Research Article

Design of Fault Prediction System for Electromechanical Sensor Equipment Based on Deep Learning

Yongtao Ding, Hua Wu, and Kaixiang Zhou

Department of Mechanical Engineering, City University of Hong Kong, 999077 Hong Kong SAR, China

Correspondence should be addressed to Yongtao Ding; yongtding2-c@my.cityu.edu.hk

Received 29 November 2021; Accepted 18 January 2022; Published 17 March 2022

Academic Editor: Baiyuan Ding

Copyright © 2022 Yongtao Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increasing complexity, scale, and intelligentization of modern equipment, the maintenance cost of equipment is increasing day by day. Moreover, once an unexpected major failure occurs, it will cause loss and damage to production, economy, and safety. Based on the considerations of system reliability and safety, fault prediction has gradually become a hot topic in the field of reliability. As a new branch of machine learning, deep learning realizes deep abstract feature extraction and expression of complex nonlinear relations by stacking deep neural networks and makes its methods solve bad problems in many traditional machine learning fields. The improvement and excellent results have been achieved. This article first introduces the model structure and working principle of the classic deep learning model noise reduction autoencoder and combines the feature extraction results of the experimental data of electromechanical sensor equipment and the model characteristics to analyze that this type of model can be trained using only normal samples. Under the restriction, the reason for abnormal features can also be correctly filtered. Then, in order to suppress the overfitting of training, the dropout layer is used in this article. The dropout layer will make the hidden layer nodes to be dropped with probability \( p \) according to the set probability. Because the lost nodes are random, each training is equivalent to training a new model. It achieves an effect similar to independently training the model and then superimposing it. Experiments prove that the dropout layer is very effective in solving overfitting. Finally, this paper conducts experimental verification on the improved algorithm. The results show that the improved model has a certain improvement in accuracy under the limited training algebra. Under the same training parameters, the accuracy is increased by approximately 2.44%, and the improved model has a better training effect and can be used for electromechanical effective prediction of sensor equipment failure.

1. Introduction

The structure of large-scale mechanical equipment such as high-speed EMU trains, steam turbines, and aerospace engine units is becoming more and more complex, and the probability of failure is gradually increasing. Fault diagnosis helps the system operate safely and efficiently. However, the traditional planned maintenance needs to stop production and cause high maintenance costs, and no matter how the maintenance interval is set, there is always the problem of insufficient maintenance or overmaintenance. Therefore, in consideration of factors such as reliability, safety, and economy of complex systems, equipment predictive maintenance provides maintenance engineers with a decision-making basis through early fault diagnosis and remaining life prediction so that the transition from planned maintenance to condition-based maintenance can be realized. United Airlines’ predictive maintenance strategy has reduced the maintenance cost of Douglas DC-8 aircraft by 30%, and the fatality rate per 100 million passenger-kilometers has dropped from 0.6 to about 0.2. After adopting the predictive maintenance strategy, the United States’ fifth-generation fighter aircraft not only reduced maintenance costs but also increased the overall reliability by more than 50%. However, the accuracy of fault diagnosis and remaining life prediction depends on the effectiveness of fault feature extraction methods. As an advanced feature extraction tool, deep learning has been widely used in natural language processing, text
processing, and other fields. In recent years, it has also attracted the attention of experts and scholars in the field of fault diagnosis [1–6].

Starting from the original monitoring data of electromechanical sensor equipment, this paper studies the fault diagnosis method of electromechanical sensor equipment based on deep learning. First, build a deep neural network prediction framework based on understanding the principles of deep learning; then start from the electromechanical sensor monitoring information; study equipment fault diagnosis methods for electromechanical sensor monitoring information; and finally, on this basis, further study more complex multisensors. The monitoring information fault diagnosis lays the foundation for the subsequent research on the remaining life prediction of equipment based on deep learning.

2. Related Work

With the development of equipment in the direction of complexity, intelligence, and integration, its working environment and operating conditions have become more and more complex, and the data collected by the monitoring system have become larger and larger. Based on the current large amount of electromechanical data, the fault diagnosis of electromechanical sensor equipment has become the research focus of machine learning. It has become a hot issue in the current PHM field. Drawing on the successful experience of deep learning, some scholars have already applied deep learning in the field of fault diagnosis and have achieved certain results. Since Tamilselvan first applied DBN to aircraft engine fault diagnosis in 2013, more and more scholars at home and abroad have paid attention to this field and achieved many research results. Tran et al. merged the DBN and TKEO algorithms and applied them to the valve fault diagnosis of reciprocating compressors. The author used GRBM instead of RBM to get rid of the shortcomings of traditional RBM that can only input binary fault signals and use the DBN model as a classifier for electromechanical sensors and a traditional Residual Neural Network, a higher accuracy rate has been achieved. Jia et al. applied deep neural networks to the feature extraction and diagnosis of bearing faults. They used the feature extraction capabilities of deep neural networks to extract and identify bearing fault features directly from frequency domain signals and achieved good results. Lu et al. specially studied the fault feature extraction ability of deep neural networks in the frequency domain signals of rotating bearings and gave the visualization results of the fault features. By summarizing the above-mentioned literature, although deep learning has been successfully applied in the field of failure, the research in this field is still immature, and there are still many problems to be solved. The main problems are summarized as follows: (1) only use deep learning as a classifier, traditional signal processing methods are still used to extract fault features, and the ability of deep learning to mine fault features is not fully utilized; (2) when using deep learning algorithms to extract fault features and identify the health status of fault signals, most of the signal frequencies are used, and when the signal does not have periodicity, this method will fail and is not general; (3) the comparative study of deep learning and other shallow models is not sufficient; and (4) the role of deep learning in the fault diagnosis of electromechanical sensor monitoring information application remains to be studied [7–15].

3. Related Theoretical Methods

3.1. Data-Driven Remaining Life Prediction Framework. A relatively complete data-driven RUL prediction framework suitable for electromechanical sensor equipment systems often covers two stages of online application and offline modeling. The online and offline parts complement and support each other. In the offline phase, make full use of equipment system test data, historical monitoring data, physical and mathematical models of subsystems or components, and so on to establish operating condition recognition, data preprocessing, feature engineering, degradation status recognition, and RUL for online practical applications. In the online stage, after necessary data preprocessing, feature engineering and other operations are performed on the data collected in real-time monitoring, based on the actual operating environment of the service equipment system (such as external environment, load conditions, etc.), with the help of intelligent algorithmic prediction system RUL. In this framework, training and learning in the offline phase and actual prediction in the online phase are fully combined, and the model can be adjusted in an offline or online dynamic manner [16–20], as shown in Figure 1.

Data-driven RUL prediction methods are currently mainly based on machine learning, mathematical statistics, and deep learning. Different prediction methods have different adaptabilities. Corresponding to condition monitoring data, there are usually two RUL prediction strategies. One is to map $m$-dimensional features to a single-dimensional HI value curve and use curve fitting, extrapolation, and other methods to predict, which is called RUL indirect prediction strategy. This article uses direct pattern matching on the original multidimensional data and extracts the $m$-dimensional characterization of degraded fault features to predict the remaining life of the equipment system, which is an RUL direct prediction strategy.

3.2. Deep Neural Network. A deep neural network can be regarded as a multihidden neural network formed by stacking multiple AEs. It uses bottom-up unsupervised learning to extract features layer by layer and uses supervised learning methods to fine-tune the parameters of the entire network. The most essential characteristics of a certain state of the object are extracted from the original data, and the deep neural network structure is shown in Figure 2 [21].
3.2.1. Autoencoder. The autoencoder is a three-layer forward artificial neural network, as shown in Figure 3. Its output is equal to the input, and it is mainly composed of an input layer, a hidden layer, and an output layer. The autoencoder is mainly composed of two parts: encoder and decoder. The input data are transformed into features through the encoder network, and the input data are encoded from high- to low-dimensional space data, and the input data are obtained in low-dimensional space. Then, the low-dimensional space data are mapped to the high-dimensional space through the decoder network to realize the reconstruction of the input to the output, and the obtained low-dimensional space data can be used as the feature representation of the input high-dimensional space data [22].

Given an unlabeled data set containing $P$ observation variables and $M$ samples $\{x_{pm}\}$, $(p = 1, 2, \ldots, P; m = 1, 2, \ldots, M)$, the coding network passes The activation function $f_\theta$ encodes the sample $x_m$ into the activation value $h_m$ of the hidden layer, and the coding process is shown in the following equation:

$$h_m = f_\theta(x_m) = \sigma(Wx_m + b).$$  \hspace{1cm} (1)\)

$f_\theta$ is the coding function, the function $s$ usually takes the Sigmoid function as the activation function of the coding network, $W$ is the network weight matrix between the input layer and the hidden layer, $b$ is the bias vector generated by the coding network, $\theta = \{W, b\}$ is the connection weight and bias parameter between the input layer and the hidden layer. The general form of the Sigmoid function is shown in the following formula:
Similarly, for the decoding network, the encoding vector $h$ obtained by the encoding network is reconstructed by the decoding network to obtain $m_y$ that is equal to the input, that is, $m_y$ and the input $m_x$ are equal. The decoding process is shown in the following equation:

$$y_m = g^T(h_m) = \sigma(W^Th_m + d).$$

$g^T$ is the decoding function, $\sigma$ is the activation function of the hidden layer to the output layer, and $d$ is the bias vector generated during the encoding process. The essence of the process of training AE is to train and optimize the network parameters $\theta$ and $\theta^T$. In order to make the output $y_m$ as close as possible to the input $x_m$ by minimizing the reconstruction error $J(\theta, \theta^T)(x, y, W, b)$ to characterize the closeness between the input and the output, as shown in the following equation:

$$J(\theta, \theta^T)(x, y, W, b) = \frac{1}{m}y - x^2.$$  

In each training process, the gradient descent method is used to update the network training parameters $\theta$ and $\theta^T$ of AE. The entire parameter update process is as follows:

$$W_l = W_l - \alpha \frac{\partial}{\partial W_l} J(\theta, \theta^T)(x, y, W, b), \quad l = 1, 2, 3,$$

$$b_l = b_l - \alpha \frac{\partial}{\partial b_l} J(\theta, \theta^T)(x, y, W, b), \quad l = 1, 2, 3,$$

where $\alpha$ is the learning rate and $\frac{\partial}{\partial W_l} J(\theta, \theta^T)(x, y, W, b)$ and $\frac{\partial}{\partial b_l} J(\theta, \theta^T)(x, y, W, b)$ are calculated using backpropagation algorithm.

### 3.2.2. Deep Neural Network Training

First, the DNN network is pretrained for the failure prediction of electromechanical sensor equipment through the training algorithm. Train the first autoencoder $AE_1$ by giving an unlabeled input data set $x$ as the input of the coding network and get the coding vector $h_1$. Set $x$ as the output of $AE_1$ to obtain the training parameter $\theta_1$; and then use the hidden layer feature $h_1$ of the first autoencoder as the input of the second autoencoder $AE_2$; and train to obtain the network parameter $\theta_2$ of $AE_2$ and $h_2$ as the hidden layer of $AE_2$. The data can be viewed as a feature representation of $AE_2$. Repeat this process to obtain the hidden layer feature $h_N$ of the $N$-th autoencoder $AE_N$ and the corresponding network training parameter $\theta_N$ [23, 24].

Second, add a classifier to the top of the DNN network. The pretraining process of the DNN is completed through the layer-by-layer unsupervised training method, which realizes the layer-by-layer extraction of feature information. However, the DNN at this time does not have a classification function. In order to achieve the output classification function, a classifier needs to be added to the top layer of the DNN. This paper uses the softmax classifier as the output layer of DNN. Assuming that the training data set is $\{x_m\}(m=1, 2, \ldots, M)$, the label is $u_m \in \{1, 2, \ldots, k\}$; each probability $p(\alpha = i|x)$ of a type $i(i=1, 2, \ldots, k)$ can be calculated by the following hypothesis function:

$$\begin{pmatrix}
pu_1 = 1|x_m; \theta \\
pu_2 = 2|x_m; \theta \\
\vdots \\
pu_k = k|x_m; \theta
\end{pmatrix} = \frac{1}{\sum_{i=1}^{k} e^{\theta_i^T X_m}} \begin{pmatrix}
e^{\theta_1^T X_m} \\
e^{\theta_2^T X_m} \\
\vdots \\
e^{\theta_k^T X_m}
\end{pmatrix}. \quad (6)$$

$\theta$ is the model parameter of softmax, which is similar to the AE model. In order to ensure the performance of the classifier, the training parameters of the model are optimized by minimizing the loss function $J_p$. The top-level network parameter $\theta_{N+1}$ can be obtained by minimizing $J_p x_m$ as follows:

$$J_p x_m = -\frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{M} [u_m = i] \log \frac{e^{\theta_i^T X_m}}{\sum_{i=1}^{k} e^{\theta_i^T X_m}}. \quad (7)$$

Finally, fine-tune the training parameters of the entire DNN. In order to ensure the accuracy of feature extraction and the classification effect of the output layer, the backpropagation algorithm is used to supervise the fine-tuning of the entire DNN training parameters through a limited number of sample labels, and the fine-tuning process is completed by minimizing the reconstruction error $E(\theta)$. The parameter update process is as follows:

$$E(\theta) = \frac{1}{M} \sum J_\theta(Y_m, u_m; \theta),$$

$$\theta = \theta - \alpha \frac{\partial E(\theta)}{\partial \theta}. \quad (8)$$
\[ Y_m \text{ is the true output value, } \theta \text{ is the parameter set obtained from the entire network training process } \theta = \{\theta_1, \theta_2, \ldots, \theta_N, \theta_{N+1}\}, \text{ the back-propagation algorithm is mainly used to continuously modify the network parameters } \theta, \text{ and } \alpha \text{ is the learning rate. The fine-tuning process uses labeled data to improve the accuracy of DNN training.} \]

4. Based on the Improved Residual Neural Network Electromechanical Sensor Equipment Fault Diagnosis Model System Design

4.1. Random Weight Average Method SWA

4.1.1. Learning Rate. During SGD training, there are two strategies for setting the learning rate: fixed learning rate and periodic learning rate. Under a certain learning rate strategy of SWA, the average of the weight data in the SDP training process is used as the weight algorithm. When the environment of the learning rate changes, assuming that the learning rate decreases linearly from \( \alpha_1 \) to \( \alpha_2 \) in a cycle, the change formula of the learning rate is

\[
\alpha(i) = 1 - t(i)\alpha_1 + t(i)\alpha_2 \alpha = \frac{1}{c} (\text{mod}(i - 1, c) + 1). \tag{9}
\]

4.1.2. Learning Rate Error Curve. The performance of the learning rate and error rate during the training of the SGD algorithm is shown in Figure 4.

In Figures 1–4, the picture shows the change curve of the learning rate, and the picture below shows the training error corresponding to the learning rate. The small circle on the learning rate curve indicates the minimum learning rate and its corresponding test error. Figure 4 shows the change in the learning rate and the corresponding error rate during a batch of training. It can be found that the learning rate does not change continuously. At the end of a cycle, the learning rate jumps directly from the minimum to the maximum. Basically, the learning rate \( \alpha_1 \geq \alpha_2 \), and the length \( c \) of a cycle is a hyperparameter. In order to improve training, it is also necessary to consider the case where the learning rate remains unchanged.

4.2. Dropout Regularization. If the training sample is insufficient, it is easy to produce overfitting phenomenon. The phenomenon of overfitting will reduce the effect of actual prediction. The failure prediction model of electromechanical sensor equipment has a smaller loss on the training set and a higher accuracy rate, while the opposite is true on the test machine. Dropout not only can effectively reduce the overfitting phenomenon but also can reduce the fault prediction error rate of electromechanical sensor equipment. Its essence is that during training, the neuron becomes 0 with probability \( p \) and does not participate in the process of forward propagation. This brings about a reduction in the dependence of the electromechanical sensor equipment failure prediction model on local features, so the electromechanical sensor equipment failure prediction model has better generalization capabilities. The structure of dropout is shown in Figure 5.

4.2.1. Workflow of Dropout

(1) Set the node hidden probability \( 0 < p < 1 \), and the neurons in the hidden layer will be shielded with probability \( p \)

(2) The neurons in the hidden layer are shielded with probability \( p \), and the shielded neurons no longer participate in the training process

(3) Neurons that have not been deleted proceed forward and backward in the normal way and update normally \((w, b)\)

(4) Restore the shielded neurons and then perform steps 1 and 2
From the above analysis, we can know that the idea of dropout is that certain neurons do not participate in training at certain times.

4.2.2. How to Implement Dropout. During the training phase, each neuron needs to be shut down probabilistically. The model diagram is shown in Figure 6. The corresponding calculation process is as follows.

When there is no dropout layer, the calculation becomes

\[ z_i^{(l+1)} = w_i^{(l+1)} y_i^l + b_i^{(l+1)} . \] (10)

The corresponding network calculation formula without dropout is as follows:

\[ r_i^{(l)} \sim \text{Bernoulli}(p), \] (11)
\[ y_i^l = r_i^l \times y_i^l, \] (12)
\[ z_i^{(l+1)} = w_i^{(l+1)} y_i^l + b_i^{(l+1)}, \] (13)
\[ y_i^{(l+1)} = f(z_i^{(l+1)}). \] (14)

Equations (12)–(14) use Bernoulli functions. The function of the Bernoulli function is to generate a random vector composed of 0 and 1 according to the probability. In addition, \( f \) represents the activation function; \( y \) represents the output; \( z \) represents the value of the neuron after the summation, which becomes the output after the activation function.

4.2.3. Effectiveness Analysis of Dropout. For model optimization, there is a classic method. For a specific electromechanical sensor device to predict the scene, train multiple models at the same time, as shown in Figure 7. This is a three-category model that predicts three categories of \( a/b/c \), among which

\[ a = \frac{a_1 + a_2}{2}, \] (15)
\[ b = \frac{b_1 + b_2}{2}, \]
\[ c = \frac{c_1 + c_2}{2}. \] (16)

Figure 7 shows a structure in which two models are averaged. Through the superposition of multiple networks, the generalization ability of the predictive model of electromechanical sensor equipment can be improved. And dropout achieves the effect of multiple networks superimposed to a certain extent. Because after regularization, the network will randomly lose some neurons each time, the model structure after discarding is different from that before discarding, which is equivalent to a new network. In this way, there will be many different networks, and then the final prediction results will be obtained by averaging or voting on these models. Dropout replaces the multimodel solution to a certain extent. From the above analysis, we know that some neurons will be randomly closed every time forward and backward. In essence, the neurons can be considered as a new network after they are closed. In this way, multiple networks are generated during the training process, and the final prediction result is the average effect of multiple networks. After taking the average, the risk of overfitting of the model is effectively suppressed. With the introduction of dropout, two neurons may not appear in the network at the same time. In this way, the weight update does not depend on the joint action of all hidden nodes. In this way, the problem of excessive reliance on special features in the training process is prevented. Make the neural network train some more common features for classification. Increase the robustness of the network. It can also be said that the introduction of dropout reduces the sensitivity of special features during the training process so that the generalization ability of the system is significantly improved. The dropout layer prevents some nodes from participating in training so that the actual training time of the node is less than the set algebra, which also reduces the risk of overfitting in the prediction process of electromechanical sensor equipment.

5. Fault Prediction and Analysis of Electromechanical Sensors Based on Improved Residual Neural Network

5.1. Operation of Electromechanical Sensor Fault Prediction System Operation. For the residual neural network, building the residual block is the most fundamental task of building a residual neural network. A good residual block design is very important for neural networks. A typical residual block contains two convolutional layers, and the two convolutional layers are not directly connected. In order to improve the recognition effect of the algorithm, we introduced the BN layer and the activation function, and the combination between them constitutes a complete residual layer. There are multiple combinations of residual blocks, and the residual structure selected in this paper is shown in Figure 8 [25, 26].
This paper analyzes the predictions of electromechanical sensor equipment for improved networks, optimized by swa algorithm, dropout regularization, and both algorithms. After the dropout optimization of the ResNet50 network structure, its network structure becomes Figure 9. In ResNet50, a total of five dropout structures are added.

5.2. Result Analysis. This paper randomly selects 30 fault features; the training algebra is set to 30 generations; and the trained electromechanical sensor equipment prediction model is ResNet50 and its improved model. After comparison of experiments, it is found that the four kinds of electromechanical sensor equipment prediction models trained have overfitting, but the difference is that their overfitting degrees are inconsistent. The following are the experimental results: the training results of the original ResNet50 are shown in Figure 10. The training results are as follows: The training accuracy rate is 99.84%, and the loss is 0.03. The training accuracy rate is 91.69%, and the loss is 0.49.

After adding dropout regularization, the training effect is shown in Figure 11. The training results are as follows: The training accuracy rate is 99.93%, and the loss is 0.04. The training accuracy rate is 92.75%, and the loss is 0.67.

After improving the model through the swa algorithm, the training effect diagram is shown in Figure 12. The training results are as follows: The training accuracy rate is 99.82%; loss is 0.02.

Through the above data verification, the following conclusions can be drawn:

1. The four electromechanical sensor equipment prediction models all have an overfitting phenomenon, that is, the test error of the test set is significantly lower than the training error of the training set.
2. To the network that joins the dropout, the training process fluctuates greatly, which is in line with the expected result. Because when the network turns off certain weights, it is equivalent to a new network, and training on this basis may fluctuate.

![Figure 8: Structure realization of the residual block.](image)

![Figure 9: ResNet50 network structure after adding dropout.](image)
Figure 10: ResNet training effect, left loss, and right Acc before the improvement.

Figure 11: ResNet training effect, left loss, and right Acc after adding dropout regularization.

Figure 12: swa improved ResNet50 training results.

Figure 13: Improved swa and dropout, ResNet50 training results.
(3) Networks that use swa algorithm improvement and simultaneous swa algorithm and dropout regularization algorithm improvement have the best test results, but the latter is slightly better. This shows that the random weight averaging algorithm and dropout have the effect of suppressing overfitting.

6. Conclusion
This article takes electromechanical sensor equipment as the research object. Aiming at the situation of strong human intervention and insufficient recognition ability in the fault diagnosis of electromechanical sensor equipment, the study of deep learning in the fault diagnosis of electromechanical sensor equipment is carried out. The work of this article is summarized as follows:

The advanced convolutional neural network model is used to maximize the use of the fault recognition ability of the deep learning network for fault diagnosis, reducing manual intervention models.

Aiming at the phenomenon that the electromechanical sensor equipment predicts that the training process is prone to overfitting, two improved methods are adopted. Based on the random weight averaging algorithm and dropout regularization, the model is improved from two aspects: backpropagation and model structure. Experiments show that when the backpropagation learning rate is fixed or periodically changes, the SWA algorithm can effectively improve the training effect. Dropout regularization is to randomly discard some neurons so that the network will not rely too much on certain local features, thereby improving the generalization ability of the model to suppress overfitting, and the effect of electromechanical sensor equipment failure prediction is better.

**Data Availability**
The data set can be accessed upon request to the corresponding author.

**Conflicts of Interest**
The authors declare that they have no conflicts of interest.

**References**

[1] A. El Mejdoubi, H. Chaoui, H. Gualous, P. Van Den Bossche, N. Omar, and J. Van Mierlo, “Lithium-ion batteries health prognosis considering aging conditions,” *IEEE Transactions on Power Electronics*, vol. 34, no. 7, pp. 6834–6844, 2019.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018.

[3] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabouj, “Real-time motor fault detection by 1-D convolutional neural networks,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[4] O. Janssens, V. Slavkovikj, B. Vervisch et al., “Convolutional neural network based fault detection for rotating machinery,” *Journal of Sound and Vibration*, 2016.

[5] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning,” *Nature Communications*, vol. 5, Article ID 4308, 2014.

[6] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, “Averaging weights leads to wider optima and better generalization,” vol. 1803, pp. 1–12, 2018, https://arxiv.org/abs/1803.05407, Article ID 5407.

[7] C. Li, R.-V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera, “Fault diagnosis for rotating machinery using vibration measurement deep statistical feature learning,” *Sensors*, vol. 16, no. 6, p. 895, 2016.

[8] S. Shili, A. Hijazi, A. Sari, X. Lin-Shi, and P. Venet, “Balancing circuit new control for supercapacitor storage system lifetime maximization,” *IEEE Transactions on Power Electronics*, vol. 32, no. 6, pp. 4939–4948, 2017.

[9] S. Subramanian, M. A. Johny, M. Malamal Neelanchery, and S. Ansari, “Self-Discharge and voltage recovery in graphene supercapacitors,” *IEEE Transactions on Power Electronics*, vol. 33, no. 12, pp. 10410–10418, 2018.

[10] A. El Mejdoubi, H. Chaoui, H. Gualous, and J. Sabor, “Online parameter identification for supercapacitor state-of-health diagnosis for vehicular applications,” *IEEE Transactions on Power Electronics*, vol. 32, no. 12, pp. 9355–9363, 2017.

[11] F. Naseri, E. Farjah, M. Allahbakhshi, and Z. Kazemi, “Online condition monitoring and fault detection of large supercapacitor banks in electric vehicle applications,” *IET Electrical Systems in Transportation*, vol. 7, no. 4, pp. 318–326, 2017.

[12] H. Yang, “Estimation of Supercapacitor charge capacity bounds considering charge redistribution,” *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 6980–6993, 2018.

[13] W. Houlian and Z. Gongbo, “State of charge prediction of supercapacitor via combination of kalman filtering and backpropagation neural network,” *IET Electric Power Applications*, vol. 12, no. 4, pp. 588–594, 2018.

[14] B. Dolenc, P. Boškoski, M. Štepančič, A. Pohjoranta, and D. Jurčič, “State of health estimation and remaining useful life prediction of solid oxide fuel cell stack,” *Energy Conversion and Management*, vol. 148, pp. 993–1002, 2017.

[15] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, “Prognostics of proton exchange membrane fuel cell stack in a particle filtering framework including characterization disturbances and voltage recovery,” in *Proceedings of the IEEE Conference on Prognostics and Health Management*, pp. 1–6, Cheney, WA, USA, June 2014.

[16] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, “Joint particle filters prognostics for proton exchange membrane fuel cell power prediction at constant current solicitation,” *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 336–349, 2016.

[17] M. Ahwiadi and W. Wang, “An enhanced mutated particle filter technique for system state estimation and battery life prediction,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 3, pp. 923–935, 2019.

[18] F. Cadini, C. Sarubafati, F. Cancelleri, and M. Giglio, “State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters,” *Applied Energy*, vol. 235, pp. 661–672, 2019.

[19] D. André, C. Appel, T. Soczka-Guth, and D. U. Sauer, “Advanced mathematical methods of SOC and SOH estimation for lithium-ion batteries,” *Journal of Power Sources*, vol. 224, pp. 20–27, 2013.

[20] K. Cho, B. V. Merrienboer, C. Gulcehre et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *Computer Science*, 2014.
[21] J. Lago, F. De Ridder, and B. De Schutter, “Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms,” *Applied Energy*, vol. 221, pp. 386–405, 2018.

[22] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, “Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2018.

[23] S. K. Ibrahim, A. Ahmed, M. A. E. Zeidan, and I. E. Ziedan, “Machine learning methods for spacecraft telemetry mining,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, pp. 1816–1827, 2019.

[24] T. Ergen and S. S. Kozat, “Efficient online learning algorithms based on LSTM neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3772–3783, 2018.

[25] M. Elhoseny, A. Farouk, and A. E. Hassanien, “K-coverage model based on genetic algorithm to extend WSN lifetime,” *IEEE sensors letters*, vol. 1, no. 4, pp. 1–4, 2017.

[26] L. Fabbri and M. H. Wood, “Accident damage analysis module (ADAM): novel European commission tool for consequence assessment-scientific evaluation of performance,” *Process Safety and Environmental Protection*, vol. 129, pp. 249–263, 2019.