

Remaining Useful Life Estimation based on Noise Injection and a Kalman Filter Ensemble of modified Bagging Predictors

Hung-Cuong Trinh^{1*}, Van-Huy Pham², and Anh H. Vo³

¹ Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh, Vietnam
[e-mail: trinhhungcuong@tdtu.edu.vn]

² Data Science Laboratory, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh, Vietnam
[e-mail: phamvanhuy@tdtu.edu.vn]

³ Department of Computer Engineering, Sejong University, Seoul, South Korea
[e-mail: vohoanganh@sju.ac.kr]

*Corresponding author: Hung-Cuong Trinh

*Received July 20, 2023; revised September 26, 2023; accepted November 21, 2023;
published December 31, 2023*

Abstract

Ensuring reliability of a machinery system involve the prediction of remaining useful life (RUL). In most RUL prediction approaches, noise is always considered for removal. Nevertheless, noise could be properly utilized to enhance the prediction capabilities. In this paper, we proposed a novel RUL prediction approach based on noise injection and a Kalman filter ensemble of modified bagging predictors. Firstly, we proposed a new method to insert Gaussian noises into both observation and feature spaces of an original training dataset, named GN-DAFC. Secondly, we developed a modified bagging method based on Kalman filter averaging, named KBAG. Then, we developed a new ensemble method which is a Kalman filter ensemble of KBAGs, named DKBAG. Finally, we proposed a novel RUL prediction approach GN-DAFC-DKBAG in which the optimal noise-injected training dataset was determined by a GN-DAFC-based searching strategy and then inputted to a DKBAG model. Our approach is validated on the NASA C-MAPSS dataset of aero-engines. Experimental results show that our approach achieves significantly better performance than a traditional Kalman filter ensemble of single learning models (KESLM) and the original DKBAG approaches. We also found that the optimal noise-injected data could improve the prediction performance of both KESLM and DKBAG. We further compare our approach with two advanced ensemble approaches, and the results indicate that the former also has better performance than the latters. Thus, our approach of combining optimal noise injection and DKBAG provides an effective solution for RUL estimation of machinery systems.

Keywords: Bagging, data augmentation, ensemble, feature construction, Kalman filter, noise addition, remaining useful life.

1. Introduction

1.1 Related Works

Prognostics and Health Management (PHM) has been developed to ensure reliability and availability in machinery systems. PHM allows a system effectively conducts maintenance and manages equipment usage. One of main tasks in PHM is estimating the remaining useful life (RUL) of degrading equipment. Many data-driven approaches have been developed to predict RUL of critical equipment. Among the proposed approaches, machine learning approaches have been progressively favored with recent improvement in sensor systems and data analysis methods. Sensor data like pressure, temperature, rotor speed, and so forth can be simply measured, and is directly used as inputs of a machine learning method.

Regards to data-driven approaches, many different kinds of machine learning methods and complemented data pre-processing algorithms have been employed to predict RUL. For instances, a hybrid convolutional neural network together with a feature attention algorithm and a multi-scale cycle attention algorithm were developed to estimate RUL of lithium-ion batteries [1]. Another study proposed a new hybrid machine learning algorithm by combining Monte Carlo simulation and adaptive dropout long short-term memory, and results showed that the hybrid method improved performance of battery RUL prediction [2]. Effectiveness of multilayer perceptron and radial basis function neural networks in RUL estimation of ball bearings has been investigated [3]. A support vector machine combining with information entropy preprocessing was proposed to predict RUL of lubricating oil [4]. A gradient boosting decision tree model in conjunction with relative entropy distance-based fault severity was integrated to estimate RUL of electronic elements [5]. These mentioned studies applied only a single machine learning algorithm for RUL estimation.

In comparison to single learning algorithms, ensemble machine learning approaches provide better performance by combining the predictions from a number of single learning models. Two main types of ensemble learning methods are averaging methods and boosting methods. In averaging methods, a number of base estimators are independently built and then average their predictions, such as bagging methods, random forest. In contrast, boosting methods are focused on iteratively combining weak single learners to build a strong combined learner, such as adaptive boosting, gradient boosting. Thus, various ensemble machine learning approaches have been developed or utilized to enhance the RUL prediction performance. A previous paper presented a smart ensemble of gradient boosted trees and multilayer perceptron neural networks to predict RUL of degrading turbofan engines in NASA C-MAPSS datasets [6]. Another study proposed a novel ensemble long short-term memory neural network approach to enhance RUL prediction of turbofan engines [7]. A Bayesian optimization stacking ensemble learning method was developed for RUL prediction of a catenary [8]. The stacking ensemble learning method combines four learning methods with large differences, deep neural networks, support vector machine, extreme gradient boosting, and k-nearest neighbors, and thus achieves better RUL prediction results. A stacking-based ensemble learning method that combines five regression algorithms (linear regression, support vector machine, decision tree, random forest, and extreme gradient boosting) was developed to increase RUL prediction performance on NASA's turbofan engine degradation datasets [9]. An optimized random forest model was proposed to obtain the underlying mapping relationship between the aging features and capacity, then RUL predictions of li-ion batteries were achieved [10]. Furthermore, several previous studies applied Kalman filter into an

ensemble learning approach to attain a more accurate RUL prediction. Leto Peel proposed a Kalman filter-based ensemble for fusing multiple neural network model predictions over the dataset of PHM 2008 Data Challenge [11]. The results showed that the filtering of models can reduce RUL prediction error. Another study utilized the Kalman filter-based ensemble for fusing RUL predictions of multiple optimal learning models obtained from a genetic algorithm search [12].

The above-mentioned studies focus on the advancements of new data pre-processing algorithms and machine learning architectures, though utilize only the original data. In all these approaches, noise is always regarded as redundant and removed from the observations. However, noise could be properly used to improve the prediction capabilities. Previous works also tried to investigate the application of noise in enhancing the input space and in deploying a stable and reliable system. Several studies have indicated that injecting noise into neural networks can increase the convergence speed of the training process and improve the predictions [13] [14] [15]. Injection of noise into the input data of a neural network during training could lead to crucial enhancements in prediction performance. Besides that, the noise change each data point, so a learning method encounter difficulties in fitting individual data points precisely; thus it will reduce over-fitting. A number of previous studies also utilized noise to avoid the over-fitting issue in machine learning methods [16] [17] [18]. Regards to the PHM field, it has been showed that noise utilization can enhance the faults detection performance for machinery systems [19] [20] [21] [22]. Nevertheless, few studies utilize noise for RUL prediction. We found only a previous study in which proposed a new RUL prediction approach by applying noise injection into a long short-term memory network [23]. Therefore, it is necessary to develop more approaches that can utilize noise to predict RUL.

Inspired by the potentiality of ensemble learning and noise injection, we develop a novel RUL prediction approach based on Gaussian noise injection and a Kalman filter-based ensemble of modified bagging predictors. Firstly, we proposed a new method to insert Gaussian noises into observation and feature spaces of an original dataset, named GN-DAFC. In this way, adding noise to observations is a simple kind of data augmentation, and inserting noise into features corresponds to a method of feature construction. Secondly, we developed a modified version of the bagging method based on Kalman filter averaging, named KBAG. The modification is that KBAG utilizes a Kalman filter-based averaging method rather than a classical averaging one. Thirdly, we further developed a new ensemble method, named DKBAG, in which a Kalman filter ensemble of KBAGs is constructed. The Kalman filters are employed in both two layers of DKBAG: one inside each KBAG, and the final one outside all KBAG estimators. Finally, we proposed a novel RUL prediction approach GN-DAFC-DKBAG in which the optimal noise-injected training dataset was determined by a GN-DAFC-based searching strategy and then inputted to a DKBAG model. The GN-DAFC-based searching strategy of optimal noise injection is the main novel point in our approach. In this strategy, we used a cross-validated grid-search to find optimal Gaussian noise-related parameters such as noise intensity, proportion of inserted noisy observations/features. To verify the effectiveness of our approach, we compared it to a traditional Kalman filter-based ensemble of single learning models (KESLM). Experimental results over the NASA C-MAPSS dataset of aero-engines showed that our approach achieved significantly better performance than the latter approach, with a practically acceptable running time. We also found that KESLM could achieve better RUL predictions by applying the optimal noise injection. Moreover, we investigated our approach in case of skipping the noise injection strategy or the DKBAG with only original data. We observed a notable decrease in the prediction performance of our approach in that case. It turn out that optimal noise-injected

data could improve the prediction performance of both the traditional ensemble of single models - KESLM and the proposed ensemble approach - DKBAG. We further compared our approach with two advanced ensemble approaches: a heuristic Kalman filter ensemble [11] and an ensemble of genetic algorithms [12]. The results indicated that our approach also has better performance than the two previous ones. In summary, our approach of combining optimal noise injection and DKBAG yields a potential solution for RUL prediction of machinery systems.

In the remainder of this paper, single/bagging machine learning models, the traditional Kalman filter-based ensemble of single machine learning models, performance metrics, and the cross-validated grid-search are presented in Section 2. Our proposed approach is presented in Section 3. Then, results of our approach and the traditional Kalman filter-based ensemble of single learning models are evaluated in Section 4. We further discussed the results in Section 5. Finally, results, limitations, and future challenges of our approach are concluded in Section 6.

1.2 Contributions

The main difference between our work and previous works is that we utilize both noise and Kalman filters to improve the prediction performance of ensemble learning methods. The following points denote key contributions of our work:

- Firstly, we propose a new method to inject Gaussian noises into two spaces of an original dataset: observation space and feature space. This method, named GN-DAFC, corresponds to a simple combination of data augmentation and feature construction methods.
- Secondly, we develop a modified version of the bagging method based on Kalman filter averaging, named KBAG. We enhance the architecture of the traditional bagging method by employing a Kalman filter-based averaging method rather than a classical averaging one.
- Thirdly, we further developed a new ensemble method, named DKBAG, in which a Kalman filter ensemble of KBAGs is constructed. The Kalman filters are employed in both two layers of DKBAG: one inside each KBAG, and the final one outside all KBAG estimators. Thus, DKBAG could be considered as a two-layer Kalman filters-based ensemble.
- Finally, we propose a novel RUL prediction approach GN-DAFC-DKBAG in which is divided into two stages as follows. First, a GN-DAFC-based searching strategy is conducted to find optimal Gaussian noise-related parameters such as noise intensity, proportion of inserted noisy observations/features. As a result, the optimal noise-injected training dataset was efficiently determined. This strategy is the most crucial point in our approach. Second, the optimal noise-injected data is then inputted into a DKBAG estimator to infer RUL. Our approach showed a significantly better prediction performance than DKBAG without noise injection, a traditional Kalman filter-based ensemble of single learning models, and two advanced ensemble approaches.

2. Background

2.1 Single machine learning models

In this study, three classical single machine learning models such as the multi-layer perceptron network, decision trees, and support vector machine were employed. Detailed usage of these single learning models is illustrated as follows.

2.1.1 Multi-layer Perceptron neural networks

A multi-layer perceptron (MLP) is a type of feed-forward artificial neural networks in which consists of multiple fully-connected layers [24]. It comprises three kinds of layers: the input layer, hidden layer, and output layer. A MLP has one input layer which receives input features to be handled, and one output layer that is responsible for tasks like prediction and classification. An arbitrary number of hidden layers which are located between the output and input layers are the main mechanism of MLP. Data is passed in a forward path from the input to the output layer in MLP. And the backpropagation learning method is applied to train all neurons in the MLP. The main advantages of MLP are applicability in complex non-linear problems and working well with large input data; though the training phase is time consuming. Moreover, with regard to the training phase, a MLP requires tuning a number of hyper-parameters such as the number of neurons in each hidden layer, the activation function for the hidden layers, and the solver for weight optimization.

2.1.2 Decision Trees

A decision tree (DT) is a non-parametric supervised learning method, which is used for both classification and regression problems [25]. It has a tree structure that comprises a root node, branches, internal nodes, and leaf nodes. An internal node, also known as a decision node denotes a test on a data feature, and its outgoing branches represent outcomes of the test. Leaf nodes or the terminal nodes represent all possible outcomes of the dataset. The DT method applies a divide-and-conquer strategy by executing a greedy search to determine the best split points within a tree. DT model can predict the values of a target variable by learning simple decision rules inferred from the data features. Some main advantages of a decision tree are its simplicity, easy interpretation, and requiring little data preparation; although a DT also has some disadvantages such as over-fitting issue and unstableness. Moreover, regarding the training process, a DT requires tuning a numbers of hyper-parameters such as the split criterion, the maximum depth of the tree, the minimum number of observations required to split an internal node.

2.1.3 Support Vector Machines

Support vector machine (SVM) is another supervised learning algorithm, which is used for both classification and regression tasks [26]. The main objective of the SVM algorithm is to find the hyper-plane that maximizes the margin (distance) between the hyper-plane and the closest data points, while also try to minimize the prediction error. The dimension of the hyper-plane depends upon the number of input features. In case the number of input features is two, then the hyper-plane is just a line. If the number of input features is three, then the hyper-plane becomes a 2D plane, and so on. The main advantages of SVM are applicability in complex non-linear problems, effectiveness in high-dimensional data, less memory requirement, and different or custom kernel functions can be specified for the decision functions; though over-fitting is a crucial problem in case the number of features is much greater than the number of observations, and it is also time-consuming for a huge dataset. Moreover, regarding the training process, a SVM requires tuning a numbers of hyper-parameters such as the kernel function, regularization parameter C , and the kernel coefficient “gamma”.

2.2 Bagging machine learning models

Ensemble learning has become a notable concept in machine learning field recently. In this study, we focus on a type of ensemble learning methods, bagging. Bagging is a technique for creating multiple versions of a learning estimator and using these to obtain an aggregated estimator [27]. Each estimator version is trained on a random subset of the original training set, and then all individual predictions of the estimator versions are aggregated to obtain a final prediction. Thus, bagging is commonly used to reduce variance of a base learning estimator like the DT, MLP, or SVM one. Regularly, bagging is a simple way to improve the stability of a single learning model, without adapting the base algorithm of the model. Some advantages of bagging are reducing over-fitting and variance, and working well with strong complex learning models; though loss of interpretability and expensive computation are its challenges. Moreover, with regard to the training phase, bagging requires tuning a numbers of hyper-parameters such as the number of base estimators, and the number of observations in a random subset used to train an estimator version. In Section 3, we provide a new improved version of the bagging technique that utilizes Kalman filter-based averaging rather than the classical averaging method.

2.3 Traditional Kalman filter-based ensemble of single machine learning models

In this section, we introduce another kind of ensemble learning methods, a Kalman filter-based ensemble of single machine learning models (KESLM). Several previous works proposed a Kalman filter-based ensemble to fuse predictions from multiple machine learning models [11] [12]. The results showed that the filtering of model predictions can reduce the prediction error. These studies applied a discrete linear Kalman filter which is a recursive method for estimating the state of a process. It comprises two stages: a predict stage and an update stage. The predict stage determines the state estimate over time, and the update stage improves the state estimate with the observations. Although there are several important parameters in a Kalman filter construction, we note that the observation noise covariance is set to mean squared error values from the learning models.

Similar to other ensemble methods, a KESLM also consists of two main steps: firstly creating individual members of the ensemble, and secondly fusing the outputs of the ensemble members. In the first step, the KESLM should use diverse learning models to improve the generalization of the ensemble. In this study, we used three types of learning modes in the KESLM: MLP, DT, and SVM. For the second step, the KESLM applies the Kalman filter to combine the outputs of all ensemble members. In comparison to traditional average methods such as weighted mean or median, the Kalman filter-based averaging can provide smoother estimations over time [11].

2.4 Performance metrics

Considering we applied the Kalman filter-based ensembles, mean squared error (MSE) is a proper measure to evaluate the performance of our and other approaches since it is also used in the observation noise covariance of the Kalman filters. The following presents the MSE formulation in RUL prediction tasks:

$$MSE = \frac{1}{n} \sum_{i=1}^n (R\hat{U}L_i - RUL_i)^2 \quad (1)$$

where $R\hat{U}L_i$ and RUL_i are the predicted and the actual RUL values of the i -th observation

among a total of n observations, respectively.

2.5 CVGS, a cross-validated grid-search

Firstly, we introduce the cross-validation (CV) technique which is widely used in machine learning fields. The most popular type of CV is k -fold CV, in which the training data is split into k smaller subsets or k “folds”. For each fold, the learning model is trained using $k - 1$ other folds, and the resulting model is validated on the current fold. Thus, a performance score is returned for each fold. The average of all performance scores is called CV score, and it is used to evaluate the performance of the learning model. In summary, CV is a technique that requires dividing the training data into multiple subsets, training the learning model multiple times, and using a different subset as the validation data for each time. This can improve the model’s performance and help to avoid over-fitting.

To find optimal hyper-parameters of a learning model, many previous studies apply a cross-validated grid-search (CVGS) method on a training data [28] [29] [30]. Initially, the grid search specifies a list of values for each hyper-parameter that needs to be optimized, and then it creates all combinations of these values. For each combination of hyper-parameters values, the grid search calculates the CV score of the learning model on the training data based on the CV technique. The combination with the best CV score is selected for the learning model. Briefly, CVGS exhaustively explores the entire search space by trying all possible combinations of hyper-parameters. Thus, CVGS is a time-consuming process in case the number of involved hyper-parameters combinations is large or the learning model is complex. However, it is guaranteed to obtain the optimal combination of hyper-parameters.

3. Proposed methods

In this section, we firstly propose a new noise injection method to insert Gaussian noises into an original dataset. Secondly, we develop a new ensemble method by combining Kalman filters and bagging methods. Finally, we propose a novel RUL prediction approach by using the above methods.

3.1 GN-DAFC, a Gaussian noise-based Data augmentation and Feature construction

We develop a new method to insert Gaussian noises into two spaces of an original dataset: observation space and feature space, named GN-DAFC. This method corresponds to a simple utilization of noise in data augmentation and feature construction methods. As shown in **Fig. 1**, the procedure of GN-DAFC consists of two continuous stages: noisy observations insertion (data augmentation), and then noisy features insertion (feature construction). Given an original dataset of m input features $D = [f_1, f_2, \dots, f_m]$, where f_k is the k -th input feature ($1 \leq k \leq m$). We also assume that D contains n observations $[ob_1, ob_2, \dots, ob_n]$. The following is the details of the GN-DAFC method:

In the first stage, steps are conducted in the following order:

- Firstly, a noisy dataset (named ND) of same size with the original dataset D is generated by adding random Gaussian noise in each feature of D . Equation (2) describes the process of creating a new noisy feature \tilde{f}_k from the original one f_k , as follows:

$$\tilde{f}_k = f_k + N(0, ((\max(f_k) - \min(f_k)) \times I_k)^2) \quad (2)$$

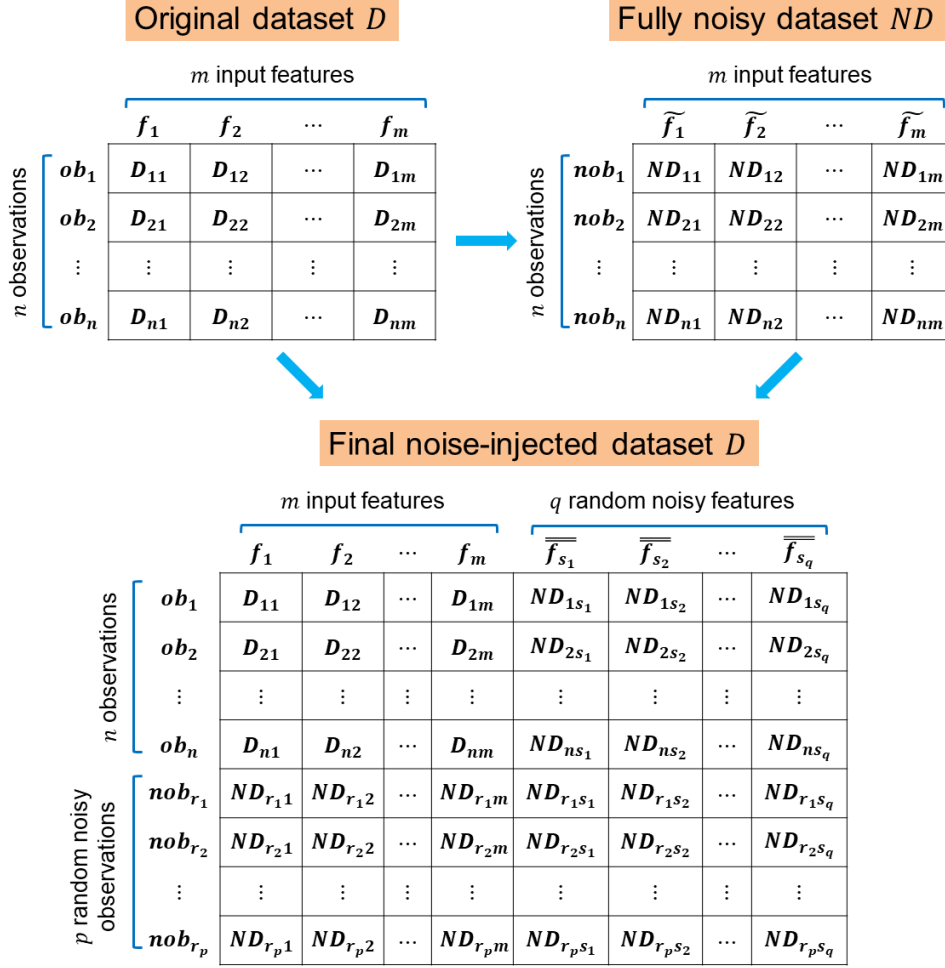


Fig. 1. Overview of the Gaussian noise-based Data augmentation and Feature construction, GN-DAFC. The procedure consists of two continuous stages: noisy observations insertion (data augmentation), and then noisy features insertion (feature construction). In our work, we specify $p \in \{0.1n, 0.3n, 0.5n\}$ and $q \in \{0.3m, 0.7m\}$ (m & n are the numbers of features/observations in the original dataset D).

where:

- $N(\cdot)$ is a random variable which is distributed normally with zero mean and standard deviation $(\max(f_k) - \min(f_k)) \times I_k$.
- $\max(f_k)$ and $\min(f_k)$ functions return the maximum and minimum values of the original feature f_k , respectively.
- I_k is the intensity of additive Gaussian noise for the feature f_k .

Based on the above formulation, the values of noisy features are generated through the addition of Gaussian noise into corresponding original features. To control the amount of spread or noise intensity, the standard deviation of the random Gaussian noise can be adapted based on the scale of each input feature. In addition, we skip the cases of the original feature f_k with constant value. Consequently, the noisy dataset ND also

has m input features $ND = [\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_m]$. Furthermore, the noise intensity I_k of each feature f_k is randomly determined by specifying a noise intensity range as follows:

$$I_k = \frac{\text{randint}(\text{low_thresh}, \text{up_thresh})}{100} \quad (3)$$

where:

- $\text{randint}(\text{low_thresh}, \text{up_thresh})$ function returns a random integer in range $[\text{low_thresh}, \text{up_thresh}]$, including both end points.
- low_thresh and up_thresh are lower and upper threshold values of noise intensity (in percentage).

Based on (2) and (3), the optimal range of noise intensity $[\text{low_thresh}, \text{up_thresh}]$ is an important parameter of the first stage in the GN-DAFC procedure. In Section 3, our proposed approach conducts a grid-search to find this optimal noise intensity range.

- Next, we randomly select p noisy observations from the noisy dataset ND , and then insert them into the original dataset D . As a result, the original dataset D contains $n + p$ observations $[ob_1, ob_2, \dots, ob_n, nob_{r_1}, nob_{r_2}, \dots, nob_{r_p}]$, where $nob_{r_1}, nob_{r_2}, \dots, nob_{r_p}$ are the generated random noisy observations. The parameter p is also crucial in GN-DAFC though it denotes how large the ratio of inserted noisy observations over the original ones is. In our experiments, we choose p values such that noisy observations take up 10%, 30%, or 50% of the number of original observations.

In the second stage, GN-DAFC continuously inserts new noisy features into the noise-injected dataset D as in the following steps.

- Step 1, we randomly select a feature \tilde{f}_k of the fully noisy dataset ND , and vertically merge it with a part of the feature f_k of the noise-injected dataset D . Accordingly, a new noisy feature \bar{f}_k is generated based on the following formulation:

$$\bar{f}_k = \tilde{f}_k \oplus f_k[n + 1, n + p] \quad (4)$$

where \oplus denotes the vertical merge operation, $f_k[n + 1, n + p]$ is a part of the feature f_k from $(n + 1)$ -th observation to $(n + p)$ -th one in the noise-injected dataset D . Then we append the new noisy feature \bar{f}_k into the dataset D .

- Step 2, move to Step 1 again until the number of newly generated noisy features is equal to q . The parameter q play an important role in the GN-DAFC procedure though it represents how large the ratio of noisy inserted features over the original ones is. In our experiments, we choose q values such that the number of inserted noisy features takes up 30% or 70% of the number of original ones.

In summary, two stages of the GN-DAFC procedure are clearly summarized **Fig. 1**. The RUL prediction performance is not only impacted by the quality of a learning model, but also the inputs of the learning model. Inspired of that, GN-DAFC utilizes Gaussian noise to expand the size of the original training dataset in both observation and feature spaces. This is also the main difference from most of existing noise utilization approaches that inject noise only in the observation space of the training data [19] [20] [21] [22] [23]. By repeating the procedure

GN-DAFC, many different noisy datasets can be generated based on the original dataset. Thus, it is an effective and convenient way to increase the generalization of a learning approach. We also need to properly adjust the intensity of noise for each input feature, and the ratio of inserted noisy observations/features. Too weak noise intensity or small ratio of inserted noisy data has no influence on the learning model, whereas too strong noise intensity or large ratio of inserted noisy data makes the learning model too challenging to train.

3.2 DKBAG, a Kalman filter-based ensemble of modified bagging estimators

In this section, we firstly propose a new improved version of the bagging technique, named KBAG, in which utilizes Kalman filter-based averaging method than the classical averaging one. Similar to a traditional bagging, KBAG also generates random subsets of an initial dataset by implementing a resampling with replacement strategy. This resampling strategy allows a given observation to be included in a given subset more than once. These random subsets are called bootstrap samples, and these samples have same number of observations or equal size. Each bootstrap sample is then inputted to a single learning estimator for training. The procedure of KBAG can be summarized in the following steps:

- Choose a kind of single learning models to be employed in the bagging, such as MLP, SVM, or DT.
- Choose a number of base estimators constructed from the selected single learning model, denoted as L . It is also equivalent to the number of bootstrap samples.
- For each base estimator, we generate a bootstrap sample and then input it to the estimator for training. As a result, we have L independent trained estimators to make predictions.
- The predictions from all base estimators are aggregated into a single prediction by using the Kalman filter. This is the major difference of KBAG compared to a traditional bagging.

Fig. 2a shows the overall flow chart of KBAG. We denote KBAG with the single learning model MLP, SVM, or DT as KBAG-MLP, KBAG-SVM, or KBAG-DT, respectively. Moreover, with regard to the training phase, KBAG requires tuning a numbers of hyper-parameters such as the number of base estimators, and the bootstrap sample size. In our experiments, we used a small number of bootstrap samples or base estimators ($L \in \{5, 10\}$) in order to reduce the training time. And each bootstrap sample S_i ($i \in [1, L]$) has the size equal to 25%, 50%, or 100% of the training data size.

Based on KBAG, we further develop a new Kalman filter-based ensemble, named DKBAG, in which a Kalman filter-based ensemble of KBAG estimators is constructed. The overall flow chart of DKBAG is illustrated in **Fig. 2b**. The Kalman filters are employed in both two layers of DKBAG: one inside each KBAG, and the final one outside all KBAG estimators. This architecture of DKBAG could increase the efficiency of Kalman filters in reducing the prediction error due to the double-stage filtering of predicted outputs. Each inner Kalman filter of a KBAG is used to smoothly fuse predictions from the base estimators. Then the outer Kalman filter would again aggregate predictions from all KBAGs into a final prediction. In addition, for each KBAG estimator, we use a 5-fold cross-validated grid-search to find its optimal hyper-parameters such as the number of base estimators L , and the bootstrap sample size. Our proposed DKBAG is similar to KESLM - the Kalman filter-based ensemble of single machine learning models (see Section 2.3 for details). The main difference is that DKBAG employs KBAG estimators as ensemble members whereas KESLM uses single learning estimators.

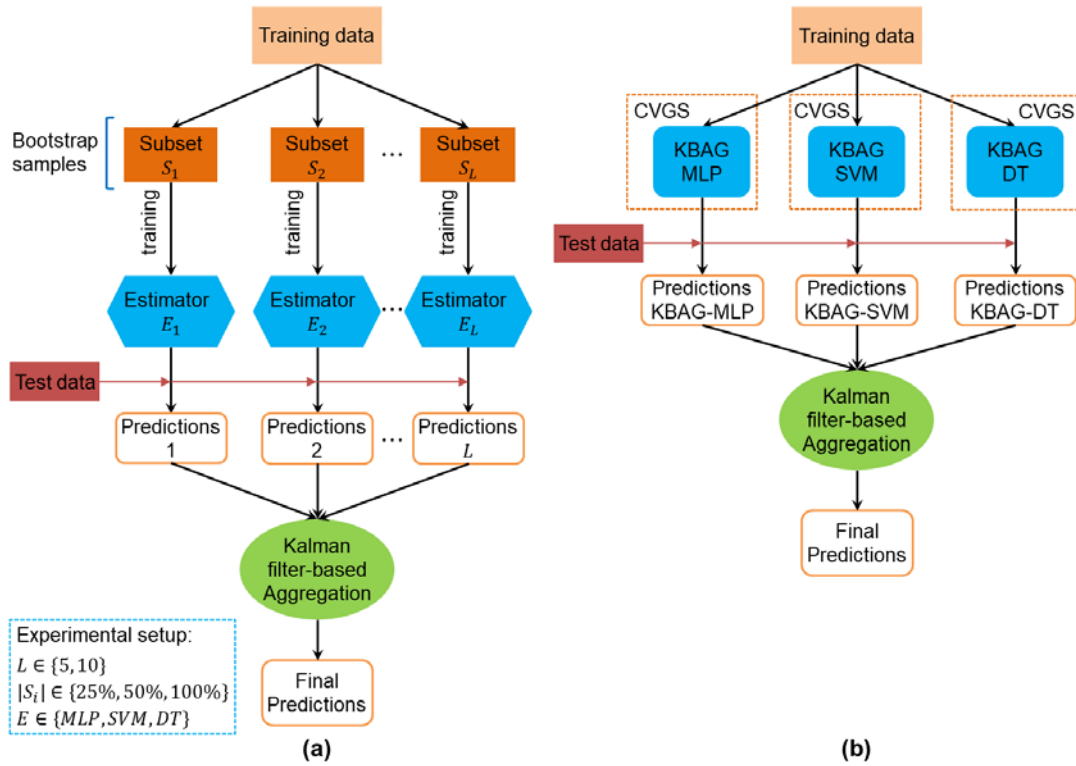


Fig. 2. Overall flow chart of the (a) Kalman filter-based Bagging model (KBAG) and (b) a Kalman filter-based ensemble of KBAGs (DKBAG). In (a), we employ base estimators by using only MLP, SVM, or DT. $|S_i|$ is the proportion of the subset or bootstrap sample i in the training data ($i \in [1, L]$). In (b), we conduct a simple DKBAG of only three learning models: KBAG-MLP, KBAG-SVM, and KBAG-DT.

3.3 GN-DAFC-DKBAG, our proposed approach of combining optimal noise injection and DKBAG

In this section, we utilize the above proposed methods GN-DAFC and DKBAG to further develop a novel approach for RUL predictions, named GN-DAFC-DKBAG. The new approach comprises a noise adjustment scheme along with DKBAG. The framework of our proposed approach is illustrated in Fig. 3. Our approach is conducted in two continuous phases: the noise adjustment phase and the training/testing phase. In the first phase, an GN-DAFC-based exhaustive searching strategy is employed to find optimal Gaussian noise-related parameters such as the range of noise intensity [low_thresh, up_thresh] and the number of inserted noisy observations/features - p and q , respectively. This searching strategy is similar to the cross-validated grid-search (CVGS) introduced in Section 2.5, and is made up of the following steps:

- Create all possible combinations of noise-related parameters values, and iterate over the combinations to compute their prediction performance.
- Each combination nc along with the training data is inputted into the GN-DAFC method to generate a corresponding noise-injected data. Then the noise-injected data is used in a CVGS strategy to obtain a CV score for a simple DKBAG of only three learning models – KBAG-MLP, KBAG-SVM, and KBAG-DT.

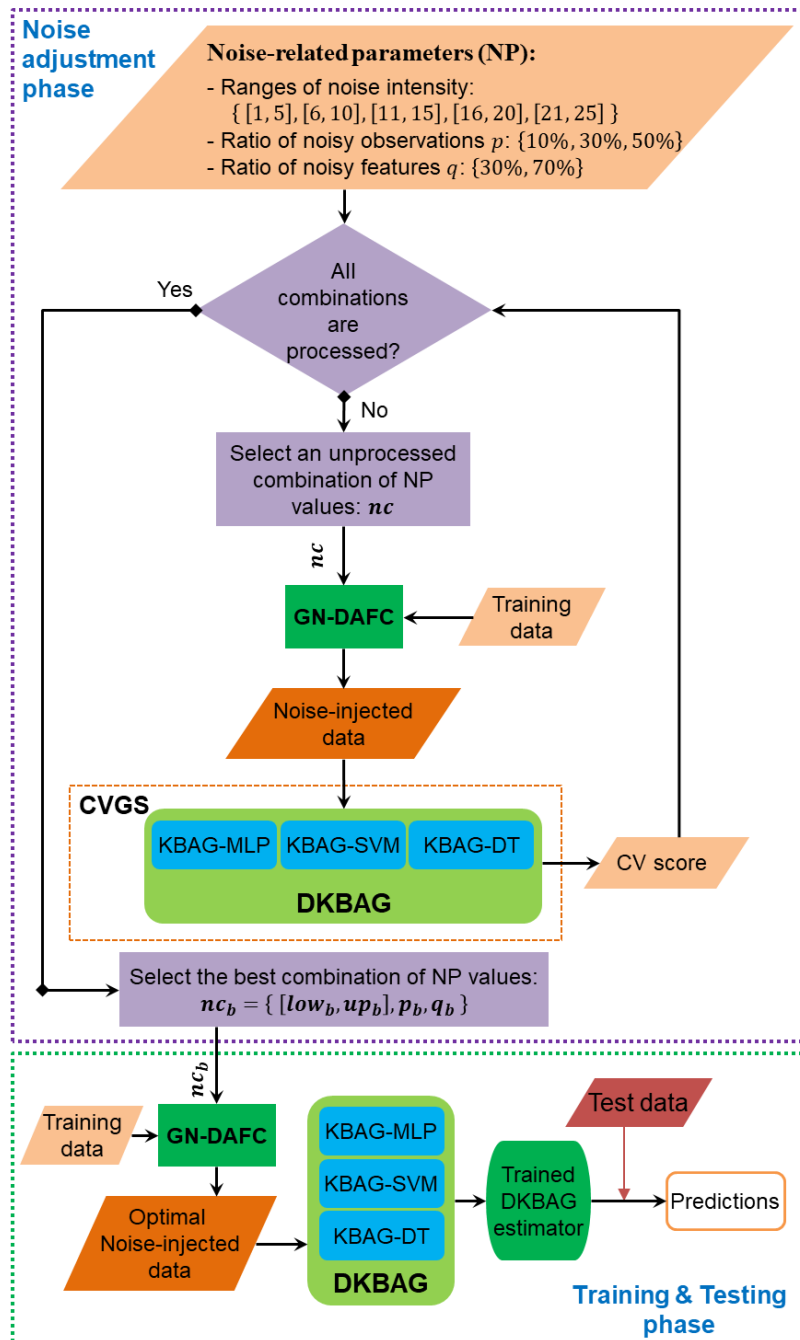


Fig. 3. Overall framework of our approach, GN-DAFC-DKBAG. The approach comprises two continuous phases: the noise adjustment phase and the training/testing phase.

- After finish iterating all combinations, the best combination nc_b is selected corresponding to the best CV score.

Though other previous studies have not provided a specific strategy to select optimal noise-related parameters [19] [20] [21] [22] [23]. The GN-DAFC-based searching strategy of

optimal noise-injected data is the main novel point in our approach, and it also plays a crucial role in the prediction performance of the approach.

As shown in Fig. 3, we apply five ranges of noise intensity for the GN-DAFC-based searching strategy. The ratio of inserted noisy observations is equal to 10%, 30%, or 50% of the training observations size. The ratio of added noisy features is equal to 30% or 70% of the number of features in the training data. In this work, we only specify a small number of ratios to reduce the searching time.

In the second phase, the best combination nc_b along with the training data is again inputted into the GN-DAFC method to generate an optimal noise-injected training data. The optimal noise-injected data is used to train the simple DKBAG, and then a trained DKBAG estimator is obtained for predicting test data later. The DKBAG method used in both two phases also takes an important role in our approach due to the strong reduction of prediction error induced by the two layers of Kalman filters inside it.

4. Results

To evaluate our proposed approach, we compared it to the traditional Kalman filter-based ensemble of single learning models (KESLM), DKBAG with only original training data (original DKBAG), and also two previous advanced ensemble learning approaches over the NASA C-MAPSS benchmark dataset. The experiments have been conducted on a personal computer with Intel® Core™ i3 CPU@2.00GHz, 4.00 GB RAM, and Windows 10. To implement the experiments, the programming language used was Python 3.11.3, together with some additional libraries such as scikit-learn, numpy, filterpy, and so on.

4.1 The NASA C-MAPSS dataset

In this study, our proposed approach is validated on the degradation dataset of the aircraft turbofan engine, which is generated by using a commercial modular aero-propulsion system simulation (C-MAPSS) [31]. The dataset is split into four sub-datasets, as illustrated in Table 1. Each sub-dataset is further divided into the training and testing sets, in which consists of a number of trajectories. Each trajectory is a multivariate time series or the life-cycles of an aircraft engine. All engines function in normal condition at the beginning, and then start to degrade over time. Moreover, the training sets include the full degradation to the end of the engines, whereas the testing sets only contain the partial degradation of engines. Our challenge is to estimate the accurate RUL value for each engine in the testing sets.

Each dataset is organized as an N -by-26 matrix where N denotes the number of data points or observations in the dataset. Each row is a single operational cycle and each column means an input feature. The dataset has 26 different features: identity of the engine, time step (in cycles), three operational conditions, and 21 sensor measurements. The three input features of operational conditions define the operational mode of an engine. There exist six operational modes that significantly affect the engine execution in FD002 and FD004 sub-datasets, though

Table 1. The NASA C-MAPSS dataset of turbofan engines

Description	FD001	FD002	FD003	FD004
No. of Training trajectories	100	260	100	249
No. of Testing trajectories	100	259	100	248
No. of operational modes	1	6	1	6

only a single operational mode exists in FD001 and FD003 sub-datasets [11]. Therefore, the operational mode is used to create additional six input features defining the number of cycles executed in their respective operational mode from the starting of the time series [11]. Moreover, all input features were normalized based on the operational modes as in [11]. We also remove the input features containing only one constant value. As a result, only 17, 18 input features are remained in FD001 and FD003 sub-datasets, respectively. And all input features are retained in FD002 and FD004 sub-datasets.

4.2 Experiments setup

In this section, we present the experimental setup for the RUL prediction of our proposed approach (GN-DAFC-DKBAG) and three other introduced ensemble approaches such as KESLM with original training data (named as “original KESLM”), KESLM combined with GN-DAFC (named as GN-DAFC-KESLM), and DKBAG with original training data (named as “original DKBAG”). The GN-DAFC-KESLM approach has similar architecture to our approach GN-DAFC-DKBAG; although, the former employs a simple KESLM model (consisting of three single learning models MLP, DT, and SVM) whereas the latter employs a simple DKBAG model (consisting of three KBAG models KBAG-MLP, KBAG-SVM, and KBAG-DT). We apply a number of 5-fold cross-validated grid-searches to find optimal hyper-parameters values of KESLM and DKBAG models in these approaches. Lists of hyper-parameters values are specified in Table 2. Detail explanations of hyper-parameters are presented as follows:

- MLP: We simply apply the MLP model with only one single hidden layer to reduce the training time. The solver for weight optimization is determined between an optimization algorithm in the group of quasi-Newton methods – the Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (“lbfgs”) [32] and a stochastic gradient-based optimizer (“adam”) [33]. The activation function type is also selected between the hyperbolic tangent function (“tanh”) and the rectified linear unit function (“relu”).
- SVM: The penalty parameter C is used to specify the degree of the acceptance of incorrect predictions in the training data.
- DT: The maximum depth of a decision tree is a stopping condition that limits the number of splits that can be conducted in the tree.
- KBAG: The number of bootstrap samples or base estimators is only selected between 5 and 10 values in order to reduce the training time. The number of observations to draw from the training data to train each base estimator (size of each bootstrap sample) is equal to 25%, 50%, or 100% of the training data size.

Table 2. Lists of hyper-parameters values used in KESLM/DKBAG-related approaches

Related approach	Model	Hyper-parameter	List of values
original KESLM & GN-DAFC-KESLM	MLP	No. of neurons in the hidden layer	{ 5, 10, 15, 20 }
		Type of weight optimization method	{ <i>adam</i> , <i>lbfgs</i> }
		Type of activation function	{ <i>relu</i> , <i>tanh</i> }
	SVM	Penalty parameter C	{1, 2, 4, 8, 16, 32}
	DT	Maximum depth of the tree	[1 – 10]
original DKBAG & GN-DAFC-DKBAG	KBAG	No. of bootstrap samples	{5, 10}
		Size of each bootstrap sample	{25%, 50%, 100%}

Table 3. Lists of Gaussian noise-related parameters values used in GN-DAFC-KESLM and GN-DAFC-DKBAG approaches

Parameter	List of values
Range of noise intensity	{ [1, 5], [6, 10], [11, 15], [16, 20], [21, 25] }
Ratio of inserted noisy observations p	{ 10%, 30%, 50% }
Ratio of inserted noisy features q	{ 30%, 70% }

In addition to these above hyper-parameters, GN-DAFC-KESLM and GN-DAFC-DKBAG approaches also need to find the optimal Gaussian noise-related parameters values as in the noise adjustment phase. **Table 3** summarizes lists of the noise-related parameters values. We apply five ranges of noise intensity for GN-DAFC method. The ratio of inserted noisy observations is equal to 10%, 30%, or 50% of the training observations size. The ratio of inserted noisy features is equal to 30% or 70% of the number of features in the training data. We only specify a small number of ratios to reduce the searching time.

4.3 Performance comparisons between GN-DAFC-DKBAG and other ensemble approaches

Based on the setup of the previous section, we compare the RUL prediction performance of our proposed approach (GN-DAFC-DKBAG) to the other three introduced ensemble approaches such as original KESLM, GN-DAFC-KESLM, and original DKBAG. All following experiments were executed over 50 trials to achieve general and reliable results. Firstly, the correlation between the training and test performance of GN-DAFC-DKBAG and other three approaches is investigated (**Fig. 4**). For all datasets, results of the approaches did not showed positive correlations between training MSE and test MSE values. It means that a better approach in the training dataset does not ensure a better performance over the test dataset. Remarkably, our approach (GN-DAFC-DKBAG) has the best performance over all the test datasets, while it was not best over all the training datasets. We further plot the average and the standard deviation of the MSE values in all the test datasets (**Fig. 5**). As illustrated in the figure, the GN-DAFC-DKBAG approach obtained significantly better RUL predictions than the other three approaches in all the test datasets (all p-values < 0.0001). The GN-DAFC-KESLM approach yielded obviously better results in comparison with the original KESLM (all p-values < 0.0001) and similar RUL prediction performance to the original DKBAG approach in all the test datasets. In these cases, the effect of optimal noise injection is clearly observed. It also turns out that the optimal noise-injected dataset obtained from the GN-DAFC method can improve the prediction performance of both the ensemble of single learning models (KESLM) and the ensemble of bagging models (DKBAG). We also found that the original DKBAG approach achieved significantly better results than the original KESLM one in all the test datasets (all p-values < 0.0001); though, the optimal noise-injected dataset used in the training phase could help the original KESLM approach to gain similar prediction performance with the original DKBAG one. Thus, the GN-DAFC-based searching strategy of optimal noise-injected data could be used in a more flexible way for learning model selections. Furthermore, we examined the best combinations of noise-related parameters values found by GN-DAFC-KESLM and GN-DAFC-DKBAG approaches (**Table 4**), and observed that they are varied across the given approaches and datasets. The corresponding best CV score of each best combination is also presented in the table. The largest range of noise

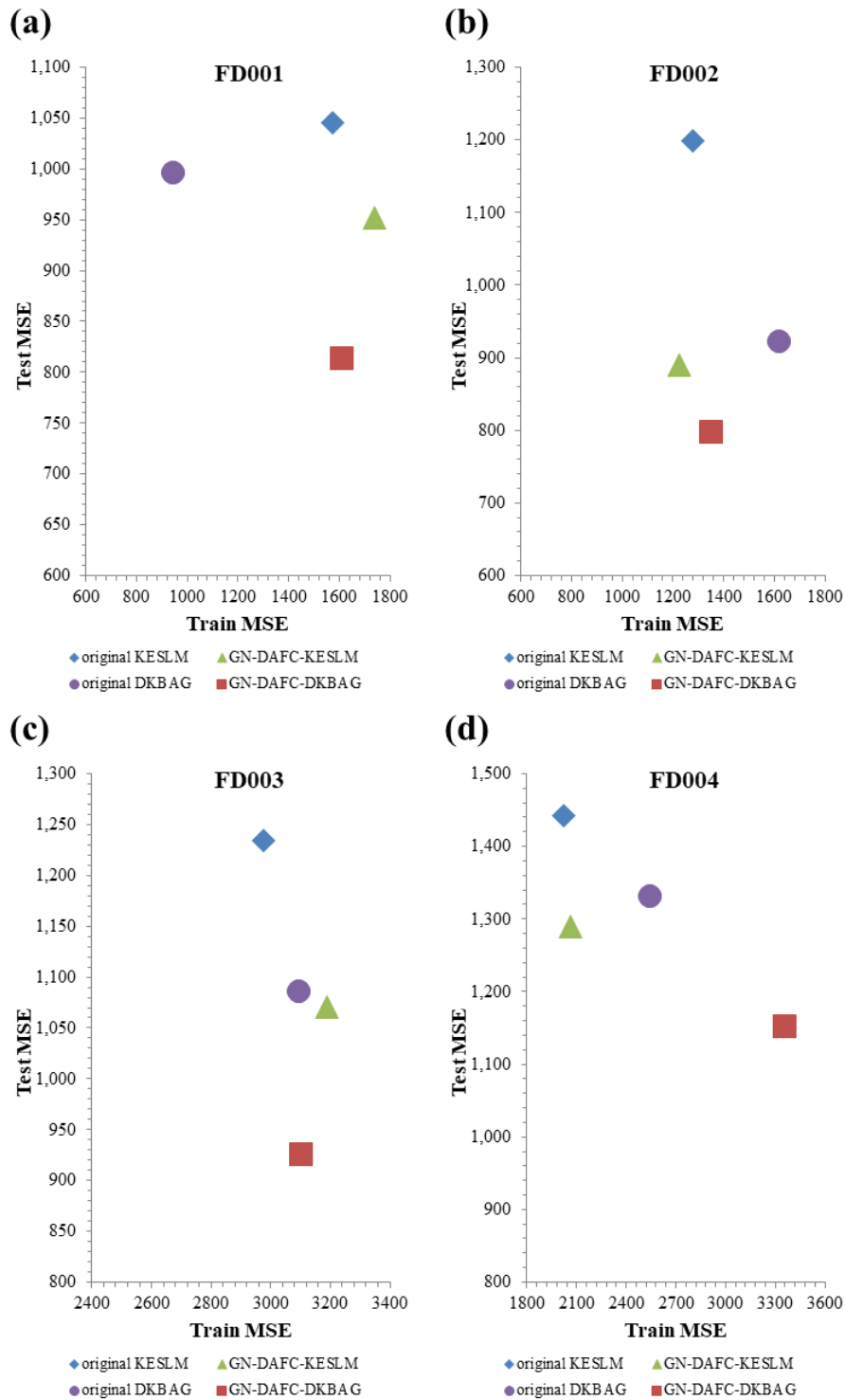


Fig. 4. Correlation between the training and test performance by GN-DAFC-DKBAG and other three approaches. (a-d) Results in FD001, FD002, FD003, and FD004 datasets, respectively. There are no positive correlations between training MSE and test MSE values in all datasets.

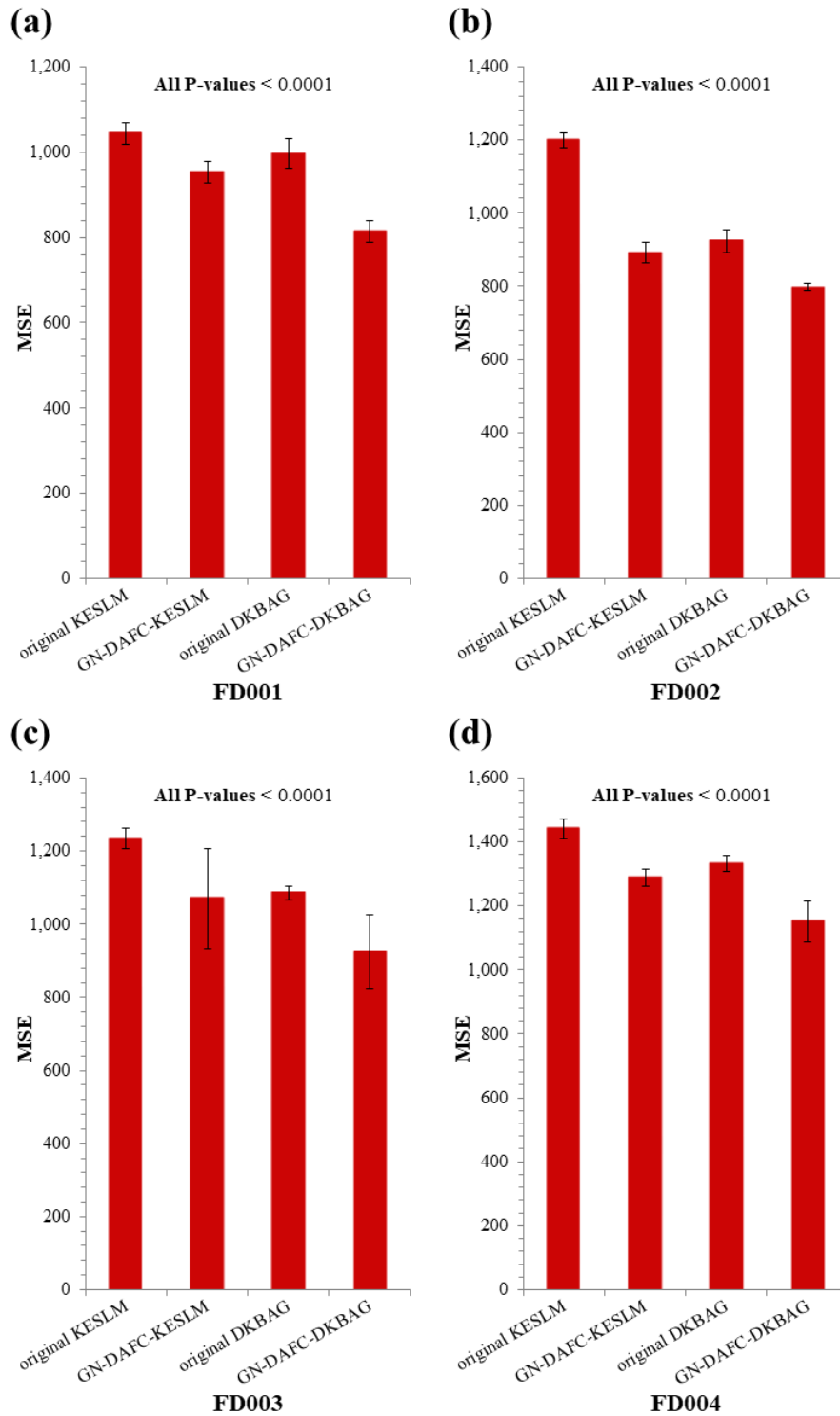


Fig. 5. Performance comparison of GN-DAFC-DKBAG and other three approaches. (a-d) Results in FD001, FD002, FD003, and FD004 datasets, respectively. GN-DAFC-DKBAG obtains the best RUL predictions, GN-DAFC-KESLM and original DKBAG have similar performance, and original KESLM is the worst in all the test datasets (all p-values <math>< 0.0001</math>).

intensity [21, 25] does not exist in all the best combinations, and only three of eight best combinations include the second largest range of noise intensity [16, 20]. The largest ratio of inserted noisy observations “50%” occurs in only one best combination, and the largest ratio of inserted noisy features “70%” exists in three of eight best combinations. These cases mean that it is very difficult to train a learning model based on a dataset of too strong noise intensity or large ratio of inserted noisy observations/features. Similarly, we also found that the smallest range of noise intensity [1, 5] exists in only three of eight best combinations. The small and medium ratios of inserted noisy observations (10%, 30%) are almost equally distributed in most of best combinations (seven of eight); and the smallest ratio of inserted noisy features “30%” also exists in most of best combinations (five of eight). It turns out that too weak noise intensity has little influence on a learning model; though a small or medium ratio of inserted noisy observations/features still strongly affect the learning model. The CV score is the negation of the average of MSE values obtained from all validation folds in the 5-fold cross-validation. These CV scores obviously have large variability due to the difficulty in training on a noise-injected dataset. The investigation in **Table 4** could help us intensely understand the effects of noise intensity/quantity on a learning model. In summary, the GN-DAFC-KESLM and GN-DAFC-DKBAG approaches have competently explored a variety of noise-related parameters’ combinations to achieve the optimal noise-injected dataset for further training a RUL prediction model.

In order to show the applicability of our proposed approach in real systems, we analyzed the running time of all approaches on our system as mentioned above. We note that the running time is measured for all phases in an approach (data preprocessing, finding the best noise-related parameters’ combinations if required, training, testing). As expected, the running time of the original KESLM/DKBAG approaches is very short in comparison to that of the GN-DAFC-KESLM and GN-DAFC-DKBAG ones (**Fig. 6**). It dues to the phase of searching optimal noise-injected training dataset in these approaches is time consuming. The noisy expansion in both the observation and feature spaces of the training dataset also requires a longer training time. However, the running time of our GN-DAFC-DKBAG approach is still in a practical way considering the enhancement of RUL prediction performance.

Finally, we further compared our proposed approach with two previous advanced ensemble approaches: a heuristic Kalman filter ensemble [11] and an ensemble of genetic algorithms [12]. In [11], a Kalman filter-based ensemble of multiple neural network models along with a heuristic ensemble selection was proposed to predict RUL. The heuristic ensemble selection aims to find a good subset of learning models in a proper running time. It reduces the search space while still provides a broad search path. We also conducted this heuristic Kalman filter ensemble on the NASA C-MAPSS dataset. In [12], the authors proposed a Kalman filter-based ensemble of multiple optimal learning models obtained from genetic algorithm searches. This previous study already investigated the NASA C-MAPSS dataset. As shown in **Table 5**, our approach achieves better performance than the two previous ensemble ones. In summary, our approach of combining optimal noise-injected data and DKBAG could be considered as a potential solution for RUL estimation of machinery systems.

5. Discussions

Latest advances in artificial intelligence and computational methods help us analyze and predict operations of machinery systems. Many data-driven prognostics approaches have been developed and achieved good improvement in health-state/RUL estimation of various systems, such as lithium-ion battery systems [34] [35] [36] [37] [38], supercapacitors [39] [40] [41],

turbofan engines [6] [7] [9]. In this work, we proposed a novel RUL prediction approach of combining optimal noise injection and Kalman filters-based bagging. Firstly, we developed GN-DAFC, a new procedure to inject Gaussian noises into both observation and feature spaces of an original dataset. Secondly, we proposed KBAG, an improved version of the bagging method based on Kalman filter averaging. The improvement is that KBAG utilizes a Kalman filter-based averaging method rather than a classical averaging one. Thirdly, we further developed DKBAG, a Kalman filter ensemble of KBAGs. Finally, we proposed GN-DAFC-DKBAG, a novel RUL prediction approach that combines optimal noise-injected training data with DKBAG.

Table 4. Best combinations of Gaussian noise-related parameters values used in GN-DAFC-KESLM and GN-DAFC-DKBAG approaches

Dataset	Approach	Range of noise intensity	Ratio of inserted noisy observations p	Ratio of inserted noisy features q	Absolute cross-validation score ($\mu \pm \sigma$)
FD001	GN-DAFC-KESLM	[11, 15]	10%	70%	1536.41 ± 318.67
	GN-DAFC-DKBAG	[11, 15]	10%	30%	1568.29 ± 299.28
FD002	GN-DAFC-KESLM	[1, 5]	30%	70%	2374.84 ± 151.68
	GN-DAFC-DKBAG	[16, 20]	30%	30%	2724.17 ± 354.29
FD003	GN-DAFC-KESLM	[1, 5]	10%	30%	5392.35 ± 819.69
	GN-DAFC-DKBAG	[16, 20]	30%	30%	6460.89 ± 1148.29
FD004	GN-DAFC-KESLM	[16, 20]	50%	30%	3974.94 ± 458.80
	GN-DAFC-DKBAG	[1, 5]	10%	70%	6172.86 ± 546.85

Table 5. Performance comparison of our approach GN-DAFC-DKBAG with two other advanced ensemble approaches on the NASA C-MAPSS dataset

Approach	Test MSE ($\mu \pm \sigma$)			
	FD001	FD002	FD003	FD004
a heuristic Kalman filter ensemble [11]	1062.86 ± 81.89	1150.54 ± 58.94	1352.16 ± 152.88	1413.13 ± 54.61
an ensemble of genetic algorithms [12]	818.36 ± 132.64	1111.05 ± 39.51	1152.99 ± 114.87	1198.07 ± 133.35
GN-DAFC-DKBAG	813.30 ± 25.54	797.54 ± 9.31	925.33 ± 101.50	1151.70 ± 64.83

GN-DAFC could be considered as a structural combination of data augmentation and feature construction methods. The main difference of GN-DAFC with existing noise injection approaches is that the existing ones insert noise only in the observation space of the training data or focus only in the data augmentation/modification [19] [20] [21] [22] [23]. By repeating the procedure GN-DAFC, different noisy datasets can be generated based on the original dataset. Thus, it is an effective and convenient way to increase the generalization of a learning approach. GN-DAFC plays an essential role in our proposed RUL prediction approach - GN-DAFC-DKBAG. The GN-DAFC-based searching strategy of optimal noise-injected training data is the crucial point of GN-DAFC-DKBAG. The searching strategy would decide the prediction performance of the followed DKBAG model. Effectiveness of the GN-DAFC-based searching strategy was proved in the Results section. As shown in Fig. 5,

GN-DAFC-DKBAG achieves the best RUL predictions, GN-DAFC-KESLM and original DKBAG have similar performance, and original KESLM is the worst in all the test datasets (all p-values < 0.0001). It means that the GN-DAFC-based searching strategy can enhance the prediction performance of both DKBAG and KESLM methods.

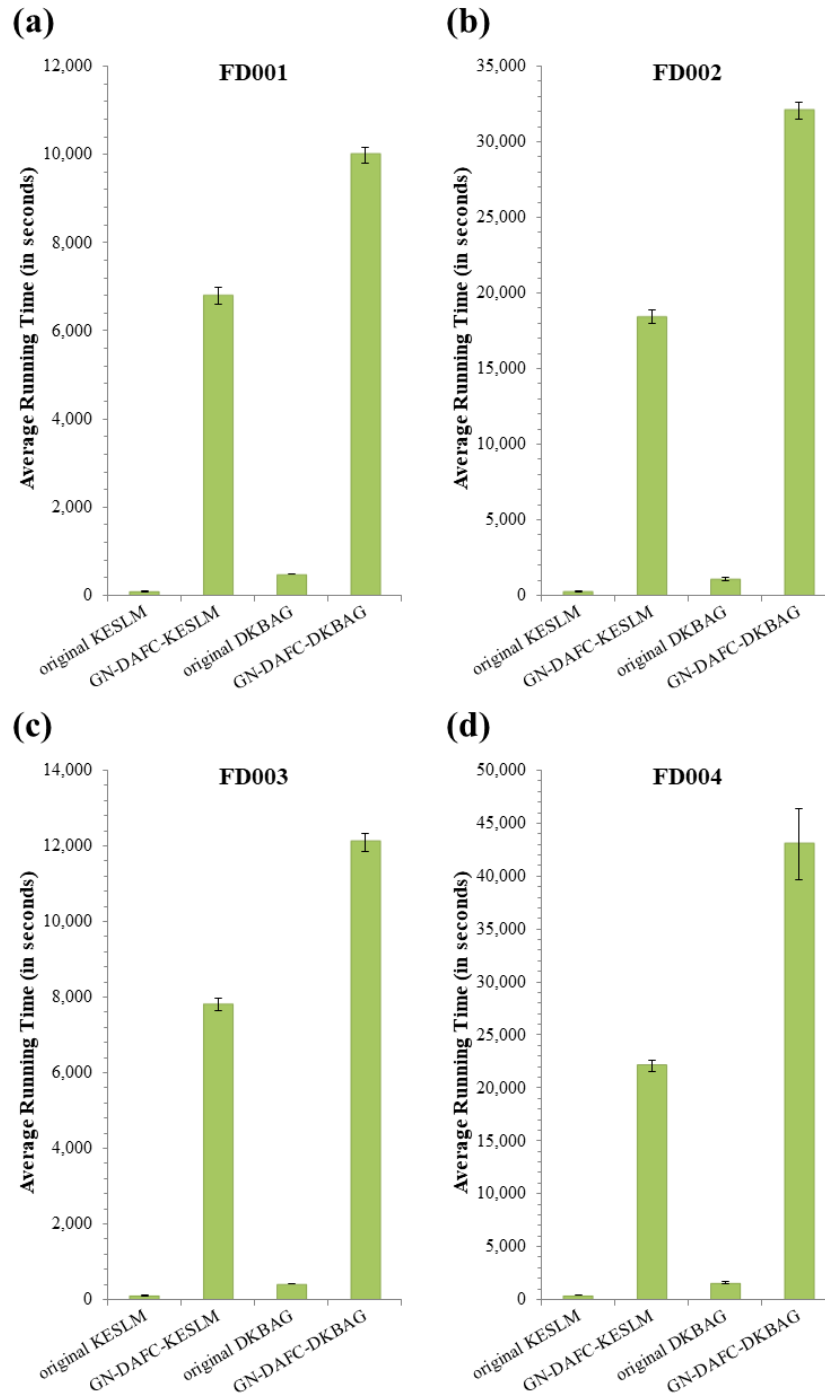


Fig. 6. Running time comparison of GN-DAFC-DKBAG and other three approaches. (a-d) Results in FD001, FD002, FD003, and FD004 datasets, respectively. Our GN-DAFC-DKBAG approach has the longest running time, though it is still in a practical way.

Regarding DKBAG, the Kalman filters are employed in both two layers of DKBAG: one inside each KBAG, and the final one outside all KBAG estimators. This double-stage architecture of DKBAG could increase the efficiency of Kalman filters in reducing the prediction error due to the two continuous filtering of predicted outputs, and then could enhance the model performance and stability. DKBAG outperforms over the conventional methods as shown in the Results section. **Fig. 5** showed that the original DKBAG has significantly better performance than the original KESLM (p-values < 0.0001), and the GN-DAFC-DKBAG also outperforms the GN-DAFC-KESLM (p-values < 0.0001).

To provide deeper insights, we further investigated the best combinations of noise-related parameters values found by GN-DAFC-KESLM and GN-DAFC-DKBAG approaches as in **Table 4**. Results of this investigation showed that an appropriate medium noise intensity and small/medium ratio of inserted noisy observations/features could obviously improve a learning model. Finally, we compared our proposed approach with two previous advanced ensemble approaches: a heuristic Kalman filter ensemble and an ensemble of genetic algorithms. Experimental results also indicated that our proposed approach achieved better prediction performance than the two previous ensemble ones.

6. Conclusion

In contrast to most previous studies, we utilized noise injection to improve the RUL prediction performance of learning models. We firstly proposed GN-DAFC, which is a new noise injection method to insert Gaussian noise into both observation and feature spaces of an original training dataset. Secondly, we developed a new ensemble learning method by combining Kalman filters and bagging methods, named DKBAG. Finally, we proposed a new RUL prediction approach GN-DAFC-DKBAG in which the optimal noise-injected training dataset was determined by a GN-DAFC-based searching strategy and then inputted to the DKBAG model. GN-DAFC-DKBAG was also considered as a new noise-based data augmentation/feature construction approach, where it optimizes parameters of noise intensity and amount of inserted noisy observations/features. The effectiveness of GN-DAFC-DKBAG was then validated on the NASA C-MAPSS dataset. Our proposed approach outperforms the traditional approach using a Kalman filter-based ensemble of single learning models (KESLM) and also the original DKBAG. As shown in the results, the combination of optimal inserted noisy data and the Kalman filter-based bagging ensemble can improve the generalization performance of leaning models and avoid the over-fitting problem. The optimal noise-injected data can also enhance the prediction performance of the KESLM approach. These results suggest that the optimal noise-injected dataset can be utilized in a more flexible way for machine learning model selections. Moreover, a proper medium noise intensity and small/medium number of inserted noisy observations/features can significantly improve a learning model as observed in the found best combinations of noise-related parameters. Furthermore, our approach was also compared with two previous advanced ensemble approaches, and the results showed that our approach achieves better performance than the two previous ones. Despite the improved performance, a parallel and heuristic version of the GN-DAFC-DKBAG approach could be developed to reduce the running time. Other types of single or ensemble learning models can be employed in our approach for further investigations, such as long short-term memory neural network, boosting and stacking ensembles. Finally, various types of noise can be additionally examined in our approach such as uniform noise, impulsive noise, and exponential noise.

References

- [1] Y. Yang, "A machine-learning prediction method of lithium-ion battery life based on charge process for different applications," *Applied Energy*, vol. 292, Jun. 2021, Art. no. 116897. [Article \(CrossRef Link\)](#)
- [2] Z. Tong, J. Miao, S. Tong, and Y. Lu, "Early prediction of remaining useful life for Lithium-ion batteries based on a hybrid machine learning method," *J. Clean. Prod.*, vol. 317, Oct. 2021, Art. no. 128265. [Article \(CrossRef Link\)](#)
- [3] M. Motahari-Nezhad and S. M. Jafari, "Comparison of MLP and RBF neural networks for bearing remaining useful life prediction based on acoustic emission," *Proc. Inst. Mech. Eng. J: J. Eng. Tribol.*, vol. 237, no. 1, pp. 129-148, Jan. 2023. [Article \(CrossRef Link\)](#)
- [4] Z. Liu, H. Wang, M. Hao, and D. Wu, "Prediction of RUL of Lubricating Oil Based on Information Entropy and SVM," *Lubricants*, vol. 11, no. 3, Mar. 2023, Art. no. 121. [Article \(CrossRef Link\)](#)
- [5] L. Wang, D. Zhou, H. Zhang, W. Zhang, and J. Chen, "Application of Relative Entropy and Gradient Boosting Decision Tree to Fault Prognosis in Electronic Circuits," *Symmetry*, vol. 10, no. 10, Oct. 2018, Art. no. 495. [Article \(CrossRef Link\)](#)
- [6] S. K. Singh, S. Kumar, and J. P. Dwivedi, "A novel soft computing method for engine RUL prediction," *Multimed. Tools. Appl.*, vol. 78, pp. 4065–4087, Feb. 2019. [Article \(CrossRef Link\)](#)
- [7] Y. Cheng, J. Wu, H. Zhu, S. W. Or, and X. Shao, "Remaining Useful Life Prognosis Based on Ensemble Long Short-Term Memory Neural Network," *IEEE Trans. Instrum. Meas.*, vol. 70, 2020, Art. no. 3503912. [Article \(CrossRef Link\)](#)
- [8] L. Liu, Z. Zhang, Z. Qu, and A. Bell, "Remaining Useful Life Prediction for a Catenary, Utilizing Bayesian Optimization of Stacking," *Electronics*, vol. 12, no. 7, Apr. 2023, Art. no. 1744. [Article \(CrossRef Link\)](#)
- [9] B. A. Ture, A. Akbulut, A. H. Zaim, and C. Catal, "Stacking-based ensemble learning for remaining useful life estimation," *Soft Comput.*, 2023. [Article \(CrossRef Link\)](#)
- [10] G. Wang, Z. Lyu, and X. Li, "An Optimized Random Forest Regression Model for Li-Ion Battery Prognostics and Health Management," *Batteries*, vol. 9, no. 6, Jun. 2023, Art. no. 332. [Article \(CrossRef Link\)](#)
- [11] L. Peel, "Data driven prognostics using a Kalman filter ensemble of neural network models," in *Proc. of ICPHM*, Denver, CO, USA, 2008. [Article \(CrossRef Link\)](#)
- [12] H-C. Trinh and Y-K. Kwon, "A Data-Independent Genetic Algorithm Framework for Fault-Type Classification and Remaining Useful Life Prediction," *Applied Sciences*, vol. 10, no. 1, Jan. 2020, Art. no. 368. [Article \(CrossRef Link\)](#)
- [13] K. Audhkhasi, O. Osoba, and B. Kosko, "Noise-enhanced convolutional neural networks," *Neural Networks*, vol. 78, pp. 15-23, Jun. 2016. [Article \(CrossRef Link\)](#)
- [14] O. Adigun and B. Kosko, "Noise-boosted bidirectional backpropagation and adversarial learning," *Neural Networks*, vol. 120, pp. 9-31, Dec. 2019. [Article \(CrossRef Link\)](#)
- [15] O. Osoba and B. Kosko, "Noise-enhanced clustering and competitive learning algorithms," *Neural Networks*, vol. 37, pp. 132-140, Jan. 2013. [Article \(CrossRef Link\)](#)
- [16] H. Zheng, Z. Zhou, and J. Chen, "RLSTM: A New Framework of Stock Prediction by Using Random Noise for Overfitting Prevention," *Comput. Intell. Neurosci.*, vol. 2021, 2021, Art. no. 8865816. [Article \(CrossRef Link\)](#)
- [17] S. S. Raju, B. Wang, K. Mehta, M. Xiao, Y. Zhang, and H-Y. Wong, "Application of Noise to Avoid Overfitting in TCAD Augmented Machine Learning," in *Proc. of SISPAD*, Kobe, Japan, 2020. [Article \(CrossRef Link\)](#)
- [18] E. J. Snider, S. I. Hernandez-Torres, and R. Hennessey, "Using Ultrasound Image Augmentation and Ensemble Predictions to Prevent Machine-Learning Model Overfitting," *Diagnostics*, vol. 13, no. 3, Feb. 2023, Art. no. 417. [Article \(CrossRef Link\)](#)
- [19] L. Xiao, J. Tang, X. Zhang, and T. Xia, "Weak fault detection in rotating machineries by using vibrational resonance and coupled varying-stable nonlinear systems," *J. Sound Vib.*, vol. 478, Jul. 2020, Art. no. 115355. [Article \(CrossRef Link\)](#)

- [20] L. Xiao, R. Bajric, J. Zhao, J. Tang, and X. Zhang, "An adaptive vibrational resonance method based on cascaded varying stable-state nonlinear systems and its application in rotating machine fault detection," *Nonlinear Dyn.*, vol. 103, pp. 715–739, Jan. 2021. [Article \(CrossRef Link\)](#)
- [21] Z. Qiao, Y. Lei, and N. Li, "Applications of stochastic resonance to machinery fault detection: A review and tutorial," *Mech. Syst. Signal Process.*, vol. 122, pp. 502-536, May. 2019. [Article \(CrossRef Link\)](#)
- [22] L. Xiao, X. Zhang, S. Lu, T. Xia, and L. Xi, "A novel weak-fault detection technique for rolling element bearing based on vibrational resonance," *J. Sound Vib.*, vol. 438, pp. 490-505, Jan. 2019. [Article \(CrossRef Link\)](#)
- [23] L. Xiao, J. Tang, X. Zhang, E. Bechhofer, and S. Ding, "Remaining useful life prediction based on intentional noise injection and feature reconstruction," *Reliab. Eng. Syst. Saf.*, vol. 215, Nov. 2021, Art. no. 107871. [Article \(CrossRef Link\)](#)
- [24] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183-197, Jul. 1991. [Article \(CrossRef Link\)](#)
- [25] S. B. Kotsiantis, "Decision trees: a recent overview," *Artif. Intell. Rev.*, vol. 39, pp. 261-283, Apr. 2013. [Article \(CrossRef Link\)](#)
- [26] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. App.*, vol. 13, no. 4, pp. 18-28, Jul. 1998. [Article \(CrossRef Link\)](#)
- [27] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123-140, Aug. 1996. [Article \(CrossRef Link\)](#)
- [28] M. Hayslep, E. Keedwell, R. Farmani, and J. Pocock, "Understanding district metered area level leakage using explainable machine learning," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 1136, no. 1, Jan. 2023, Art. no. 012040. [Article \(CrossRef Link\)](#)
- [29] G. Oluchi Anyanwu, C. I. Nwakanma, J-M. Lee, and D-S. Kim, "Optimization of RBF-SVM Kernel Using Grid Search Algorithm for DDoS Attack Detection in SDN-Based VANET," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8477-8490, May. 2023. [Article \(CrossRef Link\)](#)
- [30] Y. Du, A. R. Rafferty, F. M. McAuliffe, L. Wei, and C. Mooney, "An explainable machine learning-based clinical decision support system for prediction of gestational diabetes mellitus," *Sci. Rep.*, vol. 12, Jan. 2022, Art. no. 1170. [Article \(CrossRef Link\)](#)
- [31] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. of ICPHM*, Denver, CO, USA, 2008. [Article \(CrossRef Link\)](#)
- [32] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Math. Comp.*, vol. 24, no. 111, pp. 647-656, Jul. 1970. [Article \(CrossRef Link\)](#)
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint*, 2014. [Article \(CrossRef Link\)](#)
- [34] X. Sun, Y. Zhang, Y. Zhang, L. Wang, and K. Wang, "Summary of Health-State Estimation of Lithium-Ion Batteries Based on Electrochemical Impedance Spectroscopy," *Energies*, vol. 16, no. 15, Aug. 2023, Art. no. 5682. [Article \(CrossRef Link\)](#)
- [35] M. Zhang, D. Yang, J. Du, H. Sun, L. Li, L. Wang, and K. Wang, "A Review of SOH Prediction of Li-Ion Batteries Based on Data-Driven Algorithms," *Energies*, vol. 16, no. 7, Apr. 2023, Art. no. 3167. [Article \(CrossRef Link\)](#)
- [36] P. Li, Z. Zhang, R. Grosu, Z. Deng, J. Hou, Y. Rong, and R. Wu, "An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries," *Renew. sustain. energy rev.*, vol. 156, Mar. 2022, Art. no. 111843. [Article \(CrossRef Link\)](#)
- [37] Z. Deng, L. Xu, H. Liu, X. Hu, Z. Duan, and Y. Xu, "Prognostics of battery capacity based on charging data and data-driven methods for on-road vehicles," *Applied Energy*, vol. 339, Jun. 2023, Art. no. 120954. [Article \(CrossRef Link\)](#)
- [38] W. Guo, L. Yang, Z. Deng, J. Li, and X. Bian, "Rapid online health estimation for lithium-ion batteries based on partial constant-voltage charging segment," *Energy*, vol. 281, Oct. 2023, Art. no. 128320. [Article \(CrossRef Link\)](#)

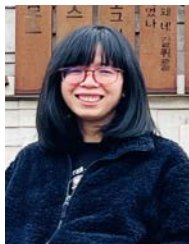
- [39] N. Ma, D. Yang, S. Riaz, L. Wang, and K. Wang, "Aging Mechanism and Models of Supercapacitors: A Review," *Technologies*, vol. 11, no. 2, Apr. 2023, Art. no. 38. [Article \(CrossRef Link\)](#)
- [40] N. Ma, H. Yin, and K. Wang, "Prediction of the Remaining Useful Life of Supercapacitors at Different Temperatures Based on Improved Long Short-Term Memory," *Energies*, vol. 16, no. 14, Jul. 2023, Art. no. 5240. [Article \(CrossRef Link\)](#)
- [41] Z. Yi, Z. Chen, K. Yin, L. Wang, and K. Wang, "Sensing as the key to the safety and sustainability of new energy storage devices," *Prot. Control Mod. Power Syst.*, vol. 8, Jun. 2023, Art. no. 27. [Article \(CrossRef Link\)](#)



Hung-Cuong Trinh received the Ph.D. degree in computer engineering from the University of Ulsan, South Korea, in 2017. Since 2018, he has been a Lecturer and a Researcher in the Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City. His research interests include bioinformatics, artificial intelligence, data science, and machine learning applications.



Van-Huy Pham received the M.S. degree in computer science from the University of Sciences, Ho Chi Minh City, Vietnam, in 2007, and the Ph.D. degree in computer science from the University of Ulsan, South Korea, in 2015. Since 2015, he has been a Lecturer and a Researcher in the Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City. His research interests include artificial intelligence, image processing, computer vision, and deep learning applications.



Anh H. Vo received the M.S. degree in computer science from the University of Sciences, Ho Chi Minh City, Vietnam, in 2015. She is currently pursuing the Ph.D. degree at Department of Computer Engineering, Sejong University, Seoul, South Korea. Her main research interests include image processing, pattern recognition, data science, and computer vision.