should be suitably long to allow that maintenance actions be timely carried out. This often requires long-term predictions in practice. However, long-term predictions are known to suffer from increasing uncertainties, which may arise from the accumulation of prediction errors or from the complex interactions and correlations in the underlying process at different time steps. This has challenged and somewhat limited the research on long-term prognostics for many years [3], [4]. To address this problem, the main focus of this paper is the development of a prognostic framework for the long-term prediction of parameters relevant to the operation of the steam generators (SGs) in nuclear power plants (NPPs).

Depending on the information and data available for the model development, prognostic approaches can be divided into two main categories: model-based and data-driven approaches [5]. Model-based approaches predict the degradation evolution by formalizing it into physical analytical or computational models. These approaches are used in applications where the model of the degradation process exists and is not too complicated, e.g. models of fatigue crack growth [6], [7], of capacity degradation in Lithium-ion batteries [8], [9]. Alternatively, data-driven approaches utilize condition monitoring data collected from sensors to learn and predict the component or system behavior and degradation via statistical and artificial intelligent (AI) models, such as autoregressive integrated moving average (ARIMA) [10], artificial neural network (ANN) [11]–[14], neuro-fuzzy (NF) [2] and support vector machine (SVM) [15]–[17]. Due to the data-adaptive nature, data-driven approaches are quite appropriate for prognostic real-world applications where models are not available whereas obtaining condition monitoring data is becoming convenient with smart sensors.

When applying data-driven approaches to prediction, models like ANN, NF and SVM are usually

limited in extracting and utilizing the temporal information of the given data which is necessary for prediction purposes. More specifically, these approaches consider each time step independently and make the prediction as a static mapping, which often takes into account only the current state of the process [18]. Recently, a connectionist neural network model called recurrent neural network (RNN) has been proposed to account for the dynamics [19]. RNN is a network with feedback connections from the hidden and output layers to the preceding ones, by which the dynamics of sequential data can be captured and the memories of the previous patterns are retained via cycles in the network. In the last decade, RNNs have been extensively investigated for a variety of prognostic applications, including engine systems [20]–[23], lithium-ion batteries [24]–[26], rolling element bearings [27]–[30] and fuel cells [31], [32]. Zhang et al. [24] utilized a RNN to extract the long-term dependencies underlying in the battery capacity degradation process. The obtained results showed that RNN outperformed the classical data-driven models in prediction robustness and accuracy. In [30], the RNN model was modified with an incremental learning technique, which was then applied for predicting the long-term propagation of rolling element bearing degradation to failure. RNN has also been used in the construction of health indicators for generator bearings of wind turbines [33]. The obtained RNN-based health indicator showed its effectiveness for improving the prediction performance of the bearing RULs.

In this paper, a variational model of RNN, which is called long short-term memory (LSTM), is employed for developing a prognostic framework for SGs in NPPs. An important feature of the proposed framework is the ability to deal with a long-term prediction horizon. A multi-input multi-output (MIMO) prediction strategy and LSTM network are integrated to predict the equipment health conditions for multiple steps ahead. The proposed framework also handles two practical problems in prediction model development. On one hand, the performance of the prediction model depends on the time-series data acquired from sensors: any anomalous or missing data that can degrade the performance should be dealt with. On the other hand, an optimal model setting for different available datasets is crucial to successfully apply the prediction model to practical problems. The effective handling of these two issues is another contribution of this work.

In summary, the main contributions of this paper are as follows:

(1) A data preprocessing module consisting of an outlier removal and a missing data imputation methods is introduced for filtering and preparing the data for the prediction task.
(2) Automatic hyperparameter optimization based on the Tree-structured Parzen Estimator (TPE) algorithm is performed. The obtained results are compared to the conventional random search (RS) algorithm with respect to different data scenarios.
(3) Dropout regularization and Monte Carlo (MC) techniques are integrated to assess the prediction uncertainty of the proposed model.
(4) A case study using the data of SGs in French NPPs measured during the period 1992-2007 is carried out to evaluate the performance of the proposed LSTM-based framework for long-term

prediction. To the authors' knowledge, this is the first study using LSTM for the long-term prediction and used on NPP SGs. Other prediction models are considered for comparison purposes.

The rest of the paper is organized as follows. Section 2 presents a brief introduction of time series prediction approaches. Section 3 introduces the LSTM neural network. The proposed multi- step ahead prediction model is presented in Section 4. Section 5 describes the experimental case study, and the obtained results and their discussion are presented in Section 6. Finally, Section 7 concludes the paper.

2. Time series prediction: background and related work

Time series is a sequence of observations collected over time from a particular measured variable of an engineering component or system. In general, the main objectives of time series analysis are: characterization, modeling and prediction (also called forecasting) [34]. Firstly, *characterization* aims to extract inherent structural characteristics of the measured variable, e.g. temporal trend, variance and seasonality. Then, the extracted information may be used to formulate an appropriate model for capturing long-term behavior of the system (*modeling*), or to estimate the evolution of the variable in the future *(prediction)*. This section presents a brief introduction of time series prediction approaches and further discusses the strategies for multi-step ahead prediction.

2.1. Time series prediction

The beginning of time series prediction might be set in 1927 when Yule [35] introduced the first autoregressive technique for predicting the annual number of sunspots. In that original work, the prediction of the next time step was estimated as a weighted sum of previous observations of the time series. This idea has become the basis of data-driven approaches for time series prediction since then.

Among data-driven approaches, statistical models which attempt to express the future values as a linear function of the historical data have been popular and widely used in many applications of time series prediction, such as wind energy generation [36]–[39], weather forecasting [40]–[42], market demand forecasting [43], [44] and nuclear component prognostics [10]. Erdem and Shi [36] used an autoregressive moving average (ARMA) for predicting wind speed and direction. Kavasseri and Seetharamen [38] proposed a variant model of ARIMA which was called fractional-ARIMA in order to extract the long dependency features of the time series data and enhance the prediction accuracy over long-term horizons. In a nuclear application, Nguyen et al. [10] applied an ARIMA model to predict the long-term evolution of the tube supporting plate clogging degradation of NPP SGs for the first time, in which the predictions were performed up to 3 months ahead.

Although statistical models have shown their notable prediction accuracy and flexibility in different time series applications, one of their major drawbacks is the presumed linear form of the associated data, which has limited their applicability to many modern dynamic systems where the collected data

are usually nonlinear and non-stationary [45], [46]. To address this problem, several machine learning algorithms have been employed in the time series prediction area, such as SVM [47]–[49], ANNs [50]–[53] and fuzzy interference system (FIS) [54], [55]. Unlike statistical approaches, machine learning models can automatically learn arbitrary complex mappings between inputs and outputs directly from the historical data and perform accurate predictions without any assumption about the mapping functions required. In addition, another advantage of machine learning approaches is the recently rapid advancements of information science technologies, particularly Big Data and deep learning techniques, which are offering opportunities for new developments in time series analysis. Kuremoto et al. [56] proposed a deep belief network (DBN) with restricted Boltzmann machines (RBMs) to address problems of initialization and local optima in chaotic time series forecasting, which was shown to outperform conventional shallow learning models. Wang et al. [57] presented a prediction model for probabilistic wind power prediction based on a convolutional neural network (CNN) model to automatically extract deep invariant structures and hidden nonlinear features exhibited at separated frequency bands of the data. A specialized kind of deep neural networks proposed for sequential data analysis is RNNs, which aim to capture the dynamics of sequential data and be able to retain the memories of the previous patterns via cycles of feedback connections between the network layers [19]. Wang and Li [46] presented a hybrid model integrating a RNN model and an optimal feature extraction technique for multi-step ahead wind speed prediction. Likewise, Li et al. [58] utilized LSTM RNN for predicting 5 steps ahead of the wind speed time series.

2.2. Multi-step ahead prediction strategies

Given a univariate time series of the observations collected up to time t, $\{x_1, x_2, ..., x_t\}$, the main goal is to predict the H next observations $\{\hat{x}_{t+h}\}, h \in [1, H]$, which can be formulated as below:

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

where f is the prediction model and d is the embedding dimension (or the number of lagged values).

Depending on the desired horizon H, a prediction method can be classified into short-, medium-, or long-term prediction. As aforementioned, the further in the future one attempts to predict, the harder it is to achieve an accurate prediction due to the increasing uncertainty and accumulation of errors. To address this problem, there are three popular prediction strategies, namely recursive, direct and MIMO predictions, which are described as follows [2].

2.2.1. Recursive prediction strategy

The recursive strategy attempts to train a model focused solely on one-step ahead prediction:

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

where f_R is the one-step ahead prediction model.

After the model is trained, the predictions are recursively estimated. In other words, intermediate predictions are used as inputs for predicting next values until the prediction at the time horizon $H, \hat{x}_{t+H}$, is obtained:

$$\begin{aligned} \hat{x}_{t+1} &= f_R(x_t, x_{t-1}, ..., x_{t-d+1}) \\ \hat{x}_{t+2} &= f_R(\hat{x}_{t+1}, x_t, ..., x_{t-d+2}) \\ &\vdots \qquad\qquad \vdots \\ \hat{x}_{t+H} &= f_R(\hat{x}_{t+H-1}, \hat{x}_{t+H-2}, ..., \hat{x}_{t+H-d+1}) \end{aligned} \tag{3}$$

An advantage of the recursive strategy is its low computational cost since only one single model is required for training. However, the prediction errors of the previous steps can easily accumulate in the next predictions, resulting in the decrease of accuracy in the long run. Besides, this prediction strategy does not take into account the data dependencies among time steps.

2.2.2. Direct prediction strategy

In contrast to the recursive strategy which uses a single model, the direct strategy [59] constructs a set of *H* different models for different time steps and the same input data are used for feeding all the models as below:

$$\begin{aligned} \hat{x}_{t+1} &= f_{D,1}(x_t, x_{t-1}, ..., x_{t-d+1}) \\ \hat{x}_{t+2} &= f_{D,2}(x_t, x_{t-1}, ..., x_{t-d+1}) \\ &\vdots \qquad\qquad \vdots \\ \hat{x}_{t+H} &= f_{D,H}(x_t, x_{t-1}, ..., x_{t-d+1}) \end{aligned} \tag{4}$$

where $f_{D,h}$ is the direct prediction model tuned to perform the prediction $\hat{x}_{t+h}$ at time $t+h, h \in [1, H]$.

In the direct strategy, each prediction model is trained and dedicated to a certain horizon, so the error accumulation can be avoided. However, training different prediction models will greatly increase the prediction complexity and time consumption, and, like the recursive strategy, the direct strategy does not take into account the dependencies among time-series observations.

2.2.3. MIMO prediction strategy

Unlike the recursive and direct approaches, the MIMO approach is a multiple output strategy, in which the output of the prediction model is a vector of future values predicted by using only one model [60]:

$$\{\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H}\} = f_{MIMO}(x_t, x_{t-1}, ..., x_{t-d+1}) \tag{5}$$

where f_{MIMO} is the multiple output prediction model. In this sense, the objective function during the model training is to simultaneously minimize the prediction errors on different horizons. By so doing, the MIMO strategy can preserve the temporal stochastic dependencies of sequential data, addressing the limitation of the recursive and direct approaches. On the other hand, the computational cost of the

MIMO approach is less than that of the direct approach because it requires only one model to be trained.

3. Long short-term memory (LSTM)

3.1. Network architecture

LSTM is a variant of RNNs developed for dealing with the long-term dependency problem, which is also known as "vanishing gradients" or "exploding gradients" problem [61]. In general, an LSTM consists of a chain of repeating network modules, in which each module contains four interacting layers, as illustrated in Fig. 1 [62].

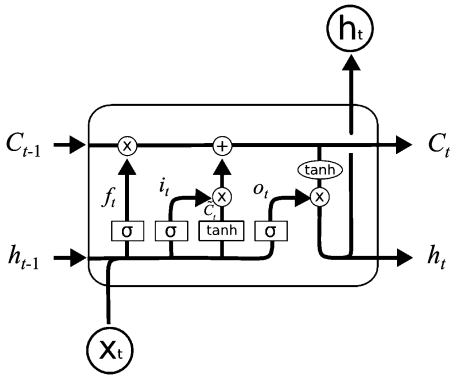


Fig. 1. The schematic of a repeating network module in a LSTM network.

The key element of a LSTM network is the cell state *C*, which is depicted as the horizontal line running through the top of the diagram in Fig. 1. This cell state plays a role as a network memory, where information is added or removed via regulated structures called gates, which can optionally let information through. They are composed of a sigmoid neural network layer and a pointwise multiplication operation. An LSTM consists of three gates, including forget, input and output gates, in order to protect and control the cell state. Details on the LSTM procedure are described as follows.

At time *t*, an input x_t is fed to the network. The forget gate first decides which information from the previous output h_{t-1} is discarded or kept, and then the output of the forget gate is calculated as:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{6}$$

where (W_f, b_f) are the input weights and bias of the forget gate, respectively, σ is a nonlinear function (e.g. sigmoid function) and " . " means matrix multiplication.

The next step is to determine which new information will be stored in the cell state, leading to the two following calculations. First, the input gate decides which states will be updated; then, a *tanh* layer generates a vector of new values $\tilde{C}_t$ that could be added to the cell state, as follows:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \tag{7}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right), \tag{8}$$

where (W_i, b_i) and (W_C, b_C) are the input weights and bias of the input gate and the cell state layer, respectively. The outputs obtained from the forget gate, input gate and *tanh* layer are, then, used to update the new cell state C_t:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \tag{9}$$

Finally, the network output h_t is generated by the output gate and a *tanh* function, as:

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right), \tag{10}$$

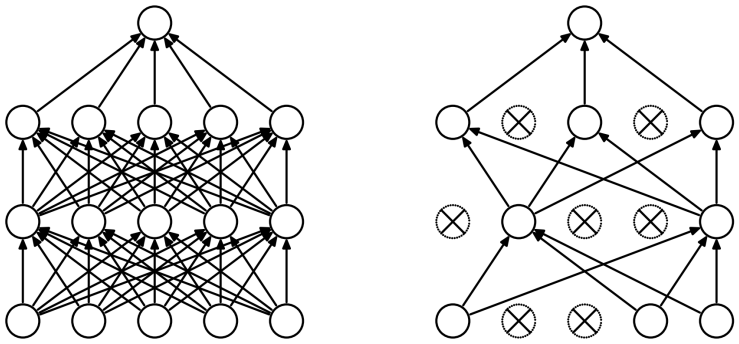
$$h_t = o_t * \tanh(C_t), \tag{11}$$

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

3.2. Dropout regularization

A well-known and critical problem of deep neural networks such as LSTM is overfitting [63]. That is, when the training data is limited, complicated mappings between the inputs and outputs that are learned by the network might be the result of sampling noise, which only exist in the training set but not in the real test set. One way to regularize such a network is averaging the outputs of all possible configurations of the parameters, in which each configuration is weighted by its posterior probability given by the training data [64]. This method can be applied only for simple or small networks. With large neural networks, the computation for training many different network architectures or training one architecture on different data sets is very expensive. Dropout is a technique that addresses this issue [64].

A motivation for dropout comes from a theory of sexual reproduction [65], in which new genes are naturally selected to spread throughout the population based on their competitiveness and less co-adaptation which may reduce the chance of a new gene improving the fitness of an individual. Likewise, dropout aims to train each hidden unit in a neural network with a randomly chosen sample of other units. By dropping a unit out, we temporarily remove it from the network along with all its connections during the training process as illustrated in Fig. 2, in order to prevent units from high co-adaptation. By so doing, each hidden unit becomes more robust and is able to create useful features on its own without relying on other units, which helps the network avoid overfitting.



(a) (b)

Fig. 2. An example of a dropout network model [64]: (a) A fully connected 2-hidden layers network; (b) The network obtained by applying dropout. Crossed units are excluded.

Consider a neural network with L hidden layers, in which the input and output vectors of layer l (for $l \in \{1, \ldots, L\}$) are denoted as $z^{(l)}$ and $y^{(l)}$, respectively. $w^{(l)}$ and $b^{(l)}$ are the weights and biases of layer l, respectively. For a standard neural network, the feed-forward operation can be described as:

$$z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)}, \tag{12}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}), \tag{13}$$

where f is the activation function and i denotes the index of hidden unit, as illustrated in Fig. 3(a).

With a dropout network (Fig. 3(b)), a vector of independent Bernoulli random variables $r^{(l)}$ with probability p is used at each hidden layer l to generate the thinned outputs $\tilde{y}^{(l)}$ as follows:

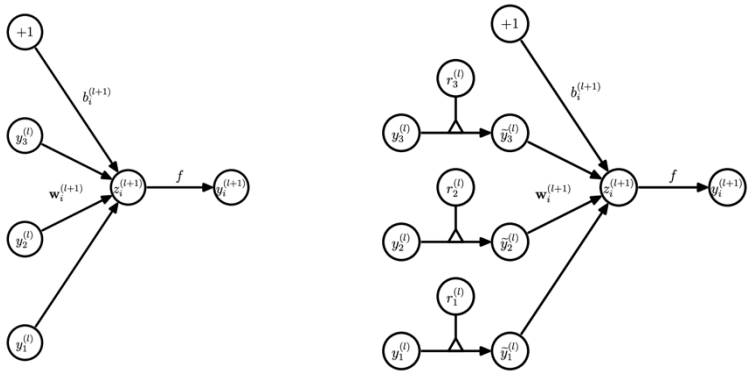
$$r_j^{(l)} \sim Bernoulli(p), \tag{14}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)}, \tag{15}$$

where * denotes an element-wise product. The thinned outputs are, then, used as inputs to the next layer of the feed-forward operation:

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}, \tag{16}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}), \tag{17}$$



(a) Standard network (b) Dropout network

Fig. 3. Comparison of the basic operations of a standard and dropout network [64].

The dropout technique was shown to significantly reduce overfitting and improve the performance of standard neural networks in a wide variety of application domains, including handwriting recognition, speech recognition, image processing, object classification and computational biology [64]. In this paper, dropout is used in the input and hidden layers of the proposed LSTM model in order to prevent overfitting and quantify the uncertainty information of the multi-step ahead predictions, which is further described in Section 4.3.

4. Proposed LSTM-based prognostic framework

In this section, we present a prognostic framework for the multi-step ahead prediction of the time-series data from SGs, as illustrated in Fig. 4.

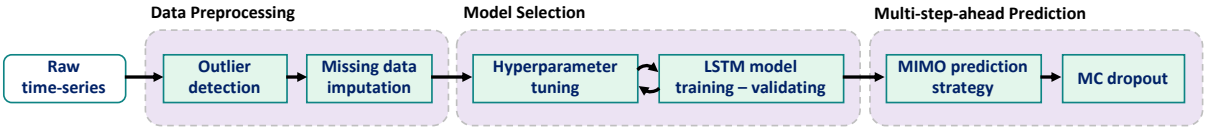


Fig. 4. The flowchart of the proposed multi-step ahead prediction framework for SGs.

The proposed framework consists of three main stages: data preprocessing, model selection and multi-step ahead prediction. Firstly, the data preprocessing stage is responsible for preparing the data for training and testing the prediction model. Then, in the second stage, a LSTM-based model is built for the MIMO prediction using the training data and its hyperparameters are automatically optimized with the objective function of minimizing the validation error. In the last stage, the performance of the trained prediction model is validated for multi-step ahead prediction and a MC dropout technique is used to capture the prediction uncertainty. The procedure of the proposed framework can be summarized as in Algorithm 1, where *max_iter* is the number of optimization iterations and N_{MC} is the

number of MC dropout realizations. The details of each stage are given in the following sections.

Algorithm 1. Procedure of the proposed multi-step prediction framework

Input: A raw time series data collected up to time t: $\{x_1, x_2, ..., x_t\}$

Output: Predictions of H next observations and their uncertainty information

Preprocessing stage

1. Detect and remove outliers
2. Impute missing data points

Model selection stage

3. **for** i in {1,…,*max_iter*} **do**
    g. Select the optimal network hyperparameters at the ith trial with TPE
    h. Validate the hyperparameters by using k-fold cross-validation
    i. Update the fitness value with the average training error measured over k folds
4. Select the best hyperparameter setting with the lowest fitness value

Multi-step ahead prediction stage

5. **for** i in {1,…,N_{MC}} **do**
    c. Build a LSTM-based prediction model with the selected hyperparameters
    d. Perform the predictions for H steps ahead $\{\hat{x}_{t+h}\}, h \in [1, H]$ by using the MIMO prediction strategy
6. Calculate the mean and confidence interval of the predictions over N_{MC} realizations

4.1. Data preprocessing

As mentioned in Section 1, the quality of the observation data for training is one of the most important factors for the successful performance of a prediction model. Due to the errors during sensor measurements or signal transmission, the acquired observations may include missing and anomalous data points, e.g. outliers, which can negatively impact the model performance. In this study, we adopt a raw data preprocessing module focusing on the two following tasks: 1) detecting and removing outliers; 2) imputing missing data points, the number of which may increase after removing outliers.

The first problem is addressed by using the Isolation Forest, an outlier detection technique built on the basis of decision trees [66]. This technique is based on an assumption that outliers are few, different and susceptible to a mechanism called isolation. In comparison with conventional distance and density measures, isolation has been proved to be a much more effective indicator to detect anomalies. In addition, Isolation Forest also requires a small linear time complexity. Further details on the algorithm of Isolation Forest can be found in [66]. Once outliers are reduced, a local polynomial regression technique is used to reconstruct missing data samples and reduce noises. The preprocessed data is later

used for training and testing the prediction model in the following stages.

4.2. Model selection

4.2.1. Prediction horizon

Several research works have been carried out on determining an optimal horizon of prediction in order to provide predictions accurately and timely, and to ensure the usefulness of the prognostic model. However, to the authors' knowledge, there is no general rule reported for dealing with this issue. We have carried out a review on the horizons selected in recent prediction studies for industrial applications during 2015-2019 [15], [58], [67]–[91] and the result is summarized in Fig. 5. The result shows that multi-step ahead prediction has been less studied than single-step ahead prediction, and that most of the works were carried out with horizons ranging from 3 to 6 steps ahead. To demonstrate the effectiveness of the proposed model, a prediction horizon of 15 steps ahead is investigated in this study.

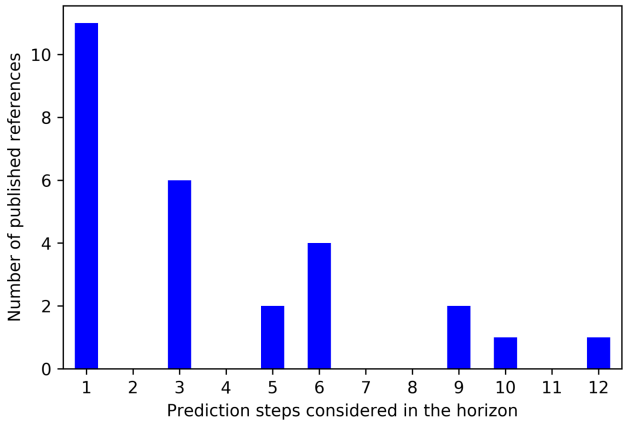


Fig. 5. Prediction horizons of recent studies.

4.2.2. Hyperparameter optimization

In machine learning, hyperparameters define the model architecture and control the learning process, e.g. the number of hidden layers, activation function type and learning rate. Automatic hyperparameter optimization is playing a fundamental role in the development of machine learning models, including the recent deep neural networks, e.g. LSTM, whose learning performance greatly depends on a number of hyperparameter choices [92]. Automatic hyperparameter optimization has several important advantages, such as: 1) reduction of the human effort in deploying machine learning, which is important in application because different hyperparameter configurations are needed for different datasets [93]; 2) improvement of the performance of machine learning models, by choosing the most appropriate (according to specified objectives) hyperparameters values for the target application at hand [94], [95]; 3) increase of the reproducibility of results, as automatic hyperparameter optimization is clearly more reproducible than manual tuning by human and allows fair comparisons

between different models by giving them the same level of tuning for the specific application [96].

In this study, we implement a variant of Bayesian optimization (BO), called Tree-structured Parzen Estimator (TPE) [97], to automatically optimize the hyperparameters of the proposed prediction model.

A common advantage of BO approaches is that they require less function evaluations than other classical optimization approaches, such as grid search or RS. This is because these approaches learn and select the best hyperparameter sets based on their distributions describing the fitness scores in the previous iterations. Thus, the number of samples drawn from the hyperparameter search space is probabilistically guided and reduced, allowing for proper evaluations of the most promising candidates for hyperparameter choices.

Recently, TPE has been put forward to address the limitation of the conventional BO approaches in working with categorical and conditional parameters, and, thus, to improve the hyperparameters selection process [97]. It has, then, been widely used to tune machine learning models in various applications, such as image processing [96], [98]–[101], electricity price forecasting [102], solar irradiance forecasting [103], rail defect prediction [104], occupational accident prediction [105]. Parzen-window density estimation, which is also known as kernel density estimation, is a non-parametric way to build a probability density function from empirical data. In the TPE algorithm, each sample from the empirical data defines a Gaussian distribution with a mean equal to the hyperparameter value and a specified standard deviation. At the start-up iterations, a random search is employed to initialize the distributions by sampling the response surface $\{\theta^{(i)}, y^{(i)}, i = 1, \ldots, N_{init}\}$, where θ denotes the hyperparameter set, y is the corresponding value on the response surface, i.e. the validation loss or the fitness value, and N_{init} is the number of start-up iterations. Then, the hyperparameter space is divided into two groups, namely *good* and *bad* samples, based on their fitness values and a predefined threshold value y^* (usually set to 15% [92]), as follows:

$$p(\theta \mid y) = \begin{cases} \Pr_{good}(\theta) & if\ y < y^* \\ \Pr_{bad}(\theta) & if\ y \geq y^* \end{cases} \tag{18}$$

where $\Pr_{good}$ and $\Pr_{bad}$ are the probabilities that the hyperparameter set θ is in the *good* and *bad* groups, respectively. Fig. 6 illustrates an example of the TPE initialization process for the hyperparameter distributions, with y^* = 15% and N_{init} = 100. The red points are the samples with the lowest fitness values after evaluation, thus being classified into the *good* group $\Pr_{good}$ whereas the others form the *bad* group $\Pr_{bad}$. In this way, the selection of optimal hyperparameters does not rely on the best observation, but on a set of best observations and their distributions. Then, the more iterations one used for initialization, the better distribution we get at the beginning. An Expected Improvement (EI) is, then, calculated as follows:

$$EI(\theta) = \frac{\Pr_{good}(\theta)}{\Pr_{bad}(\theta)} \tag{19}$$

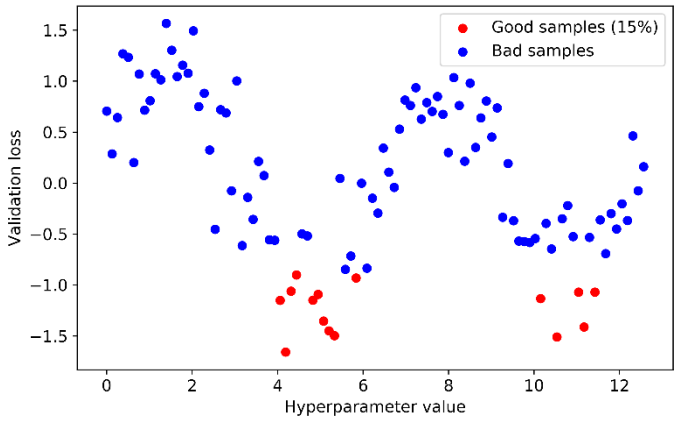


Fig. 6. Samples classification from the TPE initialization process.

At each iteration, the hyperparameter configuration $\theta*$ that maximizes the EI is chosen. Fig. 7 shows the flowchart of the TPE optimization procedure.

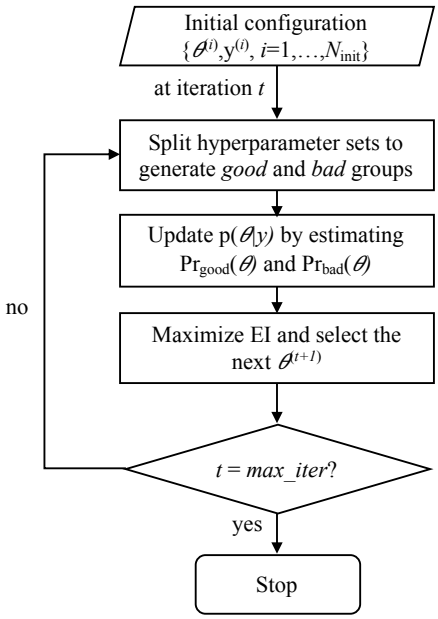


Fig. 7. Flowchart of the TPE optimization procedure.

4.3. Multi-step ahead prediction

In the testing stage, the MIMO prediction strategy introduced in Section 2.2.3, is used to predict the future values. As mentioned in Section 4.2.1, the prediction horizon h is set to 15-step ahead in this study, as shown in Fig. 8.

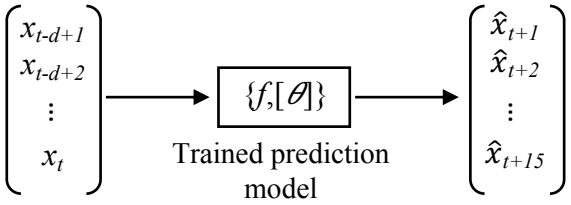


Fig. 8. Multi-step ahead prediction procedure.

To further assess the prediction performance, we adopt a Monte Carlo (MC) dropout technique [106] in order to capture the uncertainty information of the multi-step ahead predictions of the proposed model. It is important to note that the standard LSTM network is not capable to quantify the prediction uncertainty itself. In the MC dropout technique, a dropout probability is applied to all the weight layers in the network, which represents the network weights drawn from a Bernoulli distribution. Thus, the prediction uncertainty can be quantified by running several forward passes through the network. In this study, we perform $N_{MC} = 100$ stochastic forward passes, in which network units of each layer are randomly dropped out, and obtain the mean and confidence interval of the predictions.

5. Experimental study

5.1. SG data

In this paper, the prediction performance of the proposed model is evaluated on the SG data of French NPPs. SGs in pressurized water reactors (PWRs) are heat exchangers which use the heat from the primary reactor coolant to produce steam in the secondary side and, thus, drive the turbine generators. In addition, the SGs act as a safety barrier between the radioactive primary side and the non-radioactive secondary side. Due to their critical role in NPPs, any degradation mechanism in SGs should be monitored and prevented at the early stages of propagation. A widely used method of degradation monitoring is the analysis of the wide range level (WRL) dynamic behavior recorded by control sensors [107], [108].

WRL is one of the condition monitoring variables measured from the NPP SGs. It is estimated from the difference between the pressure measured at two difference heights, i.e. the dome and the bottom of the downcomer, as illustrated in Fig. 9 (label 18) [108]. Due to its nature, WRL is very sensitive to the temperature, the flow rate of the feed-water and the circulation ratio of the SG. Usually, WRL variations are monitored during slow transients and during manual control at low power load [108]. Among critical SG degradation mechanisms, clogging is a phenomenon where the flow holes of the tube support plates are partially or completely blocked by deposits, leading to the reduction of the circulation flow rate in the SGs [108]. Clogging in SG is a slow process which may take several years. In [109], it has been shown that the WRL of a SG is closely related to the clogging degradation. Thus,

the predictions of WRL can be converted to the clogging degradation state.

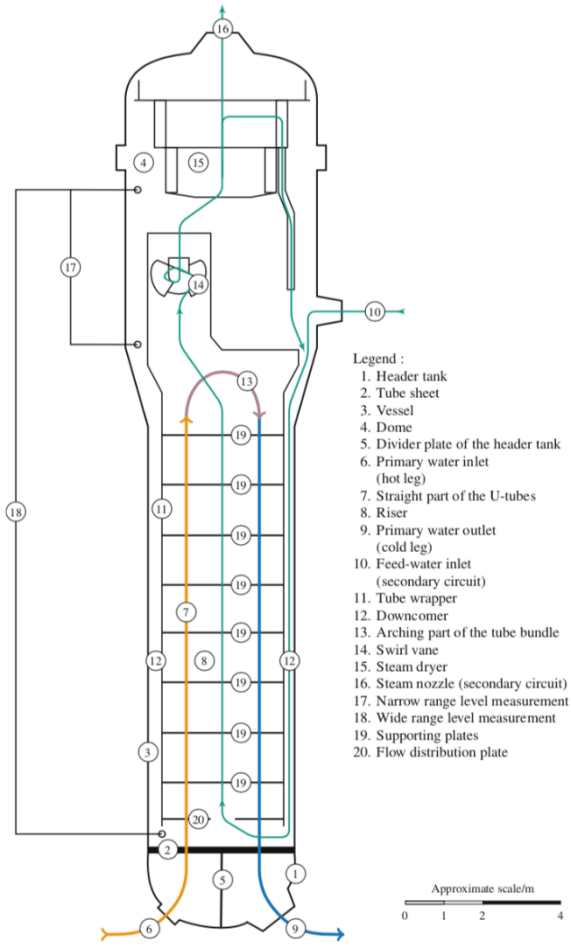
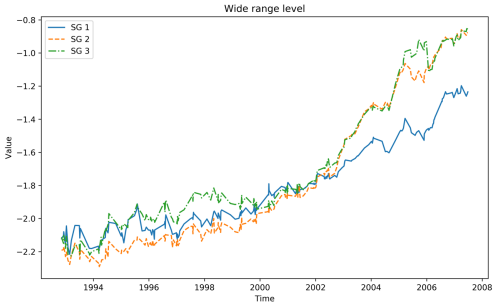


Fig. 9. The front-cut schematic of a 51B-model SG [108].

The original SG data employed in this study were collected from six SGs of two different 900-MW NPPs, which are operated by Électricité de France (EDF). Each plant consists of three SGs. The WRL data were recorded during the stationary regimes in which the power demand percentage is stably maintained greater than 90%, at an interval of 3 days from July 1992 to June 2007. Fig. 10 shows the temporal evolution of the WRL observations of the two NPPs. The names of the plants are omitted for confidentiality reasons.



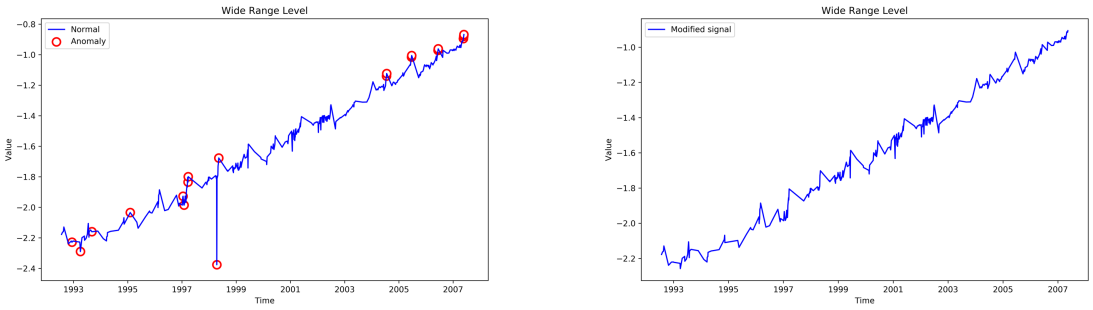
(a) Plant No. 1

(b) Plant No. 2

Fig. 10. Raw WRL measurements recorded from control sensors of different NPPs.

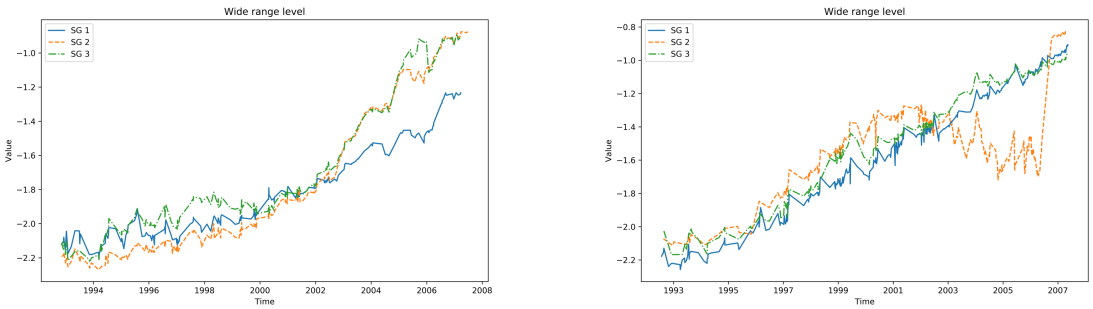
5.2. Data preprocessing

Before being used for the model development, the raw SG data are preprocessed by using the Isolation Forest and local regression approaches described in Section 4.1. Fig. 11 shows the results of applying the Isolation Forest for reducing outliers in the data of SG 1 of plant No. 2. In Fig. 11(a), the solid line indicates the normal measurements whereas the detected outliers are highlighted as circled points, which are later eliminated in Fig. 11(b). An interesting observation in Fig. 11(a) is the anomalous spike between 1997 and 1999. Without the outlier detection step, this sudden spike could highly impact, in a negative manner, on the prediction accuracy. After reducing the outliers, imputations for missing data samples are given. The preprocessed data of all SGs after the preprocessing stage are shown in Fig. 12.



(a) Original signal with outliers highlighted (b) Modified signal after outlier removal

Fig. 11. Applying the Isolation Forest to the data of SG 1 of plant No. 2.



(a) Plant No. 1 (b) Plant No. 2

Fig. 12. The results of the preprocessing stage for all SG data.

6. Results and discussion

After the preprocessing stage, each SG data series is divided into a training set and a testing set. The data for the first 11 years, from July 1992 to December 2002, which include a total of 1230 samples

at a 3-day interval, are selected to train the proposed prediction model and the next 5-year data with 510 samples are employed to test the model performance.

Before constructing the proposed model, we employ the false nearest neighbor (FNN) algorithm [110] to determine the appropriate embedding dimension *d* of the data series. The main idea of the FNN algorithm is to find the minimum dimension where the distances between the nearest neighbors in the time series do not significantly change in the next higher dimensional embedding. Fig. 13 shows the result of applying FNN to the data of SG 1 of plant No. 1. A threshold for identifying the minimum embedding dimension is set to 0. In this Figure, the minimum embedding dimension value is found at 12. We summarize the optimal embedding dimensions identified for all the SGs data series in Table 1.

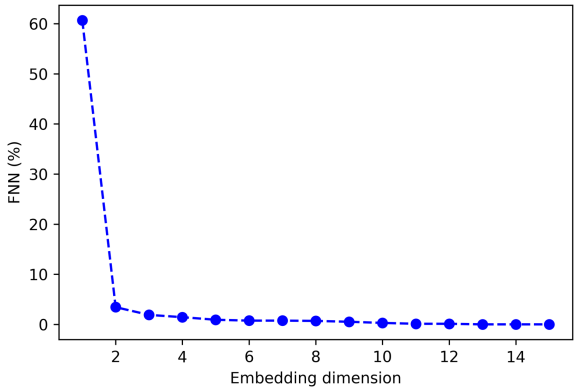


Fig. 13. FNN result for SG 1 of plant No. 1.

Table 1
Minimum embedding dimensions for all SGs.

Plant	No. 1			No. 2		
SG	1	2	3	1	2	3
Embedding dimension	12	13	9	9	11	6

In this study, we carry out three comparisons to evaluate the performance of the proposed prognostic model. The first comparison is conducted to analyze the viability of TPE in tuning the proposed model during the training stage. As a standard optimization approach, RS is considered for benchmarking purposes. Another comparison is, then, carried out to specifically validate the efficacy of dropout in the proposed prediction framework. In the third comparison, four hybrid prediction models, including single-output support vector regression using recursive strategy (SVR-REC), multi-output support vector regression using MIMO strategy (SVR-MIMO), single-output multilayer perceptron neural network using recursive strategy (MLP-REC) and multi-output multilayer perceptron neural network using MIMO strategy (MLP-MIMO), are employed as the benchmark models for comparison with the proposed model in multi-step ahead predictions. In this performance evaluation,

three prediction accuracy metrics are considered, including root mean square error (RMSE), mean absolute percentage error (MAPE) and mean absolute scaled error (MASE). Their definitions are given as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{x}_i - x_i\right)^2}\,, \tag{20}$$

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

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

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

6.1. Automatic hyperparameter optimization

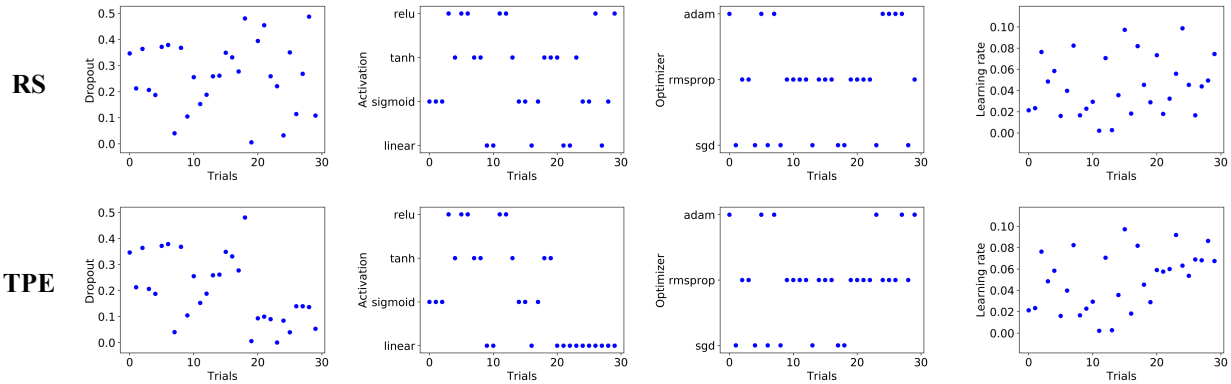
The proposed prediction model is constructed with one LSTM layer with 64 neurons. Four major hyperparameters of the model are to be tuned, including dropout rate, activation function type, optimizer type and learning rate. The details of the hyperparameter search space are shown in Table 2. For a fair comparison, the TPE and RS algorithms are evaluated by using the same model configurations and hyperparameter search space. The number of optimization trials is selected as 30 for the two algorithms. In addition, a *k*-fold cross-validation (k = 3 in this study) is adopted to prevent overfitting during training the model. The mean square error (MSE) is used as the objective function for model selection. In other words, at each optimization trial, the hyperparameter configuration with the lowest average prediction error evaluated by cross-validation is chosen. To achieve the training convergence, the number of training epochs is set to 100.

Table 2
Hyperparameters of the proposed prediction model.

Hyperparameter	Type of distribution	Value set or Range
Dropout rate	Uniform float	[0, 0.5]
Activation function	Categorical	{Linear, Sigmoid, Tanh, ReLU}
Optimizer	Categorical	{SGD, RMSprop, Adam}
Learning rate	Uniform float	[0.0001, 0.1]

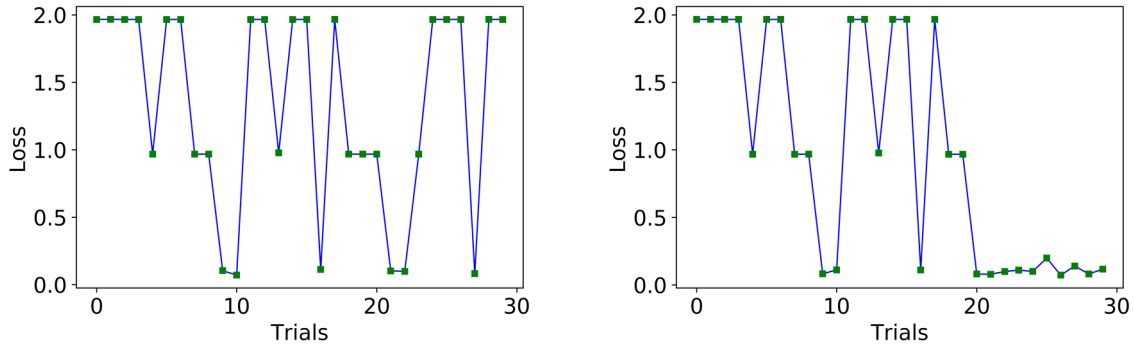
Fig. 14 shows the comparison of the TPE and RS hyperparameter searches over 30 trials for SG 1

of plant No. 1. The corresponding training loss is also given in Fig. 15. In particular, the TPE algorithm uses the first 20 startup trials for initializing the distributions of the *good* and *bad* hyperparameter sets, as mentioned in Section 4.2.2. This initialization process is performed by employing a standard RS. Therefore, in Figs. 14 and 15, we can observe a similar performance between TPE and RS in both hyperparameter searching and their obtained training losses during the first 20 trials. However, the performance of TPE is quickly improved after the initialization. It much more focuses on the good hyperparameter configurations which was found in the previous trials, leading to faster converge and lower training loss than RS within 30 trials.



(a) Dropout rate (b) Activation function (c) Optimizer (d) Learning rate

Fig. 14. Hyperparameters tuning process over 30 trials by TPE (top Figures) and RS (bottom Figures) for SG 1 of plant No.1.



(a) RS (b) TPE

Fig. 15. Training loss versus trials of TPE and RS for SG 1 of plant No. 1.

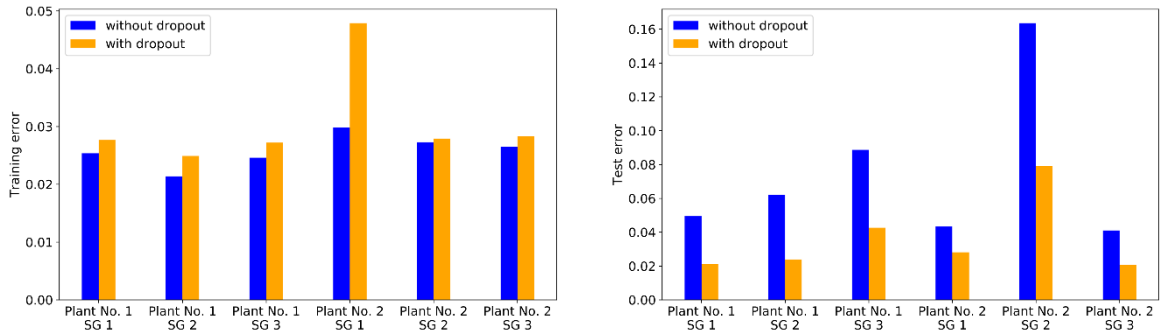
In Table 3, we show the performance comparison between TPE and RS, in terms of their obtained best training loss for all SGs. The results obviously show that the optimal configurations found by TPE generally outperform the best ones found by RS in the considered case studies. Thus, the optimal hyperparameter configurations found by TPE are used for prediction in the next stage.

Table 3
The best training loss obtained by TPE and RS in hyperparameter tuning for all SGs.

Plant	No. 1			No. 2		
SG	1	2	3	1	2	3
Random search	0.0487	0.0479	0.0307	0.0358	0.0321	0.0343
TPE	0.0440	0.0370	0.0319	0.0350	0.0314	0.0270

6.2. Dropout regularization

In this section, a comparison is carried out between the proposed prediction model and a model with the same architecture but trained without dropout. The other hyperparameters are kept identical between the two models, as described in Section 6.1. The probability of the used dropout is automatically optimized by TPE. We employ all the six SG datasets to comprehensively evaluate dropout during both the training and test phases in terms of RMSE. The comparative results are shown in Fig. 16. The result shows that the prediction model trained without dropout has lower training errors but much higher test errors, which may be an indication of the presence of overfitting. In contrast, the dropout model significantly reduces the overfitting problem with lower test errors for all the datasets. The average error reduction of the dropout model is 51.91%, which strongly indicates the efficacy of dropout in reducing overfitting and improving the prediction performance of the neural network.



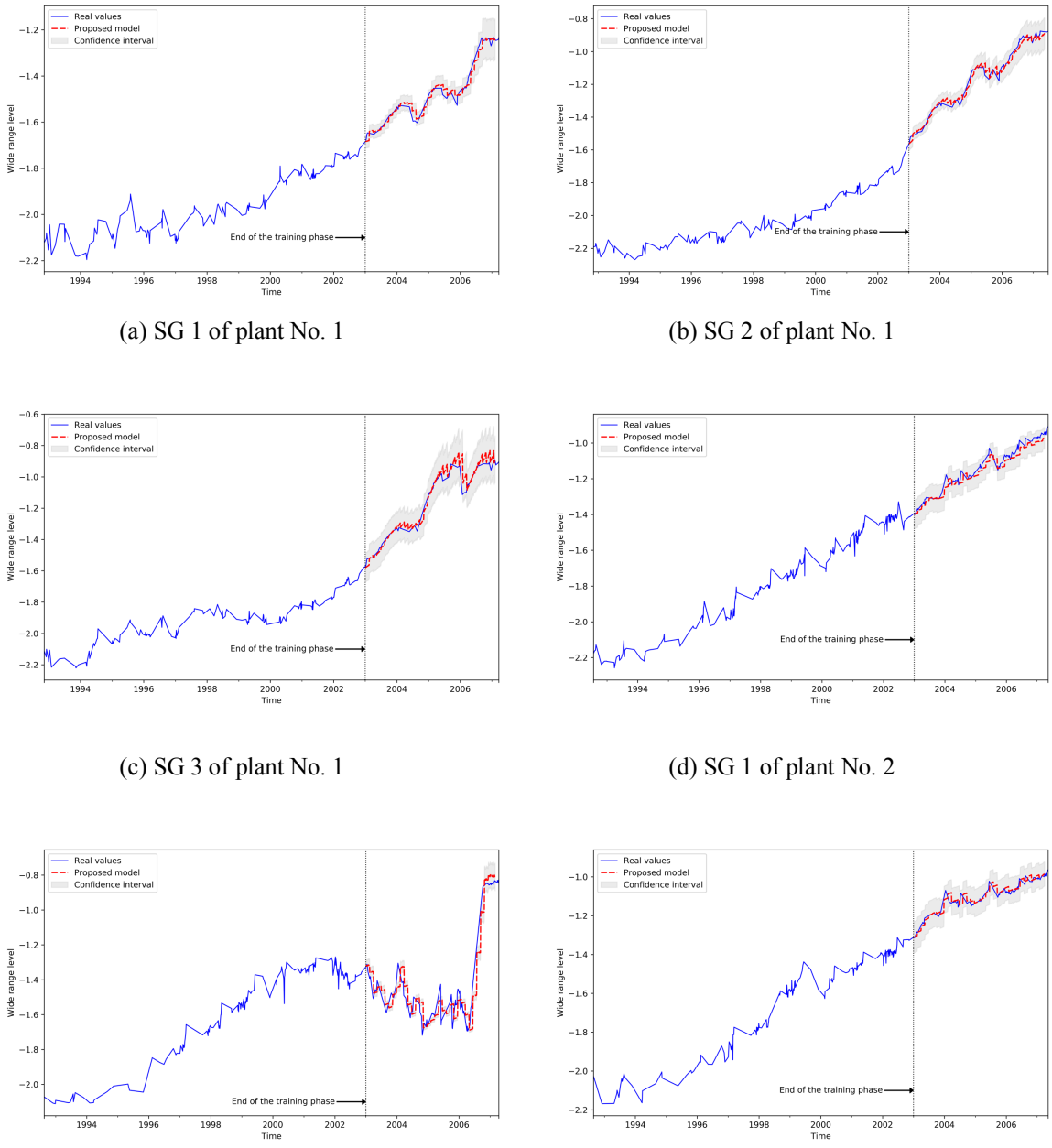
(a) Training phase (b) Test phase

Fig. 16. Training and test errors for the network architecture trained without and with dropout.

6.3. Performance evaluation

The WRL measurements of the six SGs are used for validating the developed prediction model for multi-step ahead prediction. It is important to remind that the prediction horizon used in this study is 15 steps ahead, which equals 45 operating days of the SGs. After the training is finished, the prediction model is used to continuously predict 15-step ahead in the next 5 years. Fig. 17 illustrates the prediction results of the proposed model for all SGs. The predicted values are shown as the dashed line, whereas

the solid line depicts the actual observations. The 95% confidence interval of the predictions, obtained via MC simulations, is depicted as the grey region. The results show that the proposed model is able to keep track with the changes of the WRL data while achieving accurate predictions, which are very close to the actual data for all SGs. Moreover, the 95% confidence bounds of the predictions are narrow and close to the target values, indicating predictions with a high precision. In industrial applications, these results are of crucial importance for accurately estimating the equipment RUL.



(a) SG 1 of plant No. 1

(b) SG 2 of plant No. 1

(c) SG 3 of plant No. 1

(d) SG 1 of plant No. 2

(e) SG 2 of plant No. 2

(f) SG 3 of plant No. 2

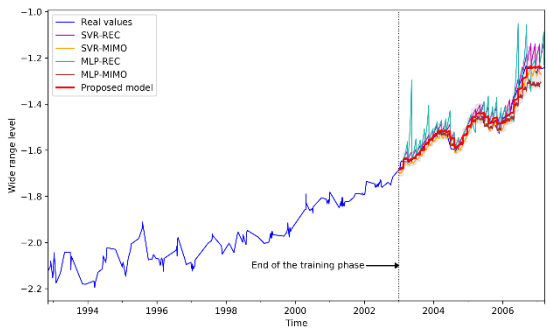
Fig. 17. Multi-step ahead prediction results by the proposed model for all SGs.

The prediction results obtained by the proposed model are, then, evaluated with respect to the four benchmark models, i.e. SVR-REC, SVR-MIMO, MLP-REC and MLP-MIMO, in terms of prediction accuracy. For a fair comparison, the hyperparameters of the compared models are optimized by using TPE with 30 trials. The details of the hyperparameter search spaces of the compared models are shown in Table 4.

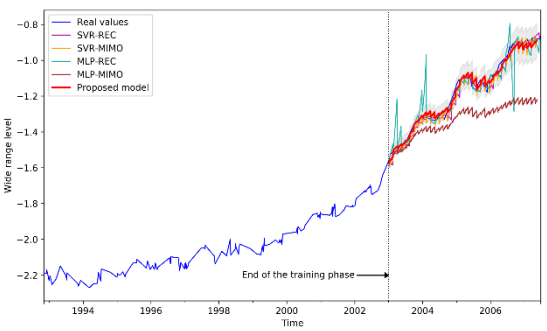
Table 4
Hyperparameters of the benchmark models.

Model	Hyperparameter	Value set or Range
SVR (including SVR-REC and SVR-MIMO)	Kernel function	{Linear, RBF, Poly, Sigmoid}
	Degree (of the polynomial kernel function)	[2, 4]
	Regularization parameter (C)	[0.01, 100]
	Kernel coefficient (gamma)	[0.01, 10]
MLP (including MLP-REC and MLP-MIMO)	Hidden layer size	[1, 5]
	Activation function	{Logistic, Tanh, ReLU}
	Optimizer	{LBFGS, SGD, Adam}
	Learning rate	{Constant, Invscaling, Adaptive}
	Regularization parameter (alpha)	[0.0001, 0.01]

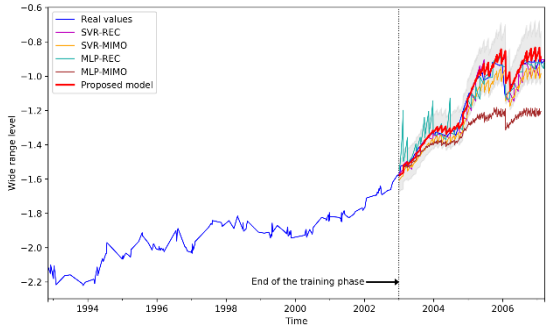
The comparative results of the proposed model and the four benchmark models for multi-step ahead predictions are shown in Fig. 18. Table 5 summarizes the prediction results in terms of the three accuracy indicators for different SG data. As can be seen in Fig. 18 and Table 5 (values in bold), the proposed prediction model outperforms the four other benchmark models and achieves higher accuracy for all SGs. The results indicate the accurate and efficient learning of the proposed prediction model for the long-term dependencies of the SG data.



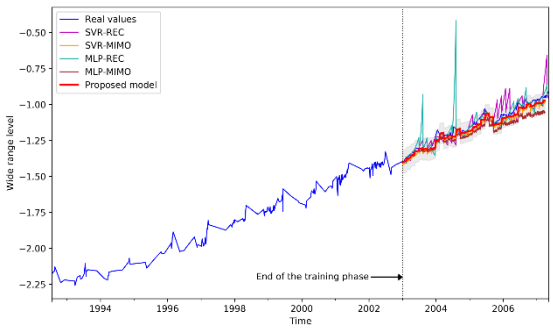
(a) SG 1 of plant No. 1



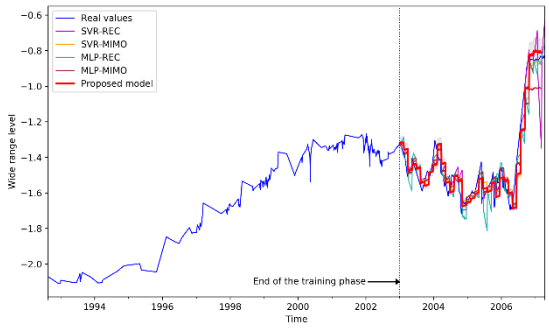
(b) SG 2 of plant No. 1



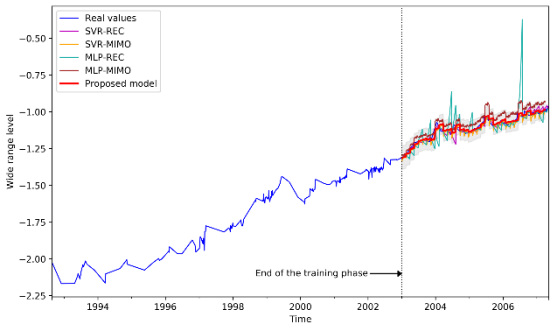
(c) SG 3 of plant No. 1



(d) SG 1 of plant No. 2



(e) SG 2 of plant No. 2



(f) SG 3 of plant No. 2

Fig. 18. Multi-step ahead predictions using different models for all SGs.

Table 5
Comparison of the prediction performance in multi-step ahead predictions for all SGs.

	Method	SG 1			SG 2			SG 3		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
Plant No. 1	SVR-REC	0.0382	2.0775	15.2484	0.0333	2.1970	10.2221	0.0508	3.3173	12.1521
	SVR-MIMO	0.0283	1.6511	12.2085	0.0331	2.3878	10.8281	0.0640	5.1793	19.2863
	MLP-REC	0.0597	2.7824	20.8682	0.0656	3.3349	15.3398	0.0577	3.3597	13.4886
	MLP-MIMO	0.0339	1.7074	11.8887	0.1888	15.1554	62.2885	0.1867	15.2453	52.7662
	Proposed model	**0.0212**	**1.0950**	**8.6166**	**0.0239**	**1.6973**	**5.8214**	**0.0426**	**2.7230**	**4.0846**
Plant No. 2	SVR-REC	0.0572	3.6462	12.8555	0.0906	4.7909	**7.0354**	0.0242	1.5005	**6.1132**
	SVR-MIMO	0.0401	3.1774	11.8732	0.0849	4.7575	7.4570	0.0247	1.7842	7.5984
	MLP-REC	0.0751	3.0309	11.2367	0.0862	4.7819	7.4403	0.0734	3.4535	14.2511
	MLP-MIMO	0.0607	4.8530	17.4741	0.0888	5.4251	7.5992	0.0499	4.1168	17.2689
	Proposed model	**0.0281**	**2.0117**	**8.6455**	**0.0791**	**4.4033**	9.3923	**0.0206**	**1.3992**	7.9604

The average computational time of training the proposed prediction model is 3.2 hours, on a GPGPU node comprising 2 Intel Xeon CPU E5-2695 (24 cores at 2.40 Hz with 32 GB of RAM) and 2

Nvidia Tesla K40m graphic cards (with 12 GB of GRAM). It is important to note that SG data used in this paper were recorded at an interval of 3 days. After being trained, the proposed model can be used to perform a 15-step ahead prediction, which is equivalent to 45 operating days ahead of the SGs. Due to this reason, the proposed prediction framework can be applied for a real-time time series prediction of the considered application.

The authors have tested the proposed framework on data from several nuclear power plants, with satisfactory results. Unfortunately, for industrial confidentiality, the data cannot be disclosed and shared.

7. Conclusion and future work

This paper presents an original multi-step ahead prediction framework for PHM applications. The framework integrates three consecutive steps: (1) data preprocessing, (2) adaptive model building and (3) multi-step ahead prediction. Initially, the problems of abnormal outliers and missing data samples are addressed by employing two preprocessing techniques: Isolation Forest and local regression. Then, a LSTM RNN is constructed for making predictions over a long-term horizon, in which the network hyperparameters are automatically optimized by a TPE algorithm. A dropout regularization and a cross-validation techniques are applied to address the overfitting problem during the training phase. Finally, the performance of the proposed model is evaluated for multi-step ahead predictions with a MIMO prediction strategy employed. A MC dropout is adopted to quantify the prediction uncertainty.

The proposed multi-step ahead prediction framework can be used for the predictions of time series of NPP operating parameters. A case study concerning the real WRL measurements of SGs which were acquired from different NPPs in France over a period of 16 years is carried out for validating the proposed framework. The experimental results show that the developed prediction framework is able to adaptively estimate the optimal setting for its architecture and capture the underlying long-term dependencies inherent in the given data, for achieving accurate predictions over a long horizon, up to 45 days ahead, outperforming conventional prediction approaches.

However, for the application of NPP SGs used in this study, sufficient information and data for performing a multivariate prediction are not provided, e.g. the information of the interdependency between measured variables and degradations (or failures), the interdependency within the variables, and the maintenance reports of the NPP SGs. Future research will be performed to develop a multivariate time series prediction model and integrate the proposed framework within a RUL estimation task for PHM and predictive maintenance.

Acknowledgement

The authors would like to thank the PRISME department of Électricité de France (EDF) R&D for

providing the data used in this paper.

References

[1] E. Zio, "Prognostics and Health Management of Industrial Equipment," in *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, S. Kadry, Ed. IGI Global, 2012, pp. 333–356.

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

[3] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16–18, pp. 2861–2869, Oct. 2007.

[4] A. S. Weigend, *Time Series Prediction: Forecasting The Future And Understanding The Past*. New York: Routledge, 1994.

[5] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications," *Mech. Syst. Signal Process.*, 2014.

[6] H. P. Nguyen, J. Liu, and E. Zio, "Ensemble of Models for Fatigue Crack Growth Prognostics," *IEEE Access*, vol. 7, pp. 49527–49537, 2019.

[7] H. P. Nguyen, J. Liu, and E. Zio, "Dynamic-weighted ensemble for fatigue crack degradation state prediction," *Eng. Fract. Mech.*, 2018.

[8] Y. Hu, P. Baraldi, F. Di Maio, and E. Zio, "A particle filtering and kernel smoothing-based approach for new design component prognostics," *Reliab. Eng. Syst. Saf.*, vol. 134, pp. 19–31, 2014.

[9] L. Liao and F. Köttig, "A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction," *Appl. Soft Comput. J.*, 2016.

[10] H. Nguyen, W. Fauriat, E. Zio, and J. Liu, "A Data-Driven Approach for Predicting the Remaining Useful Life of Steam Generators," in *2018 3rd International Conference on System Reliability and Safety (ICSRS)*, 2018, pp. 255–260.

[11] Y. Xiao and L. Feng, "A novel neural-network approach of analog fault diagnosis based on kernel discriminant analysis and particle swarm optimization," *Appl. Soft Comput. J.*, 2012.

[12] Y. Lin, X. Li, and Y. Hu, "Deep diagnostics and prognostics: An integrated hierarchical learning framework in PHM applications," *Appl. Soft Comput. J.*, 2018.

[13] J. Yu, "A hybrid feature selection scheme and self-organizing map model for machine health assessment," *Appl. Soft Comput. J.*, 2011.

[14] J. Sanz, R. Perera, and C. Huerta, "Gear dynamics monitoring using discrete wavelet transformation and multi-layer perceptron neural networks," *Appl. Soft Comput. J.*, 2012.

[15] J. Liu, V. Vitelli, E. Zio, and R. Seraoui, "A Novel Dynamic-Weighted Probabilistic Support Vector Regression-Based Ensemble for Prognostics of Time Series Data," *IEEE Trans. Reliab.*, 2015.

[16] A. Rai and S. H. Upadhyay, "An integrated approach to bearing prognostics based on EEMD-multi feature extraction, Gaussian mixture models and Jensen-Rényi divergence," *Appl. Soft Comput. J.*, 2018.

[17] L. L. Li, Z. F. Liu, M. L. Tseng, and A. S. F. Chiu, "Enhancing the Lithium-ion battery life predictability using a hybrid method," *Appl. Soft Comput. J.*, 2019.

[18] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 International Conference on Prognostics and Health Management, PHM 2008*, 2008.

[19] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An Evolutionary Algorithm that Constructs Recurrent Neural Networks," *IEEE Trans. Neural Networks*, 1994.

[20] Z. Li, D. Wu, C. Hu, and J. Terpenny, "An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction," *Reliab. Eng. Syst. Saf.*, 2019.

[21] J. Chen, H. Jing, Y. Chang, and Q. Liu, "Gated recurrent unit based recurrent neural network for remaining

useful life prediction of nonlinear deterioration process," *Reliab. Eng. Syst. Saf.*, 2019.

[22] E. Zio, M. Broggi, L. R. Golea, and N. Pedroni, "Failure and reliability predictions by infinite impulse response locally recurrent neural networks," *Chem. Eng. Trans.*, 2012.

[23] A. E. R. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Appl. Soft Comput. J.*, 2018.

[24] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Trans. Veh. Technol.*, 2018.

[25] J. Liu, A. Saxena, K. Goebel, B. Saha, and W. Wang, "An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries," *Natl. Aeronaut. Sp. Adm. Moffett F. Ca Ames Res. Cent.*, 2010.

[26] D. Liu, W. Xie, H. Liao, and Y. Peng, "An integrated probabilistic approach to lithium-ion battery remaining useful life estimation," *IEEE Trans. Instrum. Meas.*, 2015.

[27] Y. Cheng, H. Zhu, J. Wu, and X. Shao, "Machine Health Monitoring Using Adaptive Kernel Spectral Clustering and Deep Long Short-Term Memory Recurrent Neural Networks," *IEEE Trans. Ind. Informatics*, 2019.

[28] B. Zhang, S. Zhang, and W. Li, "Bearing performance degradation assessment using long short-term memory recurrent network," *Comput. Ind.*, 2019.

[29] L. Ren, X. Cheng, X. Wang, J. Cui, and L. Zhang, "Multi-scale Dense Gate Recurrent Unit Networks for bearing remaining useful life prediction," *Futur. Gener. Comput. Syst.*, 2019.

[30] A. Malhi, R. Yan, and R. X. Gao, "Prognosis of defect propagation based on recurrent neural networks," *IEEE Trans. Instrum. Meas.*, 2011.

[31] R. Ma, T. Yang, E. Breaz, Z. Li, P. Briois, and F. Gao, "Data-driven proton exchange membrane fuel cell degradation predication through deep learning method," *Appl. Energy*, 2018.

[32] K. Javed, R. Gouriveau, N. Zerhouni, and D. Hissel, "Prognostics of Proton Exchange Membrane Fuel Cells stack using an ensemble of constraints based connectionist networks," *J. Power Sources*, 2016.

[33] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017.

[34] N. A. Gershenfeld, A. S. Weigend, N. A. Gershenfeld, and A. S. Weigend, "The future of time series," *Time Ser. Predict. Forecast. Futur. Underst. Past*, 1993.

[35] G. U. Yule, "On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, 1927.

[36] E. Erdem and J. Shi, "ARMA based approaches for forecasting the tuple of wind speed and direction," *Appl. Energy*, 2011.

[37] B. Doucoure, K. Agbossou, and A. Cardenas, "Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data," *Renew. Energy*, 2016.

[38] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Renew. Energy*, 2009.

[39] C. D. Zuluaga, M. A. Álvarez, and E. Giraldo, "Short-term wind speed prediction based on robust Kalman filtering: An experimental comparison," *Appl. Energy*, 2015.

[40] 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.

[41] I. Kaushik and S. Singh, "Seasonal ARIMA Model for Forecasting of Monthly Rainfall and Temperature," *J. Environ. Res. Dev.*, 2008.

[42] M. A. Mariño, J. C. Tracy, and S. A. Taghavi, "Forecasting of reference crop evapotranspiration," *Agric. Water Manag.*, 1993.

[43] M. Ohyver and H. Pudjihastuti, "Arima Model for Forecasting the Price of Medium Quality Rice to Anticipate Price Fluctuations," in *Procedia Computer Science*, 2018.

[44] Q. Wang, X. Song, and R. Li, "A novel hybridization of nonlinear grey model and linear ARIMA residual correction for forecasting U.S. shale oil production," *Energy*, 2018.

[45] M. Qin, Z. Li, and Z. Du, "Red tide time series forecasting by combining ARIMA and deep belief network," *Knowledge-Based Syst.*, 2017.

[46] J. Wang and Y. Li, "Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy," *Appl. Energy*, 2018.

[47] J. Berbić, E. Ocvirk, D. Carević, and G. Lončar, "Application of neural networks and support vector machine for significant wave height prediction," *Oceanologia*, 2017.

[48] B. Kamranzad, A. Etemad-Shahidi, and M. H. Kazeminezhad, "Wave height forecasting in Dayyer, the Persian Gulf," *Ocean Eng.*, 2011.

[49] J. Wang and Y. Li, "Short-Term Wind Speed Prediction Using Signal Preprocessing Technique and Evolutionary Support Vector Regression," *Neural Process. Lett.*, 2018.

[50] G. W. Chang, H. J. Lu, Y. R. Chang, and Y. D. Lee, "An improved neural network-based approach for short-term wind speed and power forecast," *Renew. Energy*, 2017.

[51] P. Jain and M. C. Deo, "Real-time wave forecasts off the western Indian coast," *Appl. Ocean Res.*, 2007.

[52] J. Wang and Y. Li, "An innovative hybrid approach for multi-step ahead wind speed prediction," *Appl. Soft Comput. J.*, 2019.

[53] G. Li and J. Shi, "On comparing three artificial neural networks for wind speed forecasting," *Appl. Energy*, 2010.

[54] J. P. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes, "Hybrid wavelet-PSO-ANFIS approach for short-term wind power forecasting in Portugal," *IEEE Trans. Sustain. Energy*, 2011.

[55] M. Özger and Z. Şen, "Prediction of wave parameters by using fuzzy logic approach," *Ocean Eng.*, 2007.

[56] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing*, 2014.

[57] H. zhi Wang, G. qiang Li, G. bing Wang, J. chun Peng, H. Jiang, and Y. tao Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Appl. Energy*, 2017.

[58] Y. Li, H. Wu, and H. Liu, "Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEWT reconstruction," *Energy Convers. Manag.*, 2018.

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

[60] 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.

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

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

[63] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," *arXiv e-prints*, p. arXiv:1409.2329, Sep. 2014.

[64] I. Sutskever, G. Hinton, A. Krizhevsky, and R. R. Salakhutdinov, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, 2014.

[65] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman, "Sex, mixability, and modularity," *Proc. Natl. Acad. Sci. U. S. A.*, 2010.

[66] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008.

[67] W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, and X. Fan, "A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting," *Energy Convers. Manag.*, 2017.

[68] S. R. Moreno and L. dos Santos Coelho, "Wind speed forecasting approach based on Singular Spectrum Analysis and Adaptive Neuro Fuzzy Inference System," *Renew. Energy*, 2018.

[69] X. Mi, H. Liu, and Y. Li, "Wind speed prediction model using singular spectrum analysis, empirical mode decomposition and convolutional support vector machine," *Energy Convers. Manag.*, 2019.

[70] T. Niu, J. Wang, K. Zhang, and P. Du, "Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy," *Renew. Energy*, 2018.

[71] J. Chen, G. Q. Zeng, W. Zhou, W. Du, and K. Di Lu, "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization," *Energy Convers. Manag.*, 2018.

[72] C. Tian, Y. Hao, and J. Hu, "A novel wind speed forecasting system based on hybrid data preprocessing and multi-objective optimization," *Appl. Energy*, 2018.

[73] Z. Yang and J. Wang, "A hybrid forecasting approach applied in wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm," *Energy*, vol. 160, pp. 87–100, Oct. 2018.

[74] M. Santhosh, C. Venkaiah, and D. M. Vinod Kumar, "Ensemble empirical mode decomposition based adaptive wavelet neural network method for wind speed prediction," *Energy Convers. Manag.*, 2018.

[75] J. Naik, P. Satapathy, and P. K. Dash, "Short-term wind speed and wind power prediction using hybrid empirical mode decomposition and kernel ridge regression," *Appl. Soft Comput. J.*, 2018.

[76] Z. Yang and J. Wang, "A combination forecasting approach applied in multistep wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm," *Appl. Energy*, vol. 160, pp. 87–100, Oct. 2018.

[77] Q. Han, F. Meng, T. Hu, and F. Chu, "Non-parametric hybrid models for wind speed forecasting," *Energy Convers. Manag.*, 2017.

[78] H. Liu, Z. Duan, F. ze Han, and Y. fei Li, "Big multi-step wind speed forecasting model based on secondary decomposition, ensemble method and error correction algorithm," *Energy Convers. Manag.*, 2018.

[79] M. A. Chitsazan, M. Sami Fadali, and A. M. Trzynadlowski, "Wind speed and wind direction forecasting using echo state network with nonlinear functions," *Renew. Energy*, 2019.

[80] D. Y. Hong, T. Y. Ji, M. S. Li, and Q. H. Wu, "Ultra-short-term forecast of wind speed and wind power based on morphological high frequency filter and double similarity search algorithm," *Int. J. Electr. Power Energy Syst.*, 2019.

[81] H. Liu, X. Mi, and Y. Li, "Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM," *Energy Convers. Manag.*, 2018.

[82] 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.

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

[84] J. Liu and E. Zio, "A SVR-based ensemble approach for drifting data streams with recurring patterns," *Appl. Soft Comput. J.*, 2016.

[85] Z. Qu, K. Zhang, W. Mao, J. Wang, C. Liu, and W. Zhang, "Research and application of ensemble forecasting based on a novel multi-objective optimization algorithm for wind-speed forecasting," *Energy Convers. Manag.*, 2017.

[86] J. Hu, J. Wang, and L. Xiao, "A hybrid approach based on the Gaussian process with t-observation model for short-term wind speed forecasts," *Renew. Energy*, 2017.

[87] L. Xiao, F. Qian, and W. Shao, "Multi-step wind speed forecasting based on a hybrid forecasting architecture and an improved bat algorithm," *Energy Convers. Manag.*, 2017.

[88] P. Du, J. Wang, Z. Guo, and W. Yang, "Research and application of a novel hybrid forecasting system based on multi-objective optimization for wind speed forecasting," *Energy Convers. Manag.*, 2017.

[89] J. Wang, P. Du, T. Niu, and W. Yang, "A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting," *Appl. Energy*, 2017.

[90] X. wei Mi, H. Liu, and Y. fei Li, "Wind speed forecasting method using wavelet, extreme learning machine

and outlier correction algorithm," *Energy Convers. Manag.*, 2017.

[91] D. Wang, H. Luo, O. Grunder, and Y. Lin, "Multi-step ahead wind speed forecasting using an improved wavelet neural network combining variational mode decomposition and phase space reconstruction," *Renew. Energy*, 2017.

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

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

[94] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012.

[95] 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.

[96] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *30th International Conference on Machine Learning, ICML 2013*, 2013.

[97] 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.

[98] D. Ramachandram, M. Lisicki, T. J. Shields, M. R. Amer, and G. W. Taylor, "Bayesian optimization on graph-structured search spaces: Optimizing deep multimodal fusion architectures," *Neurocomputing*, 2018.

[99] J. Bergstra and D. Cox, "Hyperparameter Optimization and Boosting for Classifying Facial Expressions: How good can a 'Null' Model be?," in *International Conference on Machine Learning*, 2013.

[100] L. F. Rodrigues, M. C. Naldi, and J. F. Mari, "Comparing convolutional neural networks and preprocessing techniques for HEp-2 cell classification in immunofluorescence images," *Comput. Biol. Med.*, Nov. 2019.

[101] S. F. Chevtchenko, R. F. Vale, V. Macario, and F. R. Cordeiro, "A convolutional neural network with feature fusion for real-time hand posture recognition," *Appl. Soft Comput. J.*, 2018.

[102] J. Lago, F. De Ridder, P. Vrancx, and B. De Schutter, "Forecasting day-ahead electricity prices in Europe: The importance of considering market integration," *Appl. Energy*, 2018.

[103] J. Lago, K. De Brabandere, F. De Ridder, and B. De Schutter, "Short-term forecasting of solar irradiance without local telemetry: A generalized model using satellite data," *Sol. Energy*, 2018.

[104] R. Mohammadi, Q. He, F. Ghofrani, A. Pathak, and A. Aref, "Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects," *Transp. Res. Part C Emerg. Technol.*, 2019.

[105] K. Kang and H. Ryu, "Predicting types of occupational accidents at construction sites in Korea using random forest model," *Saf. Sci.*, 2019.

[106] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *arXiv e-prints*, p. arXiv:1506.02142, Jun. 2015.

[107] G. Corredera, M. Alves-Vieira, and O. de Bouvier, "Fouling and TSP Blockage of Steam Generators on EDF Fleet : Identified Correlations with Secondary Water Chemistry and planned Remedies," in *International Conference, Water chemistry of nuclear reactor systems; 2008; Berlin, Germany*, 2008.

[108] S. Girard, *Physical and Statistical Models for Steam Generator Clogging Diagnosis*. Cham: Springer International Publishing, 2014.

[109] P. Luca, "Development of Unsupervised and Semi-Supervised Clustering-Based Methods for Degradation Assessment of Nuclear Power Plant Steam Generators," Politecnico Di Milano, 2018.

[110] 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.

Paper 4:

Hoang-Phuong Nguyen, Piero Baraldi, and Enrico Zio. (2020). A Method Based on Ensemble Empirical Mode Decomposition and Long Short-Term Memory Neural Network for Multi-Step Ahead Predictions of Nuclear Power Plant Signals. Submitted to Applied Energy in February, 2020.

A Method Based on Ensemble Empirical Mode Decomposition and Long Short-Term Memory Neural Network for Multi-Step Ahead Predictions of Nuclear Power Plant Signals

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

[a] *Chair on System Science and the Energetic Challenge, EDF Foundation, CentraleSupélec, Université Paris-Saclay, 9 rue Joliot-Curie, 91192 Gif-sur- Yvette, France*

[b] *Dipartimento di Energia, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy*

[c] *MINES ParisTech / PSL Université Paris, Centre de Recherche sur les Risques et les Crises (CRC), 1 rue Claude Daunesse, 06904 Sophia Antipolis, France*

[d] *Eminent Scholar, Department of Nuclear Engineering, Kyung Hee University, 26 Kyungheedae-ro, Hoegi-dong, Dongdaemun-gu, Seoul, South Korea*

Abstract

Multi-step ahead prediction can help decision makers improving maintenance planning and minimizing unexpected shutdowns in the nuclear industry. We address this problem by developing a hybrid prediction framework based on Ensemble Empirical Mode Decomposition (EEMD) and Long Short-Term Memory (LSTM) neural network. EEMD decomposes time series into a set of components, which allow effectively describing the system dynamics and therefore facilitates the prediction task. Then, LSTM neural network models are developed for predicting the multi-step ahead behavior of the individual EEMD components and the obtained predictions are aggregated to reconstruct the time series. A Tree-structured Parzen Estimator (TPE) algorithm is employed for hyperparameter optimization. The performance of the proposed method is validated by considering different long-term prediction horizons on a practical case study concerning time series data acquired from Reactor Coolant Pumps (RCPs) of Nuclear Power Plants (NPPs). The proposed method shows superior performances with respect to alternative prediction models

Keywords: Maintenance; Prognostics; Time series prediction; Multi-step ahead prediction; Ensemble empirical mode decomposition; Long short-term memory recurrent neural network; Nuclear power plant; Reactor coolant pump.

Nomenclature

Abbreviations

AM-FM	Amplitude-Modulated-Frequency-Modulated
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
EEMD	Ensemble Empirical Mode Decomposition

EI	Expected Improvement
EMD	Empirical Mode Decomposition
ES	Exponential Smoothing
ESN	Echo State Network
FNN	False Nearest Neighbor
IMF	Intrinsic Mode Function
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
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

$c_j(t)$	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*

$IMF_i(t)$	ith decomposed IMF at time t
$\overline{IMF}_i(t)$	ith ensemble IMF at time t
J	number of the EEMD noise realizations
$L_j(t)$	lower envelope of the decomposed component at the jth sifting iteration
$m_j(t)$	envelope mean of the decomposed component at the jth sifting iteration
N	number of samples
N_c	number of IMF components
N_{epoch}	number of LSTM training epochs
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
$r_i(t)$	ith decomposed residue at time t
$SD(j)$	stopping criterion value at the jth sifting iteration
t	time instance
$U_j(t)$	upper envelope of the decomposed component at the jth sifting iteration
w_t^j	jth 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 ith 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
$\theta*$	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 engineering 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 maintenance actions to be planned at the most opportune and convenient time instances.

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 [3]. 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 horizons of months [4]–[6]. 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 have a big influence on the degradation process in complex systems [3], [7]. Large prediction uncertainty in multi-step ahead predictions has limited the development of prognostics in nuclear applications to short-term prognostics, based on one-step ahead prediction [5], [6], [8]–[10]. In this context, this work develops a prognostic method specifically designed to deal with multi-step ahead predictions for practical applications in Nuclear Power Plants (NPPs).

In general, multi-step ahead prediction models can be classified as statistical or machine learning

approaches [11], [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]–[18], Artificial Neural Network (ANN) [11], [12], [19]–[21], Neuro-Fuzzy [7], [22], [23] and Recurrent Neural Network (RNN) [24]–[26]. 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 [27]. 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 [28]–[33], solar energy [34]–[36], air quality [37]–[39], stock market [40], [41], electricity and gas demand [42]–[44], and oil and petroleum production [45], [46].

A problem typically encountered in the development of multi-step ahead prediction models is the data complexity, which means that time series collected from real-world systems can contain 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 the degradation of the prediction performance as the time horizon of the prediction increases [47]. To address this issue, research on hybrid prediction models have been recently carried out to take advantage of the strength of using ensemble of different individual models. For example, Moshkbar-Bakhshayesh and Ghofrani [6] 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 [48] 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. [49] 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. Comprehensive analyses on hybrid approaches for the applications concerning multi-step

ahead prediction can be found in [47], [50], [51].

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. EEMD is a self-adaptive decomposition technique specifically tailored for analyzing nonlinear and nonstationary data [52]. 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. Also, we address the problem of automatic hyperparameter optimization by integrating a Tree-structured Parzen Estimator (TPE) algorithm within the LSTM models.

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

h) A novel multi-step ahead prediction method based on the combination of the EEMD decomposition algorithm and the LSTM neural networks is developed and applied to the prognostics of NPP components.

i) Automatic hyperparameter optimization is integrated within the LSTM models by using the TPE optimization algorithm and a *k*-fold cross-validation technique.

j) Multi-step ahead predictions are performed based on the MIMO strategy with three different long-term horizons.

A case study based on real time-series datasets acquired from NPPs is carried out to validate the proposed model. 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. [53] 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) [54]. 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) [55].

Algorithm 1. EMD decomposition process

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] [53]).

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:

5. Initialize the index $i = 1$ and residue $r_0(t) = X_t$.
6. Extract $IMF_i(t)$:
   j. 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.
   k. 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.
   l. Compute the envelope mean:

$$m_j(t) = \left[U_j(t) + L_j(t) \right] / 2 \tag{1}$$

   m. Generate the new component $c_{j+1}(t)$ of the next sifting iteration:

$$c_{j+1}(t) = c_j(t) - m_j(t) \tag{2}$$

   n. Compute the squared difference between two consecutive siftings as follows:

$$SD(j) = \sum_{l=1}^{t} \frac{\left| c_{j+1}(l) - c_j(l) \right|^2}{\left| c_j(l) \right|^2} \tag{3}$$

   o. 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).

7. Update the residue as follows:

$$r_i(t) = r_i(t) - IMF_i(t) \tag{4}$$

8. 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 [56]. To address these problems, EEMD has been developed [52]. 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 [54]. The EEMD algorithm is described in Algorithm 2.

Algorithm 2. EEMD decomposition process

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:

4. Generate the noisy time series:

$$X_t^j = X_t + w_t^j, \ j = 1, 2, ..., J \tag{6}$$

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

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

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

$$\overline{IMF_i}(t) = \frac{1}{J} \sum_{j=1}^{J} IMF_i^j(t) \tag{7}$$

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 [27]. An LSTM network consists of a chain of repeating memory modules (Fig. 1).

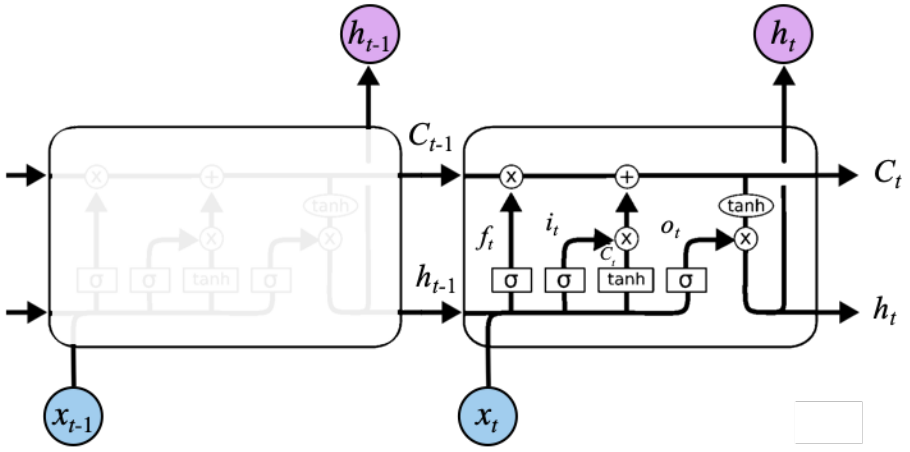


Fig. 1. The structure of LSTM repeating memory modules [57].

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 [58] 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 [59], [60]. In this study, we apply Tree-structured Parzen Estimator (TPE) [61], 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 [61].

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:

$$p(\theta \mid y) = \begin{cases} \Pr_G(\theta) & if\ y < y^* \\ \Pr_B(\theta) & if\ y \geq y^* \end{cases} \tag{16}$$

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

$$EI(\theta) = \frac{\Pr_G(\theta)}{\Pr_B(\theta)} \tag{17}$$

And the hyperparameter configuration θ^* 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.

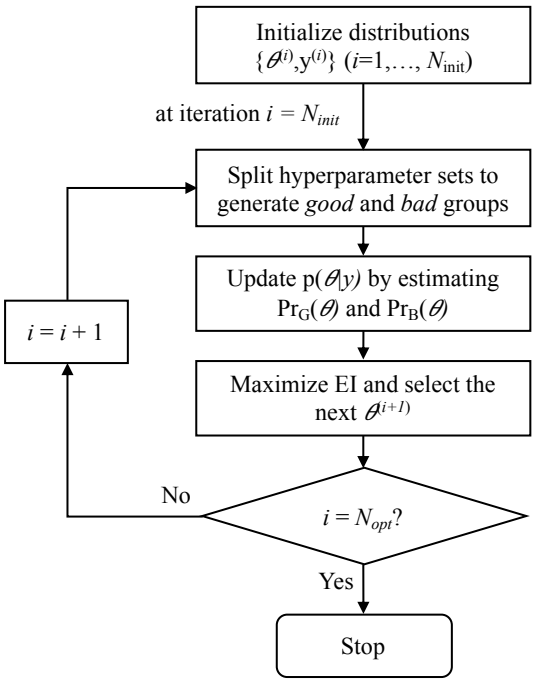


Fig. 2. Flowchart of the TPE optimization procedure.

3. Proposed 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.

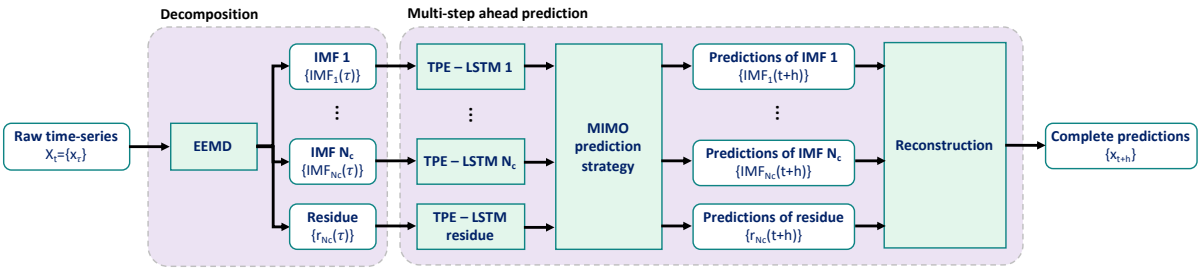
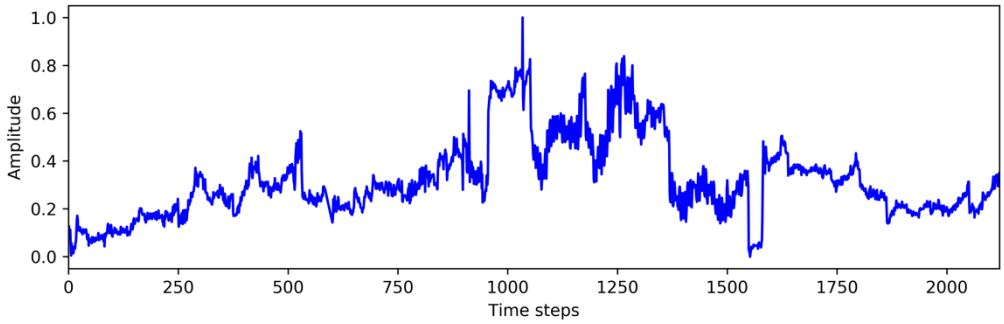


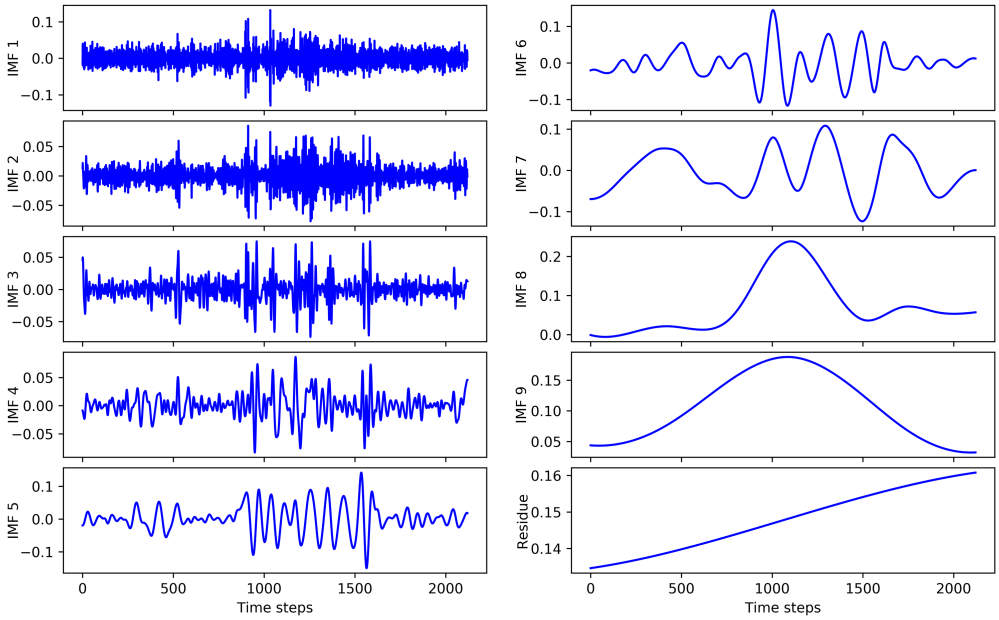
Fig. 3. Overview of the proposed multi-step ahead prediction method.

3.1. Decomposition of the original time series

EEMD is employed for decomposing the raw time series X_t into separate frequency components $\{IMF_i(t)\}$ $(i=1,2,...,N_c)$. The number of obtained IMFs N_c varies depending 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 by using EEMD.

3.2. Multi-step ahead prediction step

In the second stage of the proposed method, we develop a dedicated model for the multi-step ahead prediction 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 are performed and aggregated to obtain the multi-step ahead prediction 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. Hyperparameters of the LSTM models optimized by the TPE and possible settings of the proposed method considered.

Hyperparameter	Type of distribution	Possible setting
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

In general, there are three widely used strategies addressing multi-step ahead prediction: Recursive, Direct and MIMO [3], [7]. Each strategy is characterized by different trade-offs between accuracy and complexity. In this work, we employ the MIMO strategy, since using only one model with multiple outputs offers two main advantages: 1) avoiding error accumulation in long-term predictions; 2) reduction of the training computational cost [7]. The main difficulty to be addressed in developing a MIMO prediction model is the selection of the appropriate model configurations, which in this work is handled by TPE.

The MIMO approach (also known as the Parallel approach) aims at simultaneously predicting multiple future observations by using one single predictor [62]. It is illustrated in Fig. 5, where $\{f,[\theta]\}$ denote the predictor and its hyperparameters, respectively, and d the embedding dimension which is set by using the False Nearest Neighbor (FNN) approach [63].

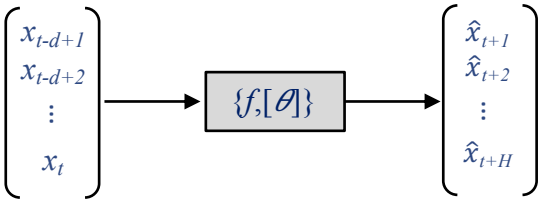
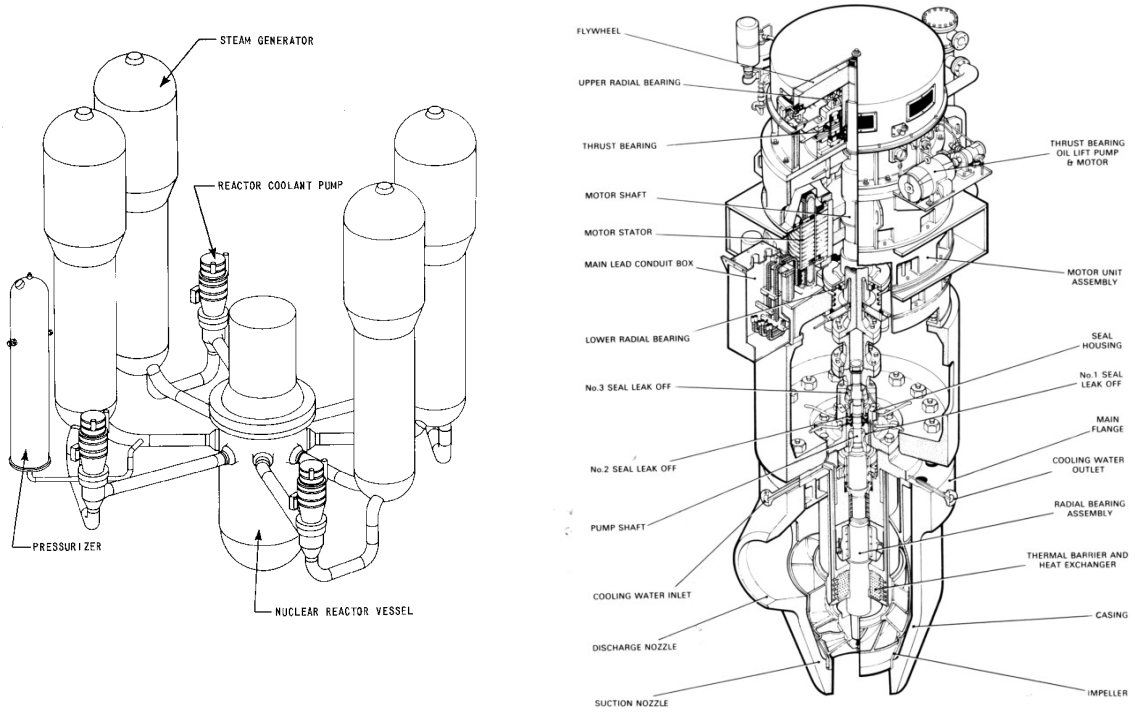


Fig. 5. Multi-step ahead prediction model with MIMO strategy.

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

We consider the Reactor Coolant Pumps (RCPs) of a NPP, which 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

Fig. 6. Detailed structures of the PWR Reactor Coolant System (RCS) and Reactor Coolant Pump (RCP). The image has been taken from [64].

One of the most vulnerable components of a RCP is the shaft seal system, which is composed of three mechanical seals located between the electric motor and the impeller, as shown in Fig. 6(b). The shaft seal system plays an important role in limiting the leakage of the primary circuit to the ambient environment, which are collected and routed to the seal leakoff system [65]. A failure of the shaft seal system can cause the loss of the reactor primary coolant, which can potentially lead to catastrophic consequences [66]. 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.

We consider five different scenarios of leakage flow from the first seal of the RCPs. The data have been acquired from real RCPs. The time series are measured at a 4-hour interval, starting from different time instances and for different durations, as shown in Fig. 7. The durations of the time series are listed in Table 2. For each scenario, the time series is divided into two subsets: the first 70% of the time series is used as training set for developing the prediction models and the latter 30% as test set for evaluating the model performance. For confidentiality reasons, the names of the NPPs are not mentioned and the time series data are normalized from 0 to 1. A different model is developed for each one of the five time series.

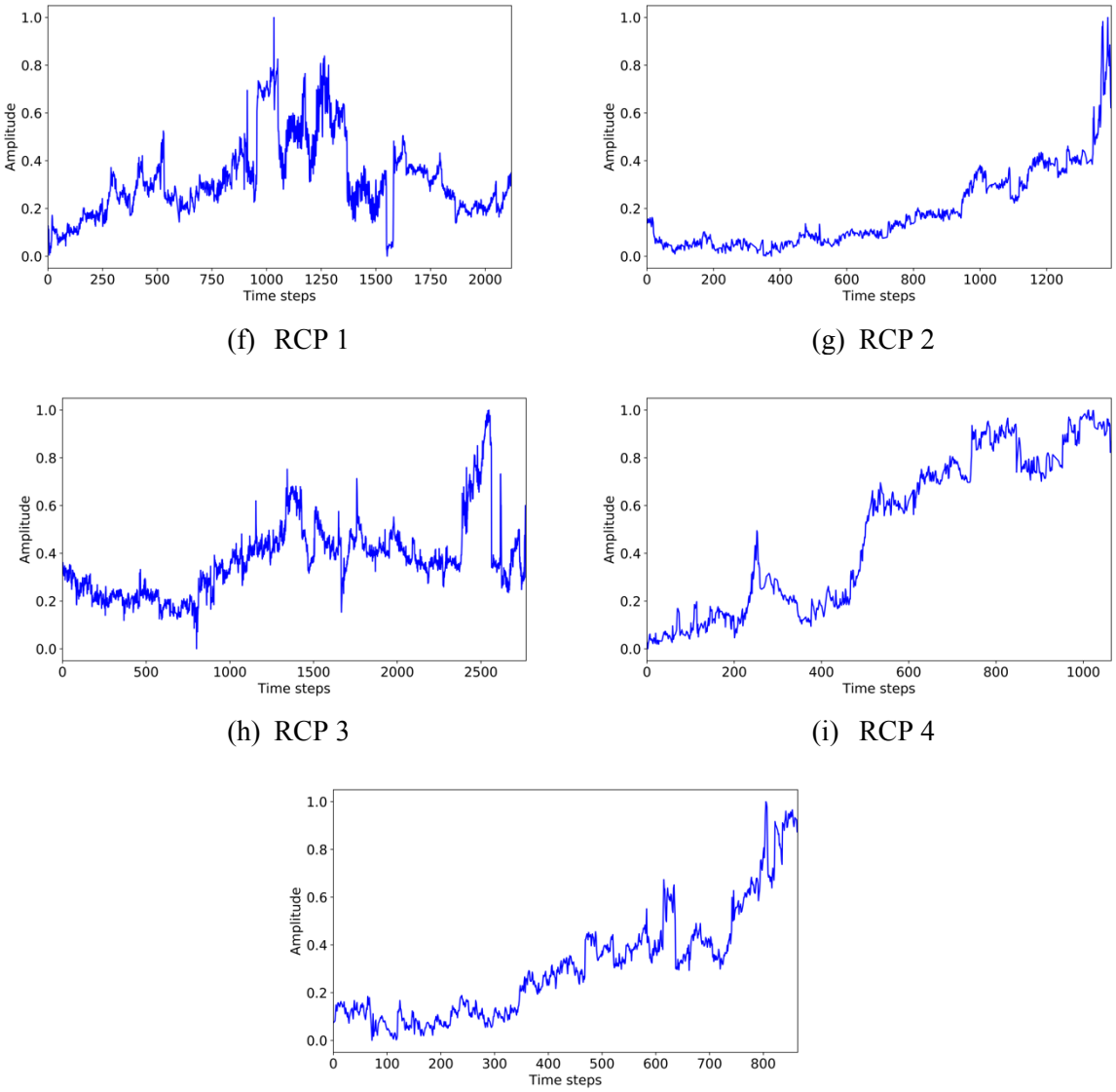


Fig. 7. The RCP leakage flow time series corresponding to the five RCP leakage scenarios considered in the case study.

Table 2. Number of observations available in the five time series.

RCP	1	2	3	4	5
Number of observations	2120	1394	2770	1064	864

5. Results and discussion

We carry out the validation of the proposed method with three performance evaluations considering different aspects of the methodology structure. Firstly, the effectiveness of the decomposition algorithm, i.e. EEMD, in improving the prediction performance is evaluated. Secondly, we employ a multivariate prediction model to validate the viability of our hybrid prediction framework, which integrates several univariate models. The final experiment focuses on the performance evaluation of the LSTM network, the central part of our method for multi-step ahead prediction. A prediction model based on Echo State Network (ESN) is employed for benchmarking purposes.

For each experiment, we consider three different prediction horizons to assess the prediction capability of the proposed method, including 6 steps (1 day), 12 steps (2 days) and 18 steps (3 days) ahead. The performance of the prediction models are measured with respect to three accuracy metrics, including RMSE (as stated in Section 3.2.1), Mean Absolute Percentage Error (MAPE) and Mean Absolute Scaled Error (MASE). The definitions of MAPE and MASE are given as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{x}_i - x_i}{x_i}\right| \times 100\%, \tag{18}$$

$$MASE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\hat{x}_i - x_i|}{\frac{1}{N-1}\sum_{j=2}^{N}|x_j - x_i|}\right), \tag{19}$$

where N is the number of test observations, x and $\hat{x}$ are the observed and predicted values, respectively. For the computational point of view, all of the experiments are implemented on 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).

5.1. Effectiveness of applying the decomposition technique for multi-step ahead prediction

In order to validate the EEMD, we employ a comparative model which is obtained by removing the EEMD module from the proposed method, as illustrated in Fig. 8. In this comparative model, the original time series are directly fed to the LSTM prediction model, with the hyperparameters optimized by TPE, as described in Section 3.2.1. It is important to note that the LSTM model is constructed with two LSTM layers consisting of 64 neurons for each layer. The predictions with the three horizons are performed using the MIMO strategy for all of the time series scenarios.

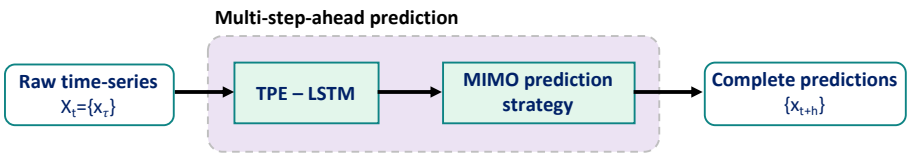
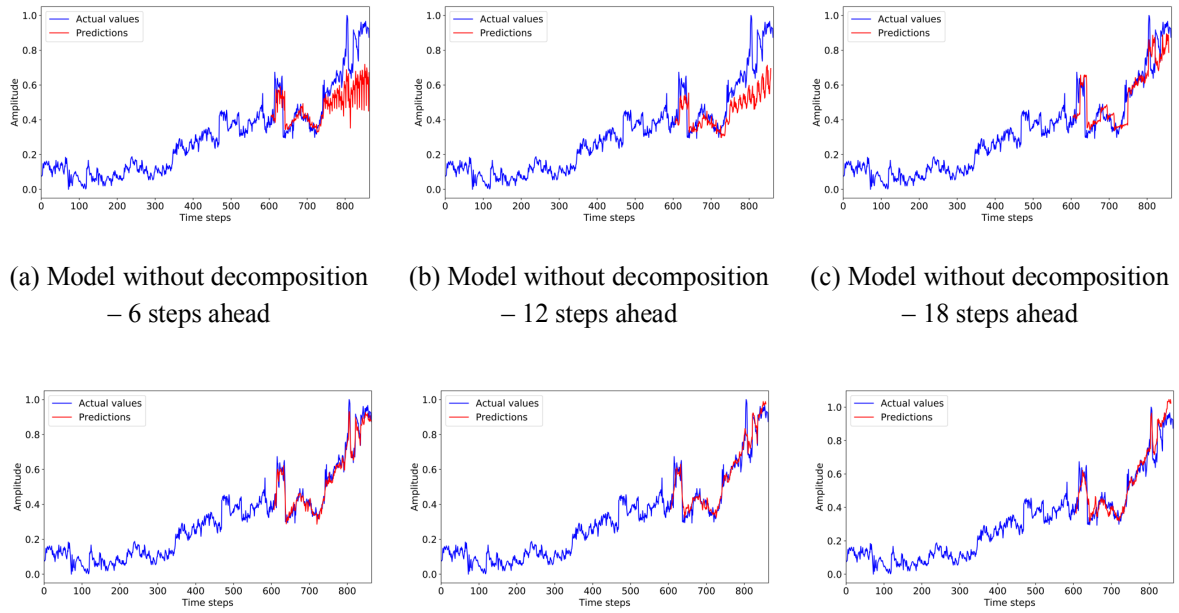


Fig. 8. Compared prediction model without using EEMD.

Fig. 9 depicts the prediction results for different horizons obtained by the comparative model (Fig. 9(a) – (c)) and the proposed model (Fig. 9(d) – (f)) for the RCP 5 time series. In each sub-figure, the predicted values are shown as the red solid line, whereas the blue line depicts the actual observations. Fig. 9 shows that the predictions of the proposed method are highly accurate and close to the actual values, whereas those of the comparative model are not so accurate in all of the three cases of the prediction horizon. The overall comparison results for all of the time series scenarios are summarized in Table 3. The more accurate results (the lower values of the metrics) are highlighted in bold. The results show that the proposed method outperforms the comparative model with a significant improvement in the prediction accuracy, implying the important contribution of the EEMD module to the proposed method for dealing with multi-step ahead predictions.



(a) Model without decomposition – 6 steps ahead

(b) Model without decomposition – 12 steps ahead

(c) Model without decomposition – 18 steps ahead

(d) Proposed model – 6 steps ahead

(e) Proposed model – 12 steps ahead

(f) Proposed model – 18 steps ahead

Fig. 9. Results of the multi-step ahead predictions for RCP 5.

Table 3. Comparison results of the proposed model and the model without using the EEMD decomposition.

Data scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Without decomposition	0.0405	13.8939	1.6168	0.0608	30.1245	2.3199	0.0667	30.4937	2.6540
	Proposed model	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Without decomposition	0.0776	11.6117	3.5690	0.0897	18.9838	5.2261	0.0893	16.2510	4.5966
	Proposed model	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Without decomposition	0.0627	7.9651	1.7586	0.0868	11.1782	2.5560	0.1081	14.2730	3.6001
	Proposed model	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Without decomposition	0.0568	5.4109	3.1283	0.0730	6.9583	4.1817	0.0891	8.3991	4.9783
	Proposed model	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Without decomposition	0.1583	16.7301	4.1357	0.1651	18.9645	4.5333	0.0988	12.9969	2.5915
	Proposed model	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**

5.2. Univariate model versus multivariate model

This experiment focuses on the evaluation of the use of the proposed hybrid framework, in which several univariate prediction models are developed for all of the decomposed components obtained from EEMD, as mentioned in Section 3.2. We employ a prediction model based on a multivariate LSTM network for comparison purposes, as illustrated in Fig. 10. Specifically, the multivariate LSTM model uses all of the decomposed IMFs as the model inputs and performs predictions using the MIMO strategy. The TPE algorithm is used to select the optimal model hyperparameters.

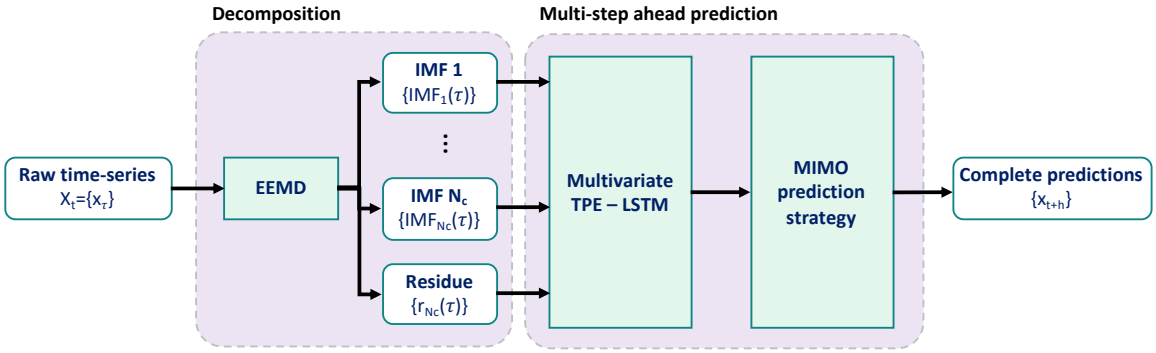
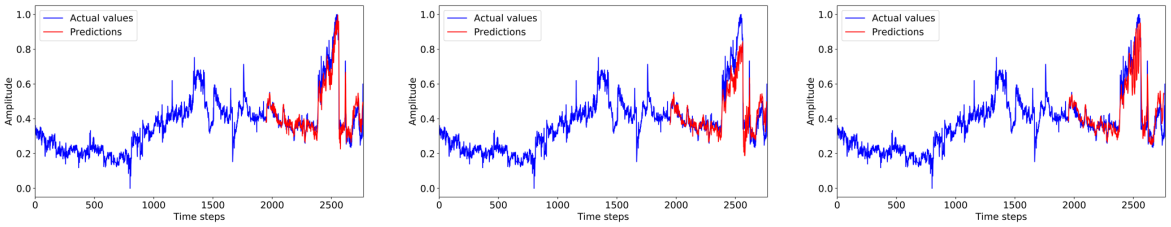


Fig. 10. Compared prediction model using a multivariate LSTM network.

The prediction results obtained by the proposed and comparative models for the RCP 3 and 4 scenarios are illustrated in Figs. 11 and 12, respectively. Table 4 summarizes the performance comparison of the two models for all of the data scenarios. As can be seen in Fig. 12, the prediction results of the multivariate LSTM model seem quite inaccurate, particularly in Fig. 12(c) with a lot of noisy spikes in the predictions. This can be explained by the fact that the data trend of the RCP 4 time series is complicated, strongly nonlinear and nonstationary, making it difficult for a single model to achieve good predictions. On the contrary, the prediction results of the proposed method, Figs. 12(d) –

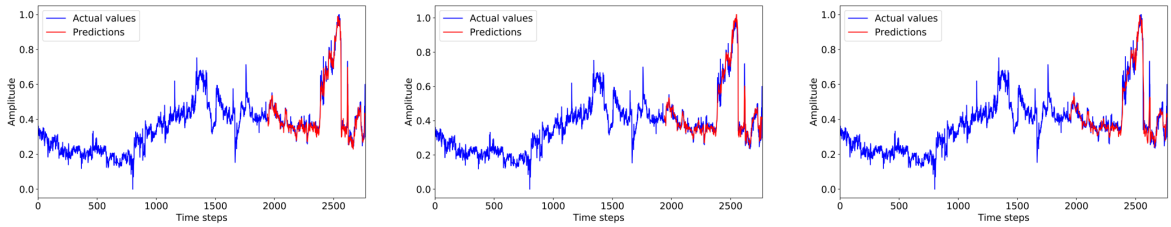
(f), are remarkably accurate. In Table 4, it appears that the hybrid framework provides better results for the whole tests on the five data scenarios than the multivariate model. This indicates that using the hybrid framework integrating univariate prediction models can achieve more accurate multi-step ahead predictions.



(a) Multivariate LSTM model – 6 steps ahead

(b) Multivariate LSTM model – 12 steps ahead

(c) Multivariate LSTM model – 18 steps ahead

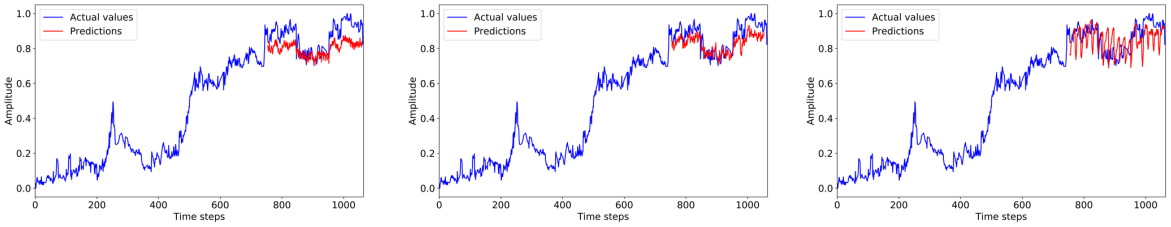


(d) Proposed model – 6 steps ahead

(e) Proposed model – 12 steps ahead

(f) Proposed model – 18 steps ahead

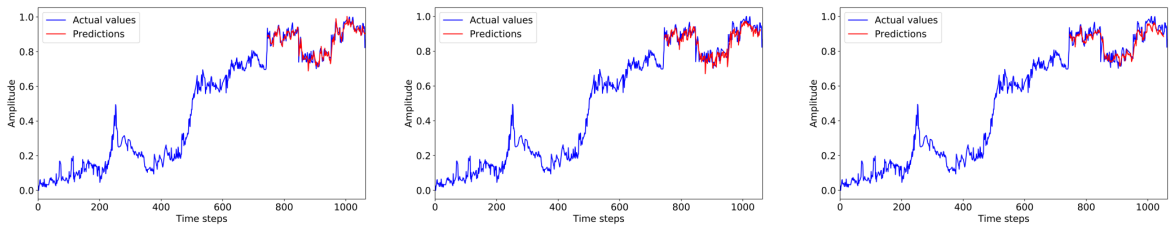
Fig. 11. Results of the multi-step ahead predictions for RCP 3.



(a) Multivariate LSTM model – 6 steps ahead

(b) Multivariate LSTM model – 12 steps ahead

(c) Multivariate LSTM model – 18 steps ahead



(d) Proposed model – 6 steps ahead

(e) Proposed model – 12 steps ahead

(f) Proposed model – 18 steps ahead

Fig. 12. Results of the multi-step ahead predictions for RCP 4.

Table 4. Comparison results of the proposed model and the multivariate LSTM model.

Data scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	Multivariate LSTM	0.0249	9.3767	1.3214	0.0386	20.3446	1.9242	0.0455	17.8378	2.4970
	Proposed model	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	Multivariate LSTM	0.0709	9.2070	2.8689	0.0483	6.3916	1.9585	0.0916	20.2789	6.2618
	Proposed model	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	Multivariate LSTM	0.0500	8.1171	1.8032	0.0747	10.8237	2.6785	0.0760	10.5087	2.5134
	Proposed model	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	Multivariate LSTM	0.0851	8.1530	5.0001	0.0607	5.6678	3.4487	0.0819	7.7580	4.5046
	Proposed model	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	Multivariate LSTM	0.1375	18.8527	3.8056	0.3038	27.1751	6.1985	0.1340	18.1811	3.6363
	Proposed model	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**

5.3. Performance evaluation of the LSTM prediction models

LSTM models play a fundamental role in our proposed method for learning complex data mappings, especially long-term dependency, and performing multi-step ahead predictions with the supports of the TPE optimization and the MIMO prediction strategy. In this Section, we validate the prediction performance of the LSTM models via a comparison with another widely used RNN called Echo State Network (ESN).

ESN is a RNN with a sparsely connected hidden layer [67]. 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 [67], [68].

The compared prediction model is developed by replacing the LSTM models with the ESN models, and the rest of the framework is kept unchanged, as illustrated in Fig. 13. In this framework, the TPE is used to optimize the two major hyperparameters of the ESN models, including the number of reservoirs and the spectral radius, as described in Table 5.

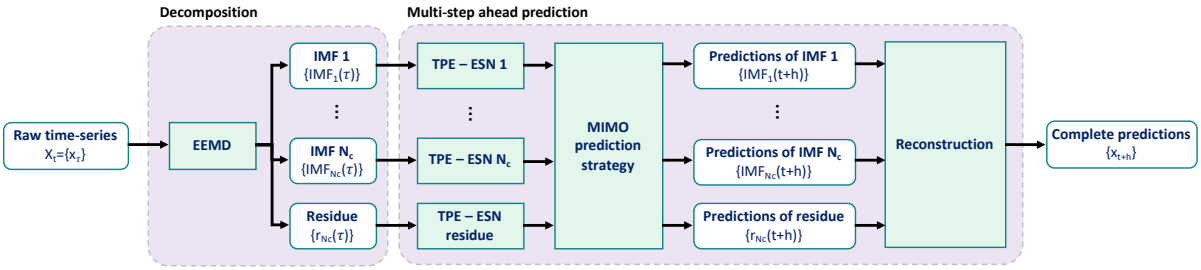
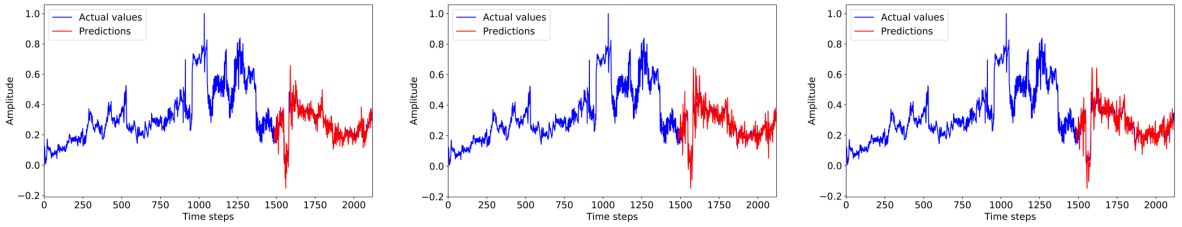


Fig. 13. Compared prediction model using the ESN RNNs.

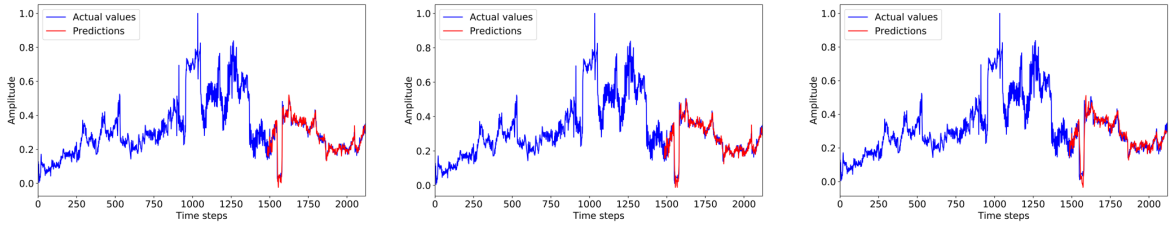
Table 5. Hyperparameters of the ESN model.

Hyperparameter	Type of distribution	Possible setting
Number of reservoirs	Uniform integer	[20, 500]
Spectral radius	Uniform float	[0.01, 1]

Figs. 14 and 15 show the results of the multi-step ahead predictions obtained by the ESN framework and the proposed framework for the RCP 1 and 2 scenarios, respectively. We summarize the overall performance comparison in Table 6. According to these results, the prediction framework using LSTMs consistently outperforms the ESN-based framework, achieving a greater accuracy for multi-step ahead predictions. Thus, LSTM is a more suitable choice for the development of a multi-step ahead prediction framework.

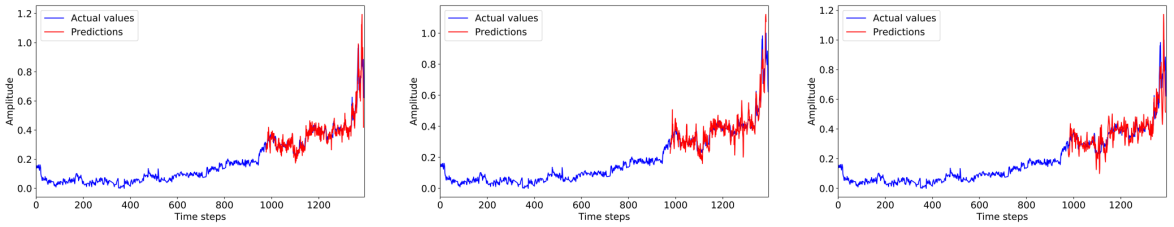


(a) ESN model – 6 steps ahead (b) ESN model – 12 steps ahead (c) ESN model – 18 steps ahead

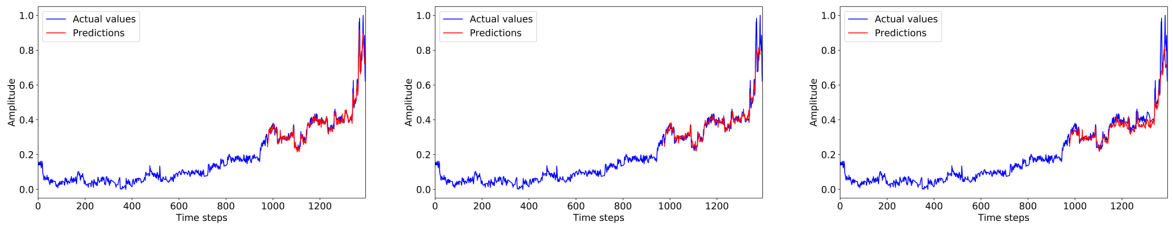


(d) Proposed model – 6 steps ahead (e) Proposed model – 12 steps ahead (f) Proposed model – 18 steps ahead

Fig. 14. Results of the multi-step ahead predictions for RCP 1.



(a) ESN model – 6 steps ahead (b) ESN model – 12 steps ahead (c) ESN model – 18 steps ahead



(d) Proposed model – 6 steps ahead (e) Proposed model – 12 steps ahead (f) Proposed model – 18 steps ahead

Fig. 15. Results of multi-step ahead predictions for RCP 2.

Table 6. Comparison results of the proposed model and the ESN-based prediction model.

Data scenario	Approach	6 steps ahead			12 steps ahead			18 steps ahead		
		RMSE	MAPE	MASE	RMSE	MAPE	MASE	RMSE	MAPE	MASE
RCP 1	ESN model	0.0450	18.2183	2.4145	0.0521	20.1742	2.7480	0.0544	21.2928	2.9608
	Proposed model	**0.0203**	**8.7511**	**1.0871**	**0.0226**	**11.4607**	**1.2278**	**0.0338**	**20.1416**	**1.7015**
RCP 2	ESN model	0.0496	7.7376	2.2372	0.0511	9.4669	2.6751	0.0672	11.1388	3.1238
	Proposed model	**0.0246**	**3.9053**	**1.1355**	**0.0300**	**4.3849**	**1.3255**	**0.0463**	**6.3652**	**1.9812**
RCP 3	ESN model	0.0647	11.3833	2.3444	0.0616	11.1641	2.3267	0.0750	12.7170	2.5916
	Proposed model	**0.0256**	**4.0837**	**0.8898**	**0.0309**	**4.9342**	**1.0701**	**0.0408**	**5.9058**	**1.2537**
RCP 4	ESN model	0.0419	3.7228	2.0821	0.0480	4.5109	2.5750	0.0675	6.0974	3.5221
	Proposed model	**0.0231**	**1.9948**	**1.1201**	**0.0303**	**2.8291**	**1.6248**	**0.0312**	**2.8147**	**1.6339**
RCP 5	ESN model	0.0380	5.1331	1.0158	0.0578	8.0459	1.5163	0.0835	13.3672	2.3941
	Proposed model	**0.0347**	**4.7995**	**1.0016**	**0.0471**	**6.1768**	**1.1888**	**0.0548**	**7.5077**	**1.4756**

6. Conclusion

The aim of this work is the development of a multi-step ahead prediction method, which is applied to time series predictions of NPPs components. For this aim, we have presented a hybrid prediction framework based on EEMD and LSTM. Moreover, our proposed method has also tackled two additional issues. Firstly, a TPE algorithm has been employed to address the automatic hyperparameter optimization for LSTM networks. Secondly, multi-step ahead predictions have been performed by applying a MIMO strategy with respect to three different long-term horizons. Several performance evaluations have been carried out to analyze and validate the methodology of the proposed method with a practical case study of the time series data acquired from real NPPs components. The results have shown the promising performance of the proposed hybrid method in achieving accurate predictions for long-term horizons.

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] 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.

[4] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering and System Safety*. 2009.

[5] M. G. Na *et al.*, "Prediction of major transient scenarios for severe accidents of nuclear power plants," *IEEE Trans. Nucl. Sci.*, 2004.

[6] 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.

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

[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] 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.

[11] M. Qi and G. P. Zhang, "Trend time-series modeling and forecasting with neural networks," *IEEE Trans. Neural Networks*, 2008.

[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] J. Liu and E. Zio, "A SVR-based ensemble approach for drifting data streams with recurring patterns," *Appl. Soft Comput. J.*, 2016.

[17] J. Liu, V. Vitelli, E. Zio, and R. Seraoui, "A Novel Dynamic-Weighted Probabilistic Support Vector Regression-Based Ensemble for Prognostics of Time Series Data," *IEEE Trans. Reliab.*, 2015.

[18] Y. Bao, T. Xiong, and Z. Hu, "Multi-step-ahead time series prediction using multiple-output support vector regression," *Neurocomputing*, 2014.

[19] D. Wang, H. Luo, O. Grunder, and Y. Lin, "Multi-step ahead wind speed forecasting using an improved wavelet neural network combining variational mode decomposition and phase space reconstruction," *Renew. Energy*, 2017.

[20] 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.

[21] B. Doucoure, K. Agbossou, and A. Cardenas, "Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data," *Renew. Energy*, 2016.

[22] 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.

[23] V. T. Tran, B. S. Yang, and A. C. C. Tan, "Multi-step ahead direct prediction for the machine condition

prognosis using regression trees and neuro-fuzzy systems," *Expert Syst. Appl.*, 2009.

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

[25] 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.

[26] 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.

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

[28] H. Liu, X. wei Mi, and Y. fei Li, "Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network," *Energy Convers. Manag.*, 2018.

[29] J. Wang and Y. Li, "Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy," *Appl. Energy*, 2018.

[30] Y. Li, H. Wu, and H. Liu, "Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEWT reconstruction," *Energy Convers. Manag.*, 2018.

[31] Y. L. Hu and L. Chen, "A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm," *Energy Convers. Manag.*, 2018.

[32] I. A. Araya, C. Valle, and H. Allende, "A Multi-Scale Model based on the Long Short-Term Memory for day ahead hourly wind speed forecasting," *Pattern Recognit. Lett.*, 2019.

[33] X. Yuan, C. Chen, M. Jiang, and Y. Yuan, "Prediction interval of wind power using parameter optimized Beta distribution based LSTM model," *Appl. Soft Comput. J.*, 2019.

[34] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, 2018.

[35] S. Srivastava and S. Lessmann, "A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data," *Sol. Energy*, 2018.

[36] K. Wang, X. Qi, and H. Liu, "Photovoltaic power forecasting based LSTM-Convolutional Network," *Energy*, 2019.

[37] Q. Wu and H. Lin, "Daily urban air quality index forecasting based on variational mode decomposition, sample entropy and LSTM neural network," *Sustain. Cities Soc.*, 2019.

[38] Y. Zhou, F. J. Chang, L. C. Chang, I. F. Kao, and Y. S. Wang, "Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts," *J. Clean. Prod.*, 2019.

[39] 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.

[40] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A Stat. Mech. its Appl.*, 2019.

[41] 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.

[42] H. Su, E. Zio, J. Zhang, M. Xu, X. Li, and Z. Zhang, "A hybrid hourly natural gas demand forecasting method based on the integration of wavelet transform and enhanced Deep-RNN model," *Energy*, 2019.

[43] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Appl. Energy*, 2019.

[44] O. Laib, M. T. Khadir, and L. Mihaylova, "Toward efficient energy systems based on natural gas consumption prediction with LSTM Recurrent Neural Networks," *Energy*, 2019.

[45] L. Yu, S. Liang, R. Chen, and K. K. Lai, "Predicting monthly biofuel production using a hybrid ensemble forecasting methodology," *Int. J. Forecast.*, 2019.

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

[47] Z. Hajirahimi and M. Khashei, "Hybrid structures in time series modeling and forecasting: A review," *Eng.*

Appl. Artif. Intell., 2019.

[48] 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.

[49] 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.

[50] 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.

[51] Z. Shao, F. Chao, S. L. Yang, and K. Le Zhou, "A review of the decomposition methodology for extracting and identifying the fluctuation characteristics in electricity demand forecasting," *Renewable and Sustainable Energy Reviews*. 2017.

[52] 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.

[53] 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.

[54] 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.

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

[56] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, 2009.

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

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

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

[60] 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.

[61] 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.

[62] 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.

[63] 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.

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

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

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

[67] H. Jaeger, "Echo state network," *Scholarpedia*, 2007.

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

université PARIS-SACLAY

ECOLE DOCTORALE N°573
Approches interdisciplinaires, fondements, applications et innovation (Interfaces)

Titre : Méthodes de prédiction basées sur des modèles et pilotées par les données pour les pronostics

Mots clés : prognostics and health management, pronostics basés sur des modèles, pronostics pilotées par les données, prédictions à pas multiples, croissance des fissures par fatigue, composants de centrales nucléaires

Résumé : La dégradation est un phénomène inévitable qui affecte les composants et les systèmes, qui peut entraîner leurs défaillances avec des conséquences potentiellement catastrophiques selon l'application. La motivation de cette Thèse est d'essayer de modéliser, d'analyser et de prédire les défaillances par des méthodes pronostiques qui peuvent permettre une gestion prédictive de la maintenance des actifs. Cela permettrait aux décideurs d'améliorer la planification de la maintenance et de minimiser les arrêts imprévus, augmentant ainsi la disponibilité et la sécurité du système. Dans cet objectif, la recherche pendant la thèse a été consacrée à l'adaptation et à l'utilisation d'approches basées sur des modèles et d'approches pilotées par les données pour traiter les processus de dégradation qui peuvent conduire à différents modes de défaillance dans les composants industriels, en utilisant différentes sources d'informations et de données pour effectuer des prédictions sur l'évolution de la dégradation et estimer la durée de vie utile restante.

Les principales contributions de cette thèse ont été divisées en deux parties traitant de deux applications pronostiques spécifiques, y compris les pronostics basés sur des modèles pour la prédiction de la croissance des fissures par fatigue et les pronostics pilotées par les données pour les prédictions à pas multiples des données de séries chronologiques des composants des Centrales Nucléaires, respectivement.

La performance d'une approche pronostique basée sur des modèles dépend du choix des modèles adoptés de Physics-of-Failure (PoF). Cependant, chaque modèle de dégradation ne convient qu'à certains processus de dégradation dans certaines conditions de fonctionnement, qui souvent ne sont pas connues avec précision. Pour résoudre ce problème, nous développons deux méthodes basées sur des modèles qui reposent sur l'ensemble de plusieurs modèles de dégradation, afin de tirer parti de la complémentarité de différents modèles, spécifiques aux tendances de dégradation à prévoir. Les principales contributions de l'ensemble proposé de méthodes basées sur des modèles sont deux nouvelles stratégies d'ensemble pondérées, qui prennent en compte les précisions de prédiction des modèles individuels lors d'instances de temps précédentes. De plus, le filtrage Bayésien récursif et le filtrage particulaire sont utilisés pour prédire et mettre à jour dynamiquement l'évolution de la dégradation et la durée de vie utile restante du composant à chaque étape de prédiction. Pour valider les performances des méthodes proposées, différentes études de cas de croissance des fissures par fatigue générées avec des conditions de fonctionnement variables dans le temps sont considérées.

Dans l'industrie nucléaire, les composants et les systèmes sont conçus pour garantir des niveaux de fiabilité très élevés étant donné les conséquences potentiellement catastrophiques de leurs défaillances, et des capacités pronostiques sont recherchées pour prédire avec précision les comportements de dégradation à long terme des composants et des systèmes, permettant de planifier les interventions de maintenance des composants critiques bien à l'avance et de réduire les coûts de maintenance. Cependant, plus loin on tente de prédire l'avenir, plus il est difficile d'obtenir une prédiction précise et stable en raison de l'augmentation de l'incertitude et de l'accumulation d'erreurs. Pour cette raison, la prédiction à plusieurs étapes est restée une tâche difficile dans de nombreuses applications pronostiques, en particulier dans l'industrie nucléaire. Pour résoudre ce problème, cette thèse propose deux nouvelles méthodes de prédiction à étapes multiples basées sur la Long Short-Term Memory (LSTM), un réseau de neurones profond développé pour traiter les dépendances à long terme dans les données de séries chronologiques. La première méthode de prédiction à plusieurs étapes est appliquée pour prédire jusqu'à 45 jours à l'avance les paramètres de fonctionnement des Générateurs de Vapeur de Centrales Nucléaires. La méthode aborde également les problèmes supplémentaires d'optimisation automatique des hyperparamètres et de quantification de l'incertitude de prédiction en utilisant respectivement un algorithme d'optimisation Tree-structured Parzen Estimator (TPE) et une technique de Monte Carlo (MC) Dropout. Une étude de cas concernant les données des Générateurs de Vapeur acquises auprès de différentes Centrales Nucléaires est réalisée pour valider les performances de la méthode proposée. D'autre part, la deuxième méthode de prédiction à plusieurs étapes est développée sur la base d'un cadre hybride intégrant la Ensemble Empirical Mode Decomposition (EEMD) et le réseau de neurones LSTM, et appliquée sur une étude de cas concernant les données de séries chronologiques acquises à partir des Pompes de Refroidissement de Réacteurs de Centrales Nucléaires. Dans ce cadre de prédiction, EEMD est utilisée pour décomposer des séries temporelles en un ensemble de composants qui permettent de décrire efficacement la dynamique du système et facilitent donc la tâche de prédiction. Ensuite, des modèles de réseaux de neurones LSTM sont développés pour prédire le comportement à plusieurs étapes des composants individuels et les prédictions obtenues sont agrégées pour reconstruire des données de séries chronologiques. Un algorithme TPE est utilisé pour l'optimisation automatique des hyperparamètres. La performance de la méthode proposée est validée en considérant trois horizons de prédiction à long terme sur une étude de cas pratique des Pompes de Refroidissement de Réacteurs de Centrales Nucléaires.

Title : Model-based and data-driven prediction methods for prognostics

Keywords : prognostics and health management, model-based prognostics, data-driven prognostics, multi-step ahead prediction, fatigue crack growth, nuclear power plant components

Abstract : Degradation is an unavoidable phenomenon that affects engineering components and systems, which may lead to their failures with potentially catastrophic consequences depending on the application. The motivation of this Thesis is trying to model, analyze and predict failures with prognostic methods that can enable a predictive management of asset maintenance. This would allow decision makers to improve maintenance planning and minimize unexpected shutdowns, thus increasing system availability and safety. To this aim, research during the Thesis has been devoted to the tailoring and use of both model-based and data-driven approaches to treat the degradation processes that can lead to different failure modes in industrial components, making use of different information and data sources for performing predictions on the degradation evolution and estimating the Remaining Useful Life (RUL).

The main contributions of the Ph.D. work have been divided into two parts addressing two specific prognostic applications, including model-based prognostics for fatigue crack growth prediction and data-driven prognostics for multi-step ahead predictions of time series data of Nuclear Power Plant (NPP) components, respectively.

The performance of a model-based prognostic approach depends on the choice of the adopted Physics-of-Failure (PoF) models. However, each degradation model is appropriate only to certain degradation process under certain operating conditions, which are often not precisely known. To address this problem, we develop two model-based methods based on the ensemble of multiple degradation models, in order to take advantage of the complementarity of different models, specific on the degradation trends to be predicted. The main contributions of the proposed ensemble of models-based methods are two novel weighted ensemble strategies, which take into account the prediction accuracies of the individual models at previous time instances. In addition, recursive Bayesian filtering and Particle Filtering (PF) are employed to dynamically predict and update the degradation evolution and the component RUL at each prediction step. To validate the performances of the proposed methods, different case studies of fatigue crack growth generated with time-varying operating conditions are considered.

In the nuclear industry, components and systems are designed to guarantee very high reliability levels given the potentially catastrophic consequences of their failures, and prognostic capabilities are sought to accurately predict the long-term degradation behaviors of the components and systems, allowing maintenance interventions of critical components to be planned well in advance and reducing maintenance costs. However, the further one attempts to predict into the future, the harder it is to achieve an accurate and stable prediction due to increasing uncertainty and error accumulation. For this reason, multi-step ahead prediction has remained a difficult task in many prognostic applications, particularly in the nuclear industry. To address this problem, this Thesis proposes two novel multi-step ahead prediction methods based on Long Short-Term Memory (LSTM), a deep neural network developed for dealing with the long-term dependencies in time series data. The first multi-step ahead prediction method is applied for predicting up to 45 days ahead the operating parameters of NPP Steam Generators (SGs). The method also addresses the additional issues of automatic hyperparameter optimization and prediction uncertainty quantification by using a Tree-structured Parzen Estimator (TPE) optimization algorithm and a Monte Carlo (MC) Dropout technique, respectively. A case study concerning SG data acquired from different NPPs is carried out to validate the performance of the proposed method. On the other hand, the second multi-step ahead prediction method is developed based on a hybrid framework integrating Ensemble Empirical Mode Decomposition (EEMD) and LSTM neural network and applied to a case study concerning time series data acquired from Reactor Coolant Pumps (RCPs) of NPPs. In this prediction framework, EEMD is used to decompose time series into a set of components which allow effectively describing the system dynamics and therefore facilitates the prediction task. Then, LSTM neural network models are developed for predicting the multi-step ahead behavior of the individual components and the obtained predictions are aggregated to reconstruct the time series. A TPE algorithm is employed for automatic hyperparameter optimization. The performance of the proposed method is validated by considering three different long-term prediction horizons on a practical case study of NPP RCPs.