eTransportation 16 (2023) 100230

Contents lists available at ScienceDirect

eTransportation

journal homepage: www.journals.elsevier.com/etransportation

Uncertainty quantification of proton-exchange-membrane fuel cells degradation prediction based on Bayesian-Gated Recurrent Unit

Wenchao Zhu [a,b,d], Bingxin Guo [a], Yang Li [a], Yang Yang [a,b,**], Changjun Xie [a,b,c,*], Jiashu Jin [a], Hoay Beng Gooi [d]

[a] School of Automation, Wuhan University of Technology, Wuhan, 430070, China
[b] Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan, 430070, China
[c] Modern Industry College of Artificial Intelligence and New Energy Vehicles, Wuhan University of Technology, Wuhan, 430070, China
[d] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Keywords:
Uncertainty quantification
Variational inference
Proton exchange membrane fuel cell
Bayesian-Gated recurrent unit

ABSTRACT

Machine learning is very important in predicting the degraded performance of fuel cell systems for advanced diagnosis and control. Unfortunately, existing machine-learning-based schemes are usually designed with point estimation, making it difficult to quantify the uncertainty of the prediction result. In this paper, we propose a Bayesian-Gated Recurrent Unit model (B-GRU) that combines the Bayesian Theory and GRU to predict the phenomenon of fuel cell voltage decay. Fuel cell data are preprocessed by the random forest, and the key feature data are then imported into the B-GRU. Variational inference and adaptive moment estimation is used to obtain the optimal parameters in the B-GRU. Probability density distributions are calculated by replacing the parameters in GRU with random variables to quantify the uncertainty in the model. In addition to providing point estimates, the B-GRU also gives interval estimates for uncertainty quantification. With small training data, the point estimation result of B-GRU is more accurate than traditional neural networks. Furthermore, compared to the Bayesian neural networks, the proposed B-GRU also exhibits superior performance both in point and interval estimation results based on the IEEE PHM 2014 DATA Challenge dataset. With its excellent ability for noise immunity and uncertainty quantification, the proposed prediction method can provide more useful decision-making recommendations for hydrogen energy devices.

1. Introduction

1.1. Background and Literature review

Due to their high power density, environmental friendliness, lightweight, and abundant resources, proton exchange membrane fuel cells (PEMFCs) have become one of the most promising power sources for a variety of transportation applications, such as hybrid vehicles and plug-in hybrid vehicles [1,2], heavy-duty trucks [3], buses [4], trains [5], and ships [6]. However, PEMFCs have cost and lifespan bottlenecks, hindering their commercialization and large-scale applications [7,8]. Accurately predicting the performance degradation of PEMFCs is thus important for estimating the remaining useful life (RUL) and making relevant optimal operational decisions.

The methods of predicting the performance of PEMFCs consist of data-driven and model-based ones. In recent years, data-driven methods have emerged as a promising tool for predicting fuel cell performance due to the rapid development of machine learning techniques and the difficulty in modeling the degradation from the physical aspects. Compared to other machine learning algorithms, deep learning has excellent scalability and generalization ability in processing large and complex data [9,10]. Therefore, PEMFC health or performance degradation prediction based on deep learning has received increasing attention [11–13]. Traditionally, the health prediction framework based on deep learning includes four main steps: data collection, index construction, health stage division, and RUL prediction. For modern applications in electrified transportation, the operating load conditions of PEMFC are very complex [14]. Zuo et al. thus proposed an attention-based recurrent neural network (RNN) model to more accurately predict the output voltage degradation of a PEMFC based on

* Corresponding author. School of Automation, Wuhan University of Technology, Wuhan, 430070, China.
** Corresponding author. School of Automation, Wuhan University of Technology, Wuhan, 430070, China.
E-mail addresses: whutyangyang@whut.edu.cn (Y. Yang), jackxie@whut.edu.cn (C. Xie).

https://doi.org/10.1016/j.etran.2023.100230
Received 13 July 2022; Received in revised form 17 January 2023; Accepted 25 January 2023
Available online 31 January 2023

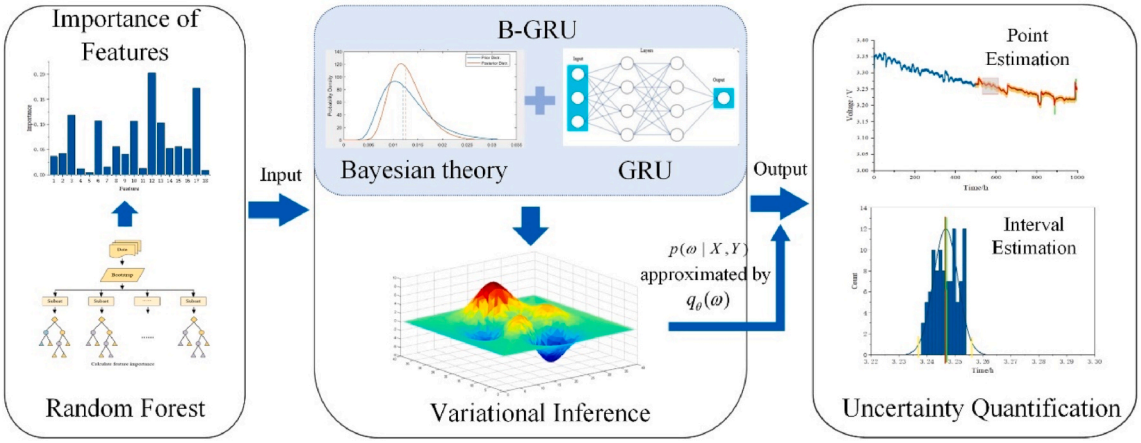

Fig. 1. Proposed fuel cell performance prediction framework.

long-term dynamic load cycle durability test data [15]. Using ensemble echo state networks (ESN) in the time-varying model space, Li et al. [16] developed an adaptive prognostic approach for PEMFC, where a prediction model was established using a model identification method based on dynamic load cycle test data. Yue [17] et al. proposed an adaptive data-driven fuel cell prediction method based on multiplicative feature decomposition and ESN to predict fuel cell degradation behavior under dynamic operating conditions. The ESN algorithm in Ref. [18] also performed RUL prediction on PEMFC under dynamic conditions and used a Linear Parameter Varying (LPV) model to simulate the dynamic process of the system. After extracting the degradation indicators through the electrochemical mechanism model, Wang [19] et al. developed a long short-term memory network (LSTM) based on dimensionality reduction symbol representation to predict the performance degradation of vehicle-oriented PEMFC. In addition, a degradation model with a sliding window was proposed to extract the health indicators. Then a symbolic representation-based LSTM was developed to predict the trend under dynamic conditions [20]. In these data-driven methods, different health indicators or degradation indicators are selected to achieve degradation prediction [15–20]. These methods achieved accurate predictions point estimations of prediction performance through deterministic neural networks. However, the absence of uncertainty quantification may cause difficulty in giving confidence in the prediction results, where the predicted results can sometimes be unreliable [21]. Furthermore, making control decisions based on single-point predictions is error-prone, which could lead to various potential safety issues.

In the area of machine learning, uncertainty can be considered either epistemic or aleatoric [21]. The epistemic uncertainty is always caused by predictive models and is also commonly referred to as model uncertainty. On the other hand, data collection methods affect the aleatoric uncertainty, which measures the noise in the observed dataset. Performance predictions can be affected by various uncertainty, such as model uncertainty, measurement uncertainty of data, and forecast uncertainty under operating conditions [22,23].

Currently, there are limited investigations on quantifying prediction uncertainty in deep learning-based performance prediction. Ghahramani et al. [21] highlighted the Bayesian approach as a promising approach. Bayesian inference is used to deal with uncertainty using Bayesian theory as the language of mathematics. Combining Bayesian theory and neural network, Wang et al. [24] developed a Bayesian Neural Network (BNN) for uncertainty quantification prediction of diesel engines. Peng et al. [25] proposed a Bayesian Deep Learning (BDL)-based method for quantifying health prediction with uncertainty and demonstrated the effectiveness of the method on ball-bearing datasets and turbine engine datasets. Cheng et al. [26] proposed a hybrid forecast way based on the least squares support vector machine (LSSVM) and regularized particle filter (RPF) to describe the uncertainty of RUL prediction through probability density distribution. Among them, the cores of particle filtering and Kalman Filtering methods are again Bayesian methods.

1.2. Research gap and contributions

For periodic sequence data such as diesel engines and ball bearings, BNNs that combines Bayesian theory and DNNs have shown their effectiveness. However, recurrent neural networks (RNNs) are superior to DNNs in feature extraction and performance prediction for more general time series data. It is highly desirable to incorporate uncertainty representation into advanced deep learning models for complex time-series data generated by fuel cells and further embed uncertainty inference into mature deep learning methods. The scalability and the generalization ability brought by the learned model are reflected in the field of fuel cell performance prediction. A more accurate life prediction of fuel cells could be made using voltage prediction results with uncertainty quantification.

This paper proposes a performance prediction method for fuel cells from the perspective of uncertainty quantification for the first time. We present a new tool named the Bayesian Gated Recurrent Unit (B-GRU), which extends the advanced deep learning model with Bayesian theory and variational inference. The novelty and advantages of the proposed methods are summarized as follows:

1) Compared to other mainstream neural network methods, B-GRU can provide more accurate prediction performance with less training data by averaging.
2) B-GRU has a strong fault tolerance for noise in data collection in terms of interval estimation. In addition, with the increase of training data, the confidence interval (CI) of interval estimation would be more concentrated and close to the true value.
3) The B-GRU is studied and compared with the current advanced interval estimation method BNN, and the superiority of B-GRU is verified from different dimensions.

The article is organized as follows: The framework for predicting fuel cell voltage with uncertainty quantification is presented in Section 2, where B-GRU, variational inference, and Adaptive Moment estimation (ADAM) optimization are incorporated. The parameters of the B-GRU are trained and optimized in Section 3. The superiority of the new model is fully demonstrated from different dimensions in Section 4. Conclusions are given in Section 5 of the paper.

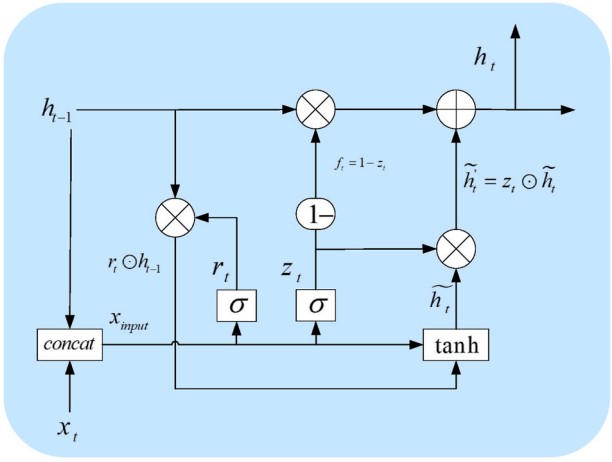

Fig. 2. GRU structure diagram.

Table 1
Calculated Gini index contribution of each data feature.

Feature	Importance	Feature	Importance
Current density	0.036242	Air inlet pressure	0.006749
Current	0.042839	Air outlet pressure	0.012470
H_2 inlet temperature	0.119115	H_2 outlet pressure	0.002881
H_2 outlet temperature	0.012176	H_2 inlet pressure	0.003073
Air inlet temperature	0.004397	H_2 inlet flow	0.002421
Air outlet temperature	0.007434	H_2 outlet flow	0.005698
Cooling water inlet temperature	0.015407	Air inlet flow	0.001926
Cooling water outlet temperature	0.005840	Air outlet flow	0.672150
Cooling water flow	0.041152	Air inlet humidity	0.008026

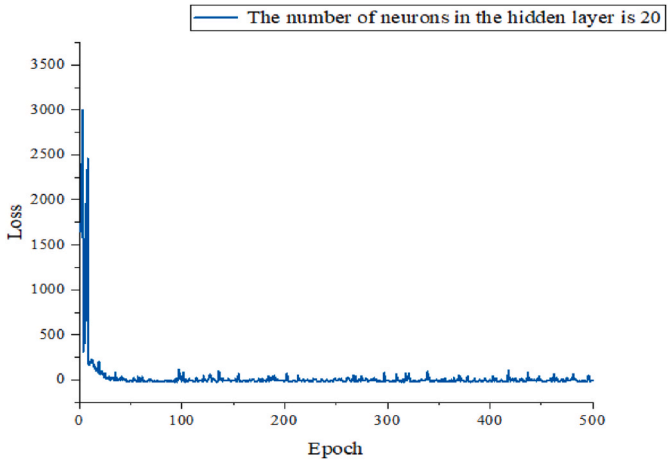

Fig. 3. Relationship between the loss and epochs with 20 neurons in the hidden layer.

2. Methodology

2.1. Prediction framework

Fig. 1 shows the proposed uncertainty quantification framework for the fuel cell stack voltage prediction. The framework mainly includes the following steps: Data preprocessing, B-GRU modeling, model training based on variational inference, and ADAM optimization.

First, the operational data of the fuel cell stack are collected, and the training dataset is preprocessed using the random forest for extracting main features. Then, the B-GRU is constructed by replacing the parameters in a classical neural network model with random variables, and the uncertainty is quantified as a probability density distribution. Thirdly, gradient-based learning and ADAM optimization are used to train a variational inference model. Finally, the result of uncertainty quantification prediction can be obtained from the training and the operational data.

In this framework, besides the interval estimation of the stack voltage prediction, the uncertainty of a point prediction can be reduced, and more instructive decision-making suggestions can be provided from the result. At the same time, the probability density distribution of the predicted results can also be obtained for practical use. The mathematical models involved in establishing this framework will be detailed below.

2.2. Random forest

The random forest is capable of processing high-dimensional data with fast training process [27]. It is thus selected as a method for preprocessing the raw data. The main steps are as follows:

1) The importance score *VIM* and the Gini coefficient *GI* of the feature are used as indicators to measure the contribution.

First, denote the feature as $X_1, X_2, \cdots X_c$, the average change of the node split impurity of the jth feature in all decision trees of the random forest denote as X_j, we have

$$GI_m = 1 - \sum_{k=1}^{|k|} p_{mk}^2 \tag{1}$$

where k represents the number of categories and p_{mk} is the fraction of the kth category in the mth node.

Next, the importance of the feature X_j in the mth node can be calculated as:

$$VIM_{jm}^{gini} = GI_m - GI_l - GI_r \tag{2}$$

where GI_l and GI_r are the Gini indices of the two nodes before and after the corresponding branch.

2) For the case that the feature X_j is contained in different nodes of the decision tree I, let m belong to the set M ($m \in M$). The importance of the first tree i can be expressed as:

$$VIM_{ij}^{gini} = \sum_{m \in M} VIM_{jm}^{gini} \tag{3}$$

3) We set the number in the random forest to 100 and the initial data feature to 18. The contribution of the feature X_j Gini indicator is:

$$VIM_j = \frac{\sum_{i=1}^{100} VIM_{ij}^{gini}}{\sum_{j=1}^{18} \sum_{i=1}^{100} VIM_{ij}^{gini}} \tag{4}$$

2.3. Gated recurrent Unit——GRU

GRU is very similar to LSTM [28]. However, compared to the three gate functions in an LSTM, there are only two gates (the update gate z_t and the reset gate r_t) in a GRU model in a GRU model, as shown in Fig. 2.

The expression of each component in the GRU structure is given as follows:

1) Input: $x_{input} = concat[h_{t-1}, x_t]$;
2) Reset gate neuron: $r_t = \sigma(x_{input} W_\tau + b_\tau)$;
3) Memory gate neuron: $\tilde{h}_t = \tanh([r_t \odot h_{t-1}, x_{input}] W_h + b_h)$;
4) Input gate neuron: $z_t = \sigma(x_{input} W_z + b_z)$;
5) Memory after input: $\tilde{h}_t' = z_t \odot \tilde{h}_t$;

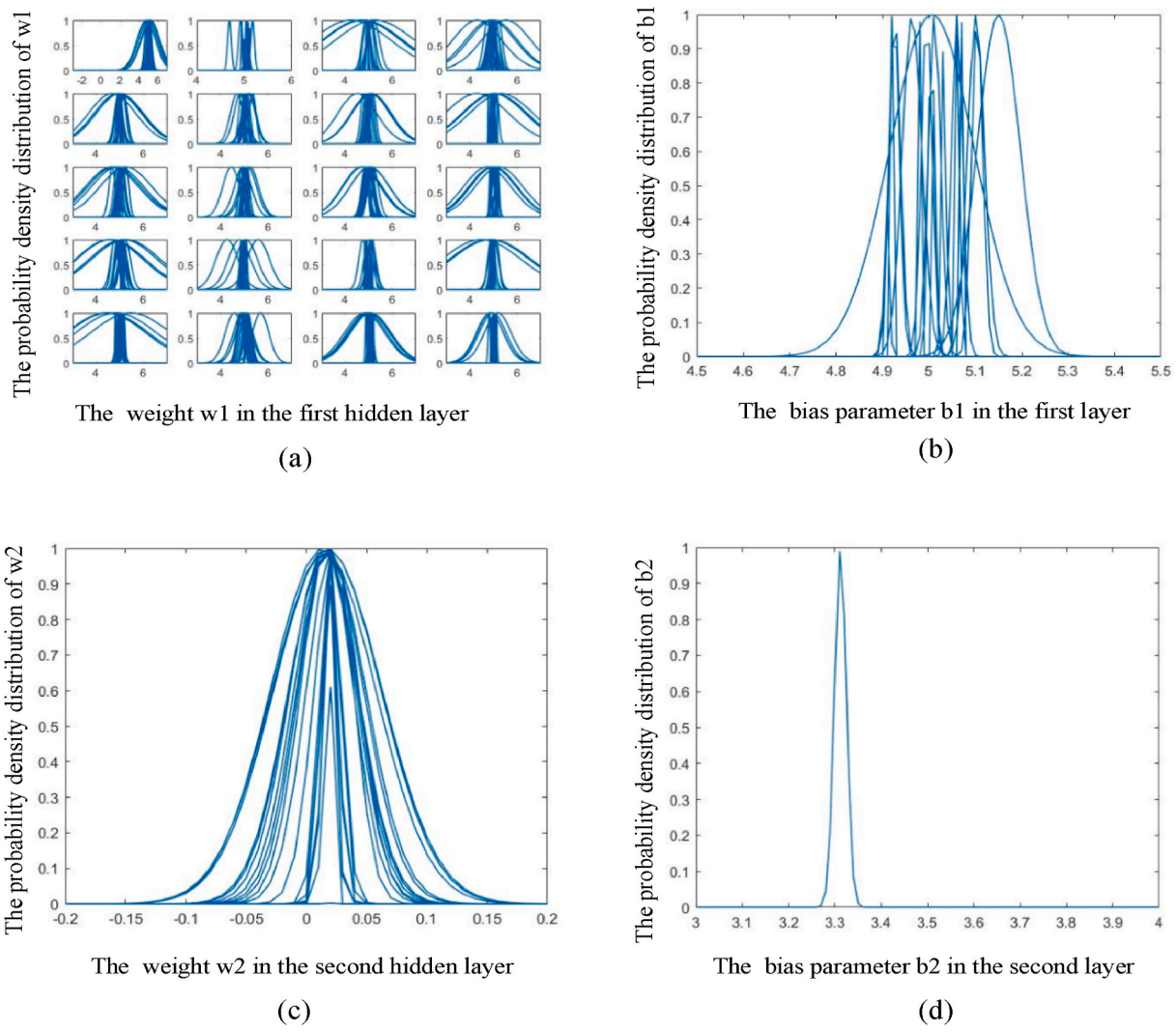

Fig. 4. Network parameter distribution.
(a) Weight distribution in the first hidden layer. (b) Bias distribution in the first hidden layer.
(c) Weight distribution in the first hidden layer. (d) Bias distribution in the second hidden layer.

Table 2
Performance of point estimation under different length training data.

Model	Training data	MSE	RMSE	MAE
B-GRU	80h	1.59E-4	1.261E-2	9.002E-2
	180h	4.9E-5	7.01E-3	4.024E-3
	280h	3.3E-5	5.744E-3	3.907E-3
	380h	**3.0E-5**	**5.477E-3**	**3.358E-3**
	480h	1.87E-4	1.368E-2	9.057E-2
	580h	2.29E-4	1.513E-2	1.087E-2
	680h	9.31E-5	9.539E-3	9.837E-3
	780h	3.15E-5	5.612E-3	3.551E-3
	880h	1.6E-5	4.006E-3	2.599E-3
	980h	1.3E-5	3.638E-3	2.247E-3

6) Forget gate neuron: $f_t = 1 - z_t$;
7) The memory at time instant $t-1$ after forgetting: $h'_{t-1} = f_t \odot h_{t-1}$;
8) The memory at the current moment t: $h_t = h'_{t-1} + \widetilde{h}'_t$.

The input x_{input} is obtained by performing the feature dimension on the memory state h_{t-1} obtained at time $t-1$ and the word vector input x_t at time t. σ refers to the sigmoid function. The output result of the reset gate neuron r_t and the input gate neuron z_t is a vector. Since both gate neurons use the sigmoid function as activation function, each element of the output vector is between 0 and 1, which is used for controlling the amount of information flowing through the valve in each dimension. The output result of the memory gate neuron $\widetilde{h}_t$ is also a vector, and it is the same as the output vector dimension of the reset gate and the input gate neuron. Since the activation function used by the memory gate neuron is the tanh function, each element of the output vector is between -1 and 1. Furthermore, W_τ, b_τ, W_h, b_h, W_z, and b_z are the parameters of each gate neuron, which shall be learned in the training process.

2.4. Modeling of B-GRU

In the B-GRU, the incorporation of the Bayesian inference with the GRU offers several advantages. On the one hand, B-GRU provides a probabilistic extension for the classic deep learning models. It retains the network topology of the classic deep learning model with high modularity and scalability. On the other hand, the constant parameters in classical DNNs are replaced with random variables. This can quantify the uncertainty through probability distribution and obtain the CI of the prediction result.

Given the training samples x and y, the B-GRU, denoted by $y = f^{\omega}(x)$, consists of a prior distribution $p(\omega)$ on the parameter space and a likelihood function of Bayesian regression $p(\mathscr{D}|\omega) = \prod_{i=1}^{N} l(y^{(i)}|f^{\omega}(x^{(i)}))$. A

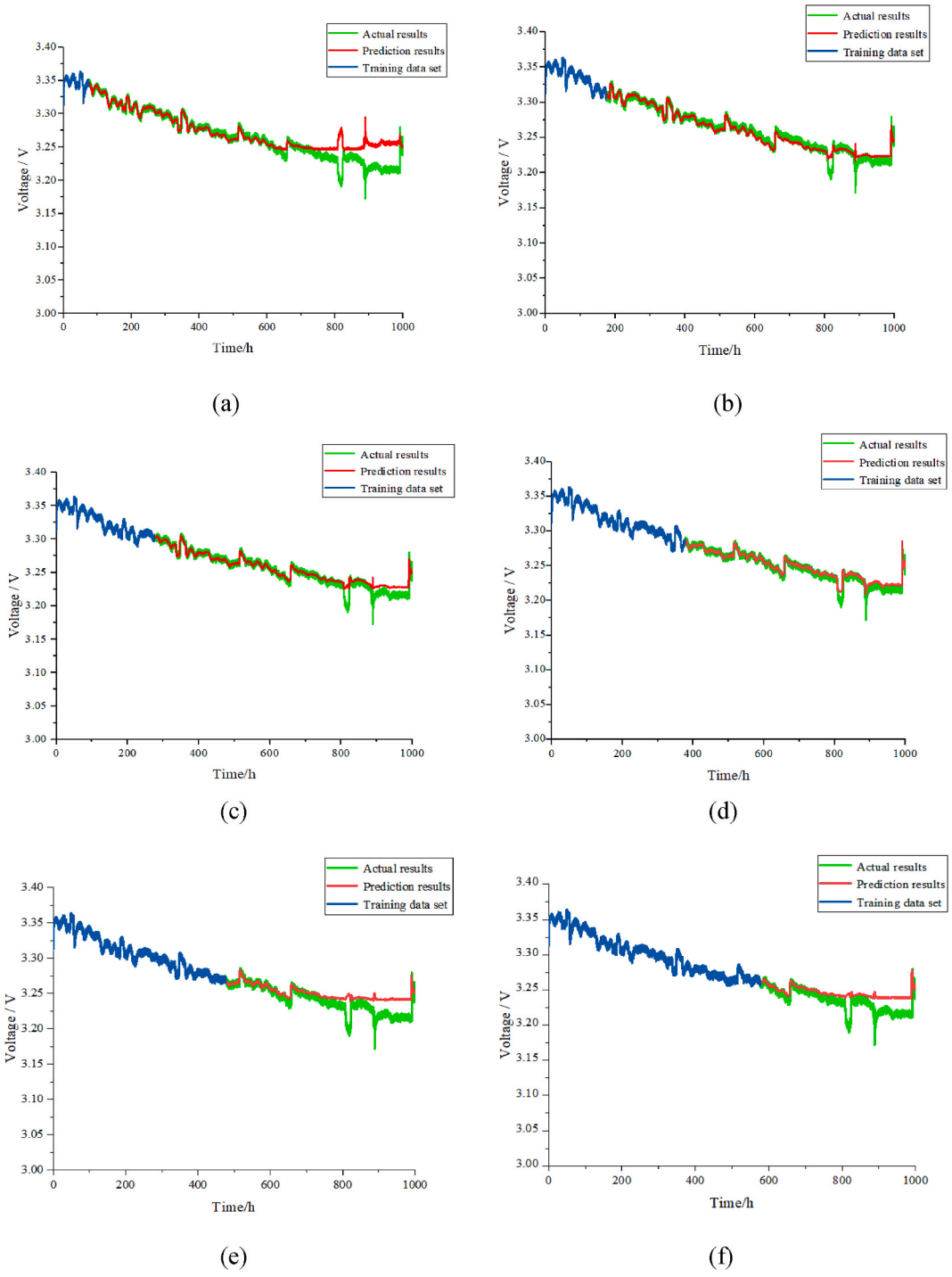

Fig. 5. Prediction results with the training data of length (a) 80 h, (b) 180 h, (c) 280 h, (d) 380 h, (e) 480 h, and (f) 580 h.

Gaussian distribution can be used for $l(y^{(i)}|f^{\omega}(x^{(i)})$ [27], and the model parameters ω are independent of the training input samples x. From Bayes' theorem, the posterior distribution of the model parameters is

$$p(\omega|D)=\frac{p(\omega)p(\mathscr{D}|\omega)}{p(\mathscr{D})}=\frac{p(\omega)p(\mathscr{D}|\omega)}{\int p(\omega)p(\mathscr{D}|\omega)d\omega}=\frac{p(\omega)\prod_{i=1}^{N}l(y^{(i)}|f^{\omega}(x^{(i)}))}{\int p(\omega)\prod_{i=1}^{N}l(y^{(i)}|f^{\omega}(x^{(i)}))d\omega} \tag{5}$$

Based on the posterior distribution $p(\omega|\mathscr{D})$, the B-GRU model $y=f^{\omega}(x)$ can be used for subsequent inference of uncertainty quantification. Specifically, given any sample data X^*, the predicted output Y^* can be obtained by

$$p(Y^*|X^*,\mathscr{D})=\lim_{\Delta\omega\to\infty}\sum p(Y^*|X^*,\omega+\Delta\omega)\times p(\omega+\Delta\omega|\mathscr{D})\times\Delta\omega=\int p(Y^*|X^*,\omega)p(\omega|\mathscr{D})d\omega \tag{6}$$

2.5. Variational inference

The main challenge of the B-GRU is that the posterior distribution $p(\omega|\mathscr{D})$ is difficult to obtain. This problem would be more complicated when the model has a complex structure with high-dimensional data. Variational inference is a method for approximating intractable distributions, effectively solving various machine learning and inference

Table 3
Prediction errors of various algorithms under different training sets.

Training data	Model	MSE	RMSE	MAE
80 h	B-GRU	1.59E-4	1.261E-2	9.002E-3
	B-LSM	2.05E-4	1.432E-2	9.247E-3
	BNN	2.23E-4	1.517E-2	9.451E-3
	RNN	1.627E-3	4.033E-2	3.347E-2
	LSTM	1.469E-3	3.832E-2	3.029E-2
	GRU	1.031E-3	3.211E-2	2.437E-2
	DNN	2.361E-3	4.859E-2	4.479E-2
180 h	B-GRU	4.9E-5	7.01E-3	4.024E-3
	B-LSTM	5.16E-5	7.183E-3	4.178E-3
	BNN	5.3E-5	7.304E-3	4.344E-3
	RNN	1.184E-3	3.442E-2	2.799E-2
	LSTM	1.044E-3	3.231E-2	2.651E-2
	GRU	6.58E-4	2.565E-2	2.159E-2
	DNN	1.352E-3	3.677E-2	3.123E-2
280 h	B-GRU	3.3E-5	5.744E-3	3.907E-3
	B-LSTM	3.42E-5	5.848E-3	3.945E-3
	BNN	3.5E-5	5.931E-3	3.987E-3
	RNN	6.83E-4	2.613E-2	2.113E-2
	LSTM	2.22E-4	1.489E-2	1.166E-2
	GRU	1.36E-4	1.166E-2	8.643E-3
	DNN	7.68E-4	2.604E-2	2.058E-2
380 h	B-GRU	**3.0E-5**	**5.477E-3**	**3.358E-3**
	B-LSTM	**3.06E-5**	**5.532E-3**	**3.373E-3**
	BNN	**3.1E-5**	**5.602E-3**	**3.419E-3**
	RNN	2.53E-4	1.592E-2	1.181E-2
	LSTM	2.2E-5	4.675E-3	2.834E-3
	GRU	2.1E-5	4.583E-3	2.758E-3
	DNN	3.95E-4	1.987E-2	1.383E-2
480 h	B-GRU	1.87E-4	1.367E-2	9.057E-2
	B-LSTM	1.90E-4	1.378E-2	9.471E-2
	BNN	1.94E-4	1.393E-2	1.018E-2
	RNN	1.23E-4	1.112E-2	8.032E-3
	LSTM	3.13E-4	1.769E-2	1.408E-2
	GRU	2.02E-4	1.421E-2	1.254E-2
	DNN	3.04E-4	1.744E-2	1.283E-2
580 h	B-GRU	2.29E-4	1.513E-2	1.087E-2
	B-LSTM	2.31E-4	1.520E-3	1.093E-2
	BNN	2.34E-4	1.531E-2	1.109E-2
	RNN	1.01E-4	1.004E-2	7.336E-3
	LSTM	4.56E-4	2.157E-2	1.659E-2
	GRU	3.84E-4	1.960E-2	1.489E-2
	DNN	4.03E-4	2.007E-2	1.563E-1

problems [29]. The core idea of the variational inference is to approximate the actual posterior distribution through a probability density distribution (variational distribution) that is easy to evaluate and infer.

By variational inference, a distribution $q_\theta(\omega|\theta)$ controlled by a set of parameters θ is used to approximate the true posterior distribution $p(\omega|\mathscr{D})$ [30]. A Gaussian distribution is used as the approximation. Let $\theta = (\mu,\sigma)$, and each network parameter ω_i obeys a Gaussian distribution with a parameter of (μ_i,σ_i). The difference between $q_\theta(\omega|\theta)$ and $p(\omega|D)$ is measured using the Kullback-Leibler (KL) divergence, i.e.,

$$\theta^* = \underset{\theta}{\operatorname{argmin}} KL[q_\theta(\omega|\theta)\|p(\omega|\mathscr{D})] \tag{7}$$

where

$$\begin{aligned} KL[q_\theta(\omega|\mathscr{D})\|p(\omega|\mathscr{D})] &= \int q_\theta(\omega|\mathscr{D})\frac{q_\theta(\omega|\mathscr{D})}{p(\omega|\mathscr{D})}d\theta \\ &= \int q_\theta(\omega|\mathscr{D})\frac{q_\theta(\omega|\mathscr{D})p(\mathscr{D})}{p(\omega,\mathscr{D})}d\theta \\ &= \int q_\theta(\omega|\mathscr{D})\log q_\theta(w|\mathscr{D})d\theta + \int q_\theta(\omega|\mathscr{D})\log p(\mathscr{D})d\theta - \int q_\theta(\omega|\mathscr{D})\log p(\omega,\mathscr{D})d\theta \\ &= \int q_\theta(\omega|\mathscr{D})\log q_\theta(\omega|\mathscr{D})d\theta + \log p(\mathscr{D}) - \int q_\theta(\omega|\mathscr{D})\log\frac{p(\omega,\mathscr{D})}{q_\theta(\omega|\mathscr{D})}d\theta - \int q_\theta(\omega|\mathscr{D})\log p(\omega,\mathscr{D})d\theta \end{aligned} \tag{8}$$

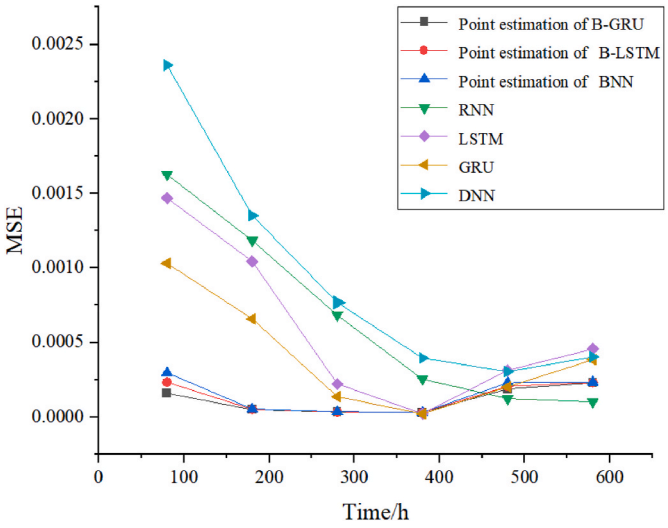

Fig. 6. Error analysis of different algorithms.

Since $KL > 0$, the following inequality holds

$$\log p(\mathscr{D}) \geq \int q_\theta(\omega|\mathscr{D})\log[p(\omega,\mathscr{D}) / q_\theta(\omega|\mathscr{D})]d\theta \tag{9}$$

where the LHS is the likelihood of the data, namely the evidence, and the RHS is called the evidence lower bound (ELBO). Assuming the evidence is constant, minimizing KL is equivalent to maximizing the ELBO. This gives

$$\theta^{opt} = \underset{\theta}{\operatorname{argmin}} KL = \underset{\theta}{\operatorname{argmax}} ELBO \tag{10}$$

where

$$\begin{aligned} ELBO &= \int q_\theta(\omega|\mathscr{D})\log\frac{p(\omega,\mathscr{D})}{q_\theta(\omega|\mathscr{D})}d\theta \\ &= \int q_\theta(\omega|\mathscr{D})\log\frac{p(\mathscr{D}|\omega)p(\omega)}{q_\theta(\omega|\mathscr{D})}d\theta \\ &= \int q_\theta(\omega|\mathscr{D})\log p(\mathscr{D}|\omega)d\theta + \int q_\theta(\omega|\mathscr{D})\log\frac{p(\omega)}{q_\theta(\omega|\mathscr{D})}d\theta \\ &= \mathbb{E}_{q_\theta(\omega|\mathscr{D})}\log p(\mathscr{D}|\omega) - KL[q_\theta(\omega|\mathscr{D})\|p(\omega)] \end{aligned} \tag{11}$$

By estimating (11) using the sampling Monte Carlo method [31], and the loss function can be found as

$$\begin{aligned} \mathscr{F}(\mathscr{D},\theta) &= KL[q_\theta(\omega|\mathscr{D})\|p(\omega)] - \mathbb{E}_{q(\omega|\mathscr{D})}\log p(\mathscr{D}|\omega) \\ &= \sum_{i=1}^{n}\log q_\theta\left(w^{(i)}|\mathscr{D}\right) - \log p\left(\omega^{(i)}\right) - \log p\left(\mathscr{D}|\omega^{(i)}\right) \end{aligned} \tag{12}$$

2.6. ADAM optimization algorithm

In order to solve for the model parameters, an optimization algorithm with fast convergence speed and high computational efficiency

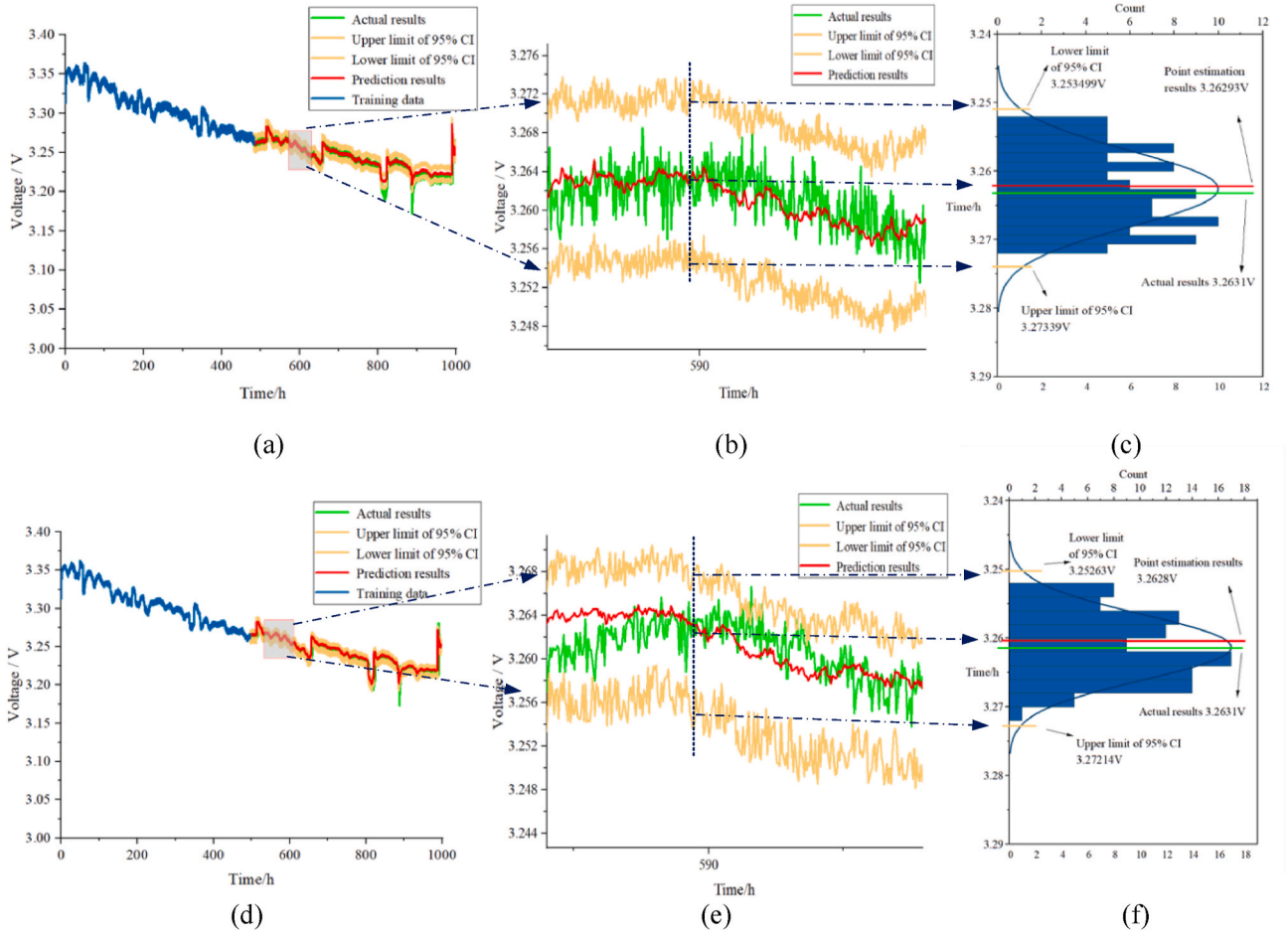

Fig. 7. Interval estimation of B-GRU under noises effect. (a) Interval estimation of raw data. (b) A zoomed-in view of the interval estimation at 590 h. (c) Probability density distribution of voltage based on original data at 590 h. (d) Interval Estimation of filtered data. (e) A zoomed-in view of the interval estimation of filtered data at 590 h. (f) Probability density distribution of voltage based on filtered data at 590 h.

needs to be developed. In this paper, the ADAM optimization algorithm is adopted, which consists of the following steps:

1) Calculate the gradient g of the objective function $f(\theta)$ with respect to parameter θ

at time t:

$$g_t = \nabla_\theta f_t(\theta_{t-1}) \tag{13}$$

2) Calculate the first moment m_t of the gradient at time t, which is a weighting

average between the first moment at the previous time $t-1$ and the current gradient g_t:

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_2)\cdot g_t \tag{14}$$

where m_t is the first moment of the gradient at time t, $m_0 = 0$.

3) Calculate the second moment of the gradient, which is the average of the pastgradient squares and the current gradient squares:

$$v_t = \beta_1 \cdot v_{t-1} + (1-\beta_2)\cdot g_t^2 \tag{15}$$

where β_1 and β_2 are the exponential decay rates of the first and the second moment estimations. We set $\beta_1 = 0.9$ and $\beta_2 = 0.999$ in this work.

4) Correct the first-order moment m_t. The initial value m_t is zero, reducing theinfluence of this bias after processing. The calculation equation is:

$$\widehat{m}_t = m_t \,/\, \left(1-\beta_1^t\right) \tag{16}$$

where $\widehat{m_t}$ is the modified first moment of the gradient at time t.

5) Correct the second-order moment v_t to reduce the influence of this bias afterprocessing:

$$\widehat{v}_t = v_t \,/\, \left(1-\beta_2^t\right) \tag{17}$$

Table 4
Interval estimation results at different times.

Time/h	Actual value	Point estimation	95%CI	Length of CI
400	3.28	3.278876	[3.2687, 3.2901]	0.0214
450	3.272	3.272865	[3.2626, 3.2845]	0.0219
500	3.2628	3.263	[3.255, 3.273]	0.018
550	3.26	3.26165	[3.2508, 3.2706]	0.0198
600	3.2555	3.25591	[3.24467, 3.26382]	0.01915
650	3.2356	3.2358	[3.22763, 3.24596]	0.01833
700	3.2485	3.24915	[3.2393, 3.2605]	0.0212
800	3.2355	3.23485	[3.22467, 3.24382]	0.01915

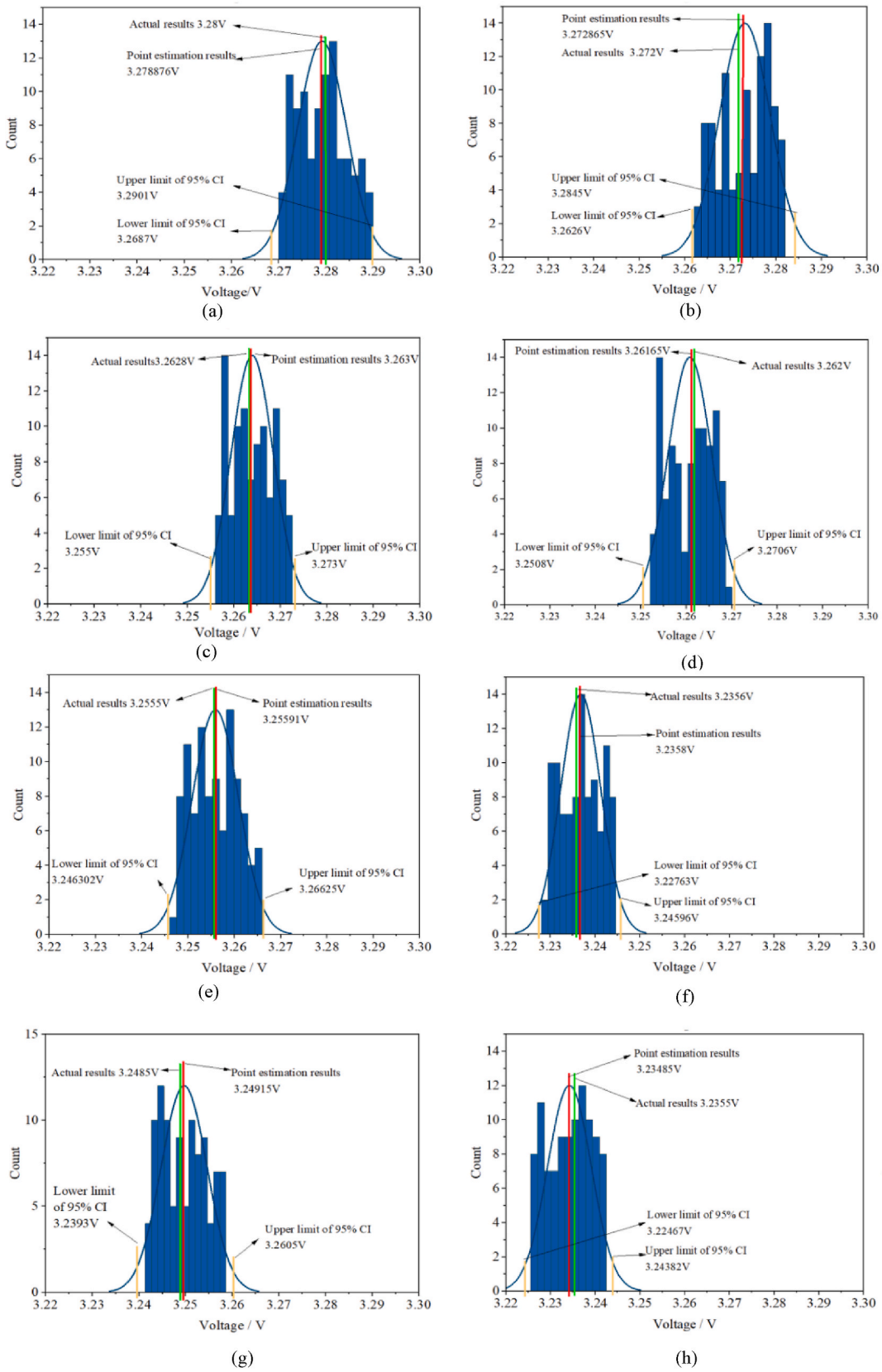

Fig. 8. Probability density distribution of the prediction results at (a) 400 h, (b) 450 h, (c) 500 h, (d) 550 h, (e) 600 h, (f) 650 h, (g) 700 h, and (h) 800 h.

where $\widehat{v_t}$ is the modified second moment of the gradient at time t.

6) Update parameters θ_t according to

$$\theta_t = \theta_{t-1} - \alpha \cdot \widehat{m_t} / (\sqrt{\widehat{v_t}} + \varepsilon) \tag{18}$$

where $\alpha = 0.1$ is the learning rate and $\varepsilon = 10^{-8}$.

3. Model training

3.1. Data preprocessing

The data used in this work were obtained from IEEE PHM 2014 data challenge.

Regeneration phenomena, material properties, and experimental conditions can significantly affect the prognostic outcomes. In data-driven methods, the raw data is rarely used directly as the input, since extracting the correct global degradation trend from the complex high-dimensional experimental data is difficult. We preprocessed the high-dimensional raw data using the random forest algorithm as described in Section 2.2. The calculated contribution of the Gini index (importance) of each data feature is shown in Table 1.

It can be seen from Table 1 that some features have negligible influence on the fuel cell voltage. We selected the features with a contribution greater than 0.01 as the input for subsequent neural network training. As a short-term prediction method, the B-GRU model would use the data of the first 60 h of these 8 variables to predict the results of the 61st h. It then uses the 8 variables of the 61st h to predict the voltage of the same hour.

3.2. Network hyperparameter selection and visualization

The hyperparameters of the B-GRU, such as the numbers of hidden layers and neurons, significantly impact the prediction performance. The mean absolute error (MAE), root-mean-square error (RMSE), and mean-square error (MSE) are used to quantify the impact of different hyperparameter settings. They are defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i|} \tag{20}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{21}$$

where y_i and $\widehat{y_i}$ are the true and the predicted fuel cell voltages, and n is the number of test samples.

It is found that the models with two hidden layers can provide the best performance, where the number of selected features directly determines the number of neurons in the first layer. Therefore, the second-layer neurons should be given more attention. The number of hidden layers and the number of neurons in the second layer are detailed in **Appendix A**. The loss function in the B-GRU is calculated by changing the number of neurons in the hidden layer while keeping other parameters the same. For a model with 20 neurons in the hidden layer, the loss over the training epochs is visualized in Fig. 3, where it can be seen that the loss function can approach zero. The corresponding parameter distributions of the B-GRU are shown in Fig. 4. It can be seen that the obtained network parameters follow a Gaussian distribution, indicating that each prediction corresponds to a specific set of network parameters.

4. Results and discussion

This section will first evaluate the prediction performance of the B-GRU from three aspects: point estimation of B-GRU, interval estimation of B-GRU, and the comparative analysis of B-GRU. Then, we will analyze the results of the performance using both static and dynamic data. Specifically, the data were mainly analyzed under static conditions, and the performance is analyzed under dynamic conditions in **Appendix B**.

4.1. Point estimation of B-GRU

Not only can the B-GRU quantify the uncertainty and provide interval estimates for the final prediction results, but it also provides point estimation of the prediction results by averaging. Two aspects of the research results are described to offer a more comprehensive analysis of point estimation performance. In the first step, the training sets with different data lengths are used to evaluate the performance of the B-GRU point estimation. Next, the B-GRU point estimation is compared with other data-driven approaches, including Deep Neural Network (DNN), RNN, LSTM, GRU, and BNN (point estimation) under the same length of the training set. The parameters of neural networks for comparison are given in Appendix C. Note that the B-GRU output would differ slightly due to uncertainty in network parameters. The point estimation is derived from the average of 100 predictions. The MSE, MAE, and RMSE are used as evaluation indicators.

4.1.1. Analysis under different length data sets

The IEEE PHM 2014 Data Challenge provides about 1000 h of stack operating data. The data in the first 80, 180, 280, 380, 480, 580, 680, 780, 880, and 980 h were used to train the B-GRU model and predict the subsequent results. Example prediction results are shown in Table 2, and the calculated evaluation indicators are shown in Fig. 5.

As shown in Table 2, the three prediction errors show a common trend: it first decreases, increases, and finally decreases. Initially, the increase of the data set can improve the accuracy of the prediction. When the training data set increases to a certain extent, the prediction model would appear overfitting. As the training dataset further increases, the prediction error would decrease again. As most of the data are used in training, the B-GRU model becomes familiar with the data set, reducing the error in the training process and thereby improving prediction accuracy.

Although the error would continue to decrease after 780 h, using smaller training data to obtain lower training errors is a cost-effective choice. Therefore, the research in the following sections is mainly carried out around the training dataset for around 380 h.

4.1.2. Comparison with other algorithms

We compare the B-GRU point estimation with RNN, LSTM, DNN, GRU, BNN (point estimation), and B-LSTM (point estimation). The evaluation indicators of various algorithms under different training sets are shown in Table 3.

As shown in Table 3, the B-GRU point estimation prediction error is smaller than that of the B-LSTM, BNN, RNN, LSTM, DNN, and GRU when the length of training data is shorter than 380 h. Using the MSE as the comparison index, the error comparison chart in Fig. 6 illustrates the trend of the prediction errors of these algorithms with the training data.

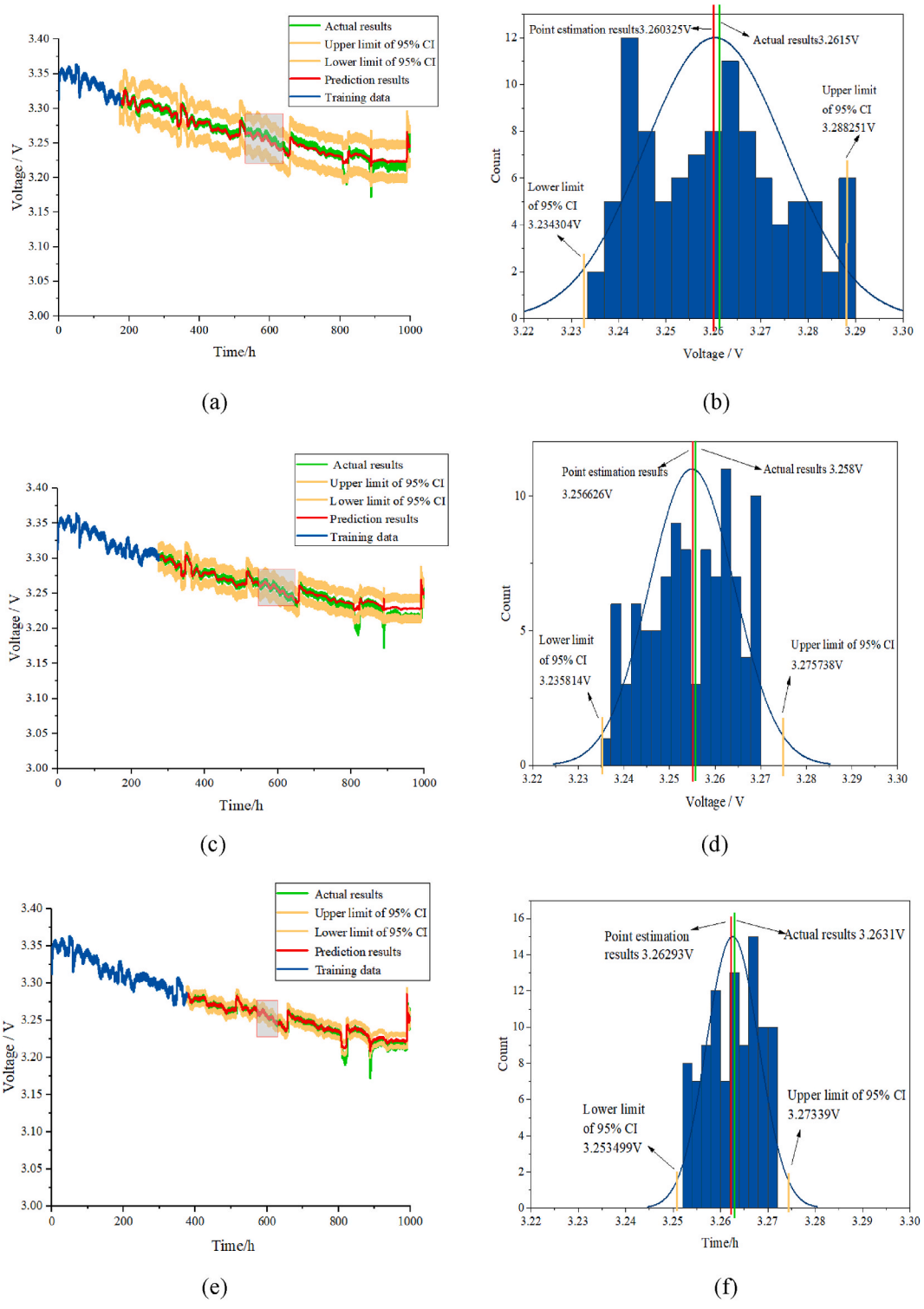

Fig. 9. B-GRU model interval estimation and probability density distribution of its prediction results at time 580 h (a) Interval estimation with training data length of 180 h. (b) Probability density distribution of prediction results at 580 h in Fig. 9(a). (c) Interval estimation with training data length of 280 h. (d) Probability density distribution of prediction results at 580 h in Fig. 9(c). (e) Interval estimation with training data length of 380 h, (f) Probability density distribution of prediction results at 580 h in Fig. 9(e).

Table 5
Performance improvement results for B-GRU point estimation and interval estimation compared with BNN.

Training data	Point estimation MSE reduction ratio (ξ_1)	Interval estimation concentration ratio (ξ_2)
80 h	28.7%	9.23%
180 h	7.55%	7.05%
280 h	5.71%	6.11%
380 h	3.23%	5.39%
480 h	3.61%	1.57%

Table 6
Performance improvement results for B-GRU point estimation and interval estimation compared with B-LSTM.

Training data	Point estimation MSE reduction ratio (ξ_1)	Interval estimation concentration ratio (ξ_2)
80 h	22.44%	7.35%
180 h	5.04%	6.47%
280 h	3.51%	6.11%
380 h	1.96%	4.38%
480 h	1.56%	1.74%

Compared to the conventional neural networks without Bayesian inference (e.g., RNN, LSTM, DNN, and GRU), the prediction accuracy of the neural network with Bayesian inference (BNN, B-GRU, and B-LSTM) is significantly improved when the length of training data is shorter than 380 h. This illustrates the superiority of the Bayesian part. The Bayesian-based neural network can quantify the uncertainty of the data. Specifically, they can achieve high accuracy with insufficient training data. At the same time, the prediction results of the Bayesian-based neural network fluctuate within a certain range. Their point estimation results are calculated based on multiple prediction results, which ensure low uncertainty and accuracy of point estimations.

Compared with BNN and B-LSTM, the prediction results of B-GRU also show its superiority, which may illustrate the superiority of the GRU part. Specifically, GRU and LSTM belong to recurrent neural networks, which are more suitable for life prediction of time series than DNN (neural network part of BNN).

Although the B-GRU prediction results cannot always be maintained at the lowest level, the B-GRU point estimation errors are smaller than those of BNN, RNN, LSTM, DNN, and GRU before 380 h and even smaller than those of RNN and LSTM at 480 h and 580 h. The results demonstrate the suitability to predict fuel cell performance using the superiority of the B-GRU point estimation with a small amount of data.

4.2. Interval estimation of B-GRU

Interval estimation is very important for the quantification of uncertainty. The distribution of the prediction results output by the B-GRU is similar to the Gaussian distribution. The prediction of uncertainty quantification can be realized by selecting different CIs.

4.2.1. Performance analysis under noises effect

During data acquisition, noise is an inevitable source of uncertainty. In many data-driven methods, it is usually necessary to filter the raw data to achieve a better prediction effect [32]. To compare B-GRU's prediction performance, raw and filtered data are put into B-GRU and compared.

The Averaging Filtering (AF) is adopted to process the raw data of the PEMFC, which is a commonly used method to filter out noise [33]. According to the analysis results of the Random Forest, it is necessary to use the AF to preprocess the 18 kinds of data. After smoothing, the uncertainty in the data would be reduced. **Appendix D** shows the raw and filtered data with a relatively large Gini coefficient.

The raw data and filtered data of the first 480 h are used for the training of B-GRU, and the results of interval estimation are shown in Fig. 7. The interval estimation results based on the filtered data are more compact due to the less uncertainty created by the filtered data. Compared to the raw data, the predicted results have a tighter kernel distribution. It may be possible to understand the problem more clearly by simultaneously comparing the voltage probability density distributions of the raw data and the filtered data. Fig. 7(b) and (c) show the voltage distribution at 590 h, respectively. When the confidence coefficient is selected as 0.95, the CI length would only make a slight difference. The CI based on filtered data is [3.2371, 3.25602], and the confidence interval length is 0.01892. The confidence based on raw data is [3.2348, 3.2562], and the confidence interval length is 0.0214.

This comparison result shows that the noise contained in the raw data has little impact on the prediction results of B-GRU. The original data would be denoised to obtain the neural network parameters with higher reliability in a large number of data-driven life predictions. However, B-GRU can also obtain the ideal prediction results with the raw data, which is the advantage of B-GRU.

4.2.2. Performance analysis for different forecast times

This section uses the operating data of the fuel cell for the first 380 h as the training set. The interval estimation and kernel distribution of the prediction results at different times are provided in Table 4 and Fig. 8, respectively. The green line represents the actual stack voltage, and the red line is the point estimation result of the B-GRU model. It can be observed that the kernel distribution is around the point estimation in the range of 400–800 from the probability density distribution diagram (in Fig. 8).

Furthermore, with a longer forecast horizon, the error between the point estimation based forecast result and the actual result does not change significantly, and the length of the confidence interval (CI) is kept around 0.02, which illustrates that the B-GRU model is credible for the forecast result in the next 400 h.

Moreover, the uncertainty contained in RUL could be presented as confidence in the prediction results. The distribution of the probability density could characterize the reliability of the prediction method. The amount of the training data would have a certain impact on the RUL prediction result, and selecting an appropriate training data set would also have a certain impact on the confidence of the RUL prediction result.

4.2.3. Performance analysis for different training times

This section uses 180 h, 280 h, and 380 h data for training. The prediction results and the probability density distribution corresponding to 580 h are provided as follows.

In order to facilitate comparative analysis, Fig. 9(b), (d), and (f) are set to the same coordinates in this section. With the increase of training data, the point estimation results of the three are not much different. In other words, the point estimation is less affected by the training data set. However, the interval estimation results gradually concentrate on the actual results, and the length of the CI gradually shortens. The increase in training data has a positive effect on the results of interval estimation. Overall, the predicted stack voltages are generally within the 95% confidence interval.

4.3. Comparative analysis of B-GRU

As mentioned in the Introduction section, B-GRU is mainly proposed to complete the prediction of time series data and quantify the uncertainty. Despite the positive results discussed in Sections 4.1 and 4.2, a further comparison should be made to demonstrate its superiority over the state-of-the-art Bayesian neural network method.

Both BNN and B-LSTM, proposed by Wang et al. [24] and Peng et al. [25], respectively, are used to compare the performance of B-GRU point estimation in Section 4.1. An intuitive evaluation of the improvement of point estimation and interval estimation is proposed using two

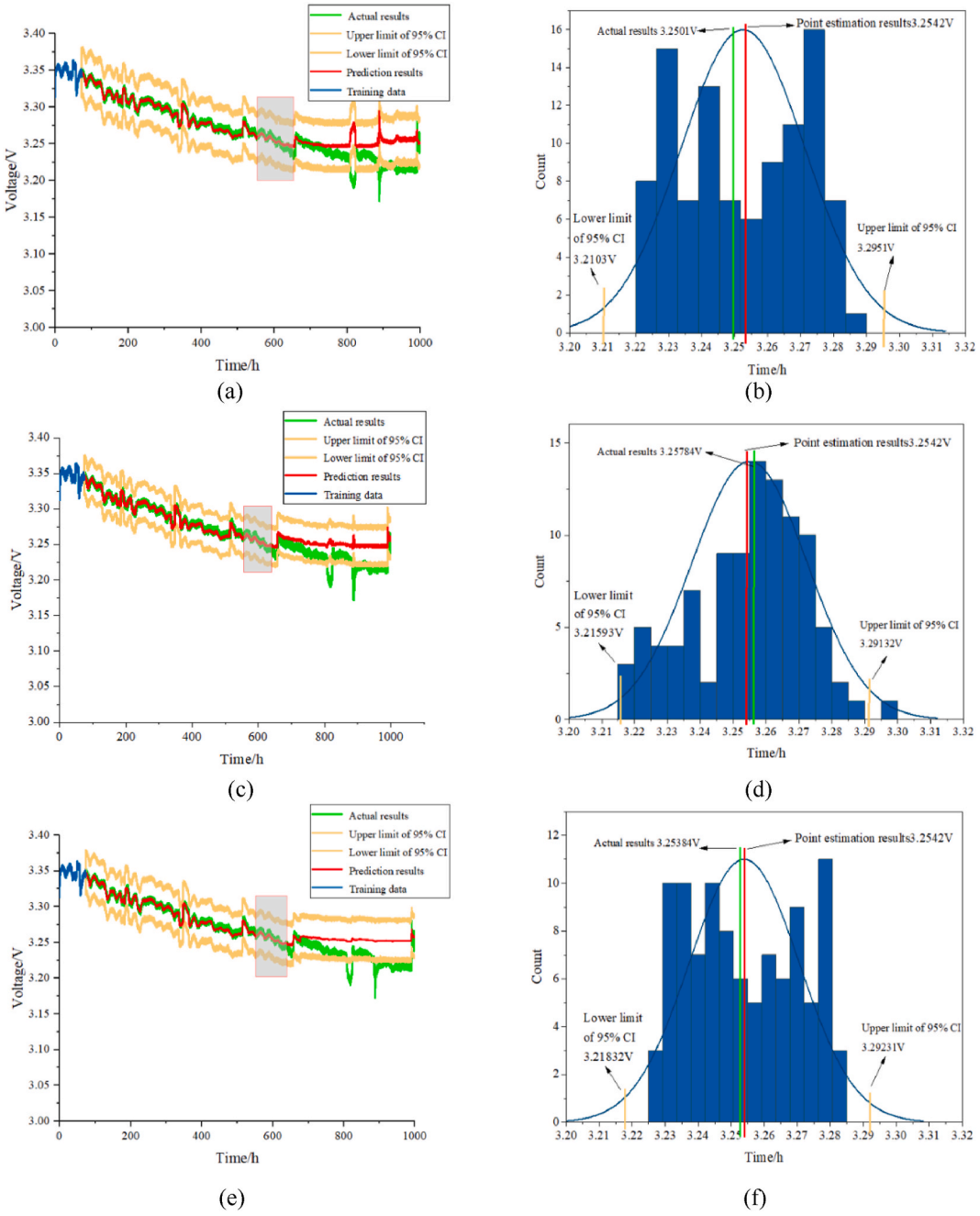

Fig. 10. Comparison of prediction results before and after improvement when the training data is 80 h. (a) interval estimation of BNN; (b) probability density distribution of BNN at 600 h; (c) interval estimation of B-LSTM; (d) probability density distribution of B-LSTM at 600 h; (e) interval estimation of B-GRU; (f) probability density distribution of B-GRU at 600 h.

indicators to demonstrate the superiority of B-GRU.

$$\xi_1 = \frac{RMS_B - RMS_{B-GRU}}{RMS_B} \times 100\% \tag{21a}$$

$$\xi_2 = \frac{\left|CI_{upper} - CI_{lower}\right|_B - \left|CI_{upper} - CI_{lower}\right|_{B-GRU}}{\left|CI_{upper} - CI_{lower}\right|_B} \times 100\% \tag{22}$$

where ξ_1 and ξ_2 represent the MSE reduction rate of point estimation and interval estimation concentration ratio. RMS_B and $RMS_{B\text{-}GRU}$ are the RMSE of the point estimation prediction of the Bayesian neural network method (BNN and B-LSTM) and B-GRU, CI_{upper} and CI_{lower} are the upper and lower bounds of the 95% confidence interval of the interval estimation, respectively, $\left|CI_{upper} - CI_{lower}\right|_B$ and $\left|CI_{upper} - CI_{lower}\right|_{B-GRU}$ are the length of the interval estimation of BNN and B-GRU, respectively. The training data is 80 h, 180 h, 280 h, 380 h, and 480 h. The performance improvement results are shown in Table 5 and Table 6.

Under the training data from 80 to 480 h, the B-GRU shows varying degrees of improvement. This trend of improvement becomes more protruding when the amount of training data is small. Specifically, when using 80 h of training data, compared with BNN, the improvement of ξ_1 and ξ_2 can reach 28.7% and 9.23%, respectively. Compared with B-LSTM, the improvement of ξ_1 and ξ_2 can reach 22.4% and 7.35%, respectively.

As is shown in Fig. 10, the kernel distribution of interval estimation in the B-GRU is more concentrated, and the point estimation result is closer to the actual value. Since all three algorithms share a common Bayesian component, their neural network part may play a more important role. Compared with BNN, benefitting from the properties of recurrent neural networks, GRU can get an expanded data set, which is trained and predicted based on the original training set and could make the training set more accurate. When compared to B-LSTM, it can be seen from Fig. 6 that although both GRU and LSTM belong to RNNs, the performance of GRU is better than that of LSTM.

However, this performance improvement gradually diminishes as the training dataset increases. Specifically, when the training data are more than 380 h, the improvement of B-GRU's prediction results in ξ_1 and ξ_2 is not obvious. With the increase of training data, the amount of data can meet the training requirements, in which case the GRU does not have significant advantages. In addition, the data expansion of GRU would bring cumulative errors and overfitting to the training process and reduce the prediction accuracy, affecting the results of subsequent uncertainty quantification.

Overall, the B-GRU outperforms the BNN and B-LSTM in point estimation and interval estimation for complex time series forecasting, especially in the scenarios with small training datasets.

5. Conclusion

In this paper, a method combining Bayesian theory and GRU is proposed for the first time. The B-GRU can realize uncertainty quantification in the prediction process and provide an important basis for the operational decision. A large number of performance studies have been carried out to verify the performance of the B-GRU, and the conclusions are as follows:

1) When the training dataset is less than 380 h, the performance of B-GRU point estimation is better than the traditional mainstream neural network.
2) B-GRU is less affected by noise and can quantify the uncertainty in the prediction process by interval estimation.
3) Compared with the cutting-edge Bayesian-based neural network, B-GRU point estimation improvements up to 28.7% (BNN) and 22.4% (B-LSTM), and interval estimation improvements up to 9.23% (BNN) and 7.35% (B-LSTM).

In future work, we will explore more appropriate parameter distributions to improve performance prediction in the hydrogen vehicle field.

CRediT authorship contribution statement

Wenchao Zhu: Data curation, Investigation, Methodology, Writing – original draft. **Bingxin Guo:** Data curation, Investigation, Software, Writing – original draft. **Yang Li:** Formal analysis, Methodology. **Yang Yang:** Conceptualization, Validation. **Changjun Xie:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing. **Jiashu Jin:** Formal analysis, Validation. **Hoay Beng Gooi:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2020YFB1506802), China Scholarship Council (202106950031) and the Office of Naval Research Global (ONRG), USA under CODE 33D, Naval Energy Resiliency and Sustainability in Broad Agency Announcement N00014-18-S-B001, and ONRG award number: N62909-19-1-2037.

Appendix

Appendix A

The optimal number of hidden layers is explored by changing the number of hidden layers under the same training dataset. The variation of the loss with epoch under different number of hidden layers is shown in Figure A.1. When the number of hidden layers is 2, the loss change is minimized, so the optimal number of hidden layers for B-GRU is 2.

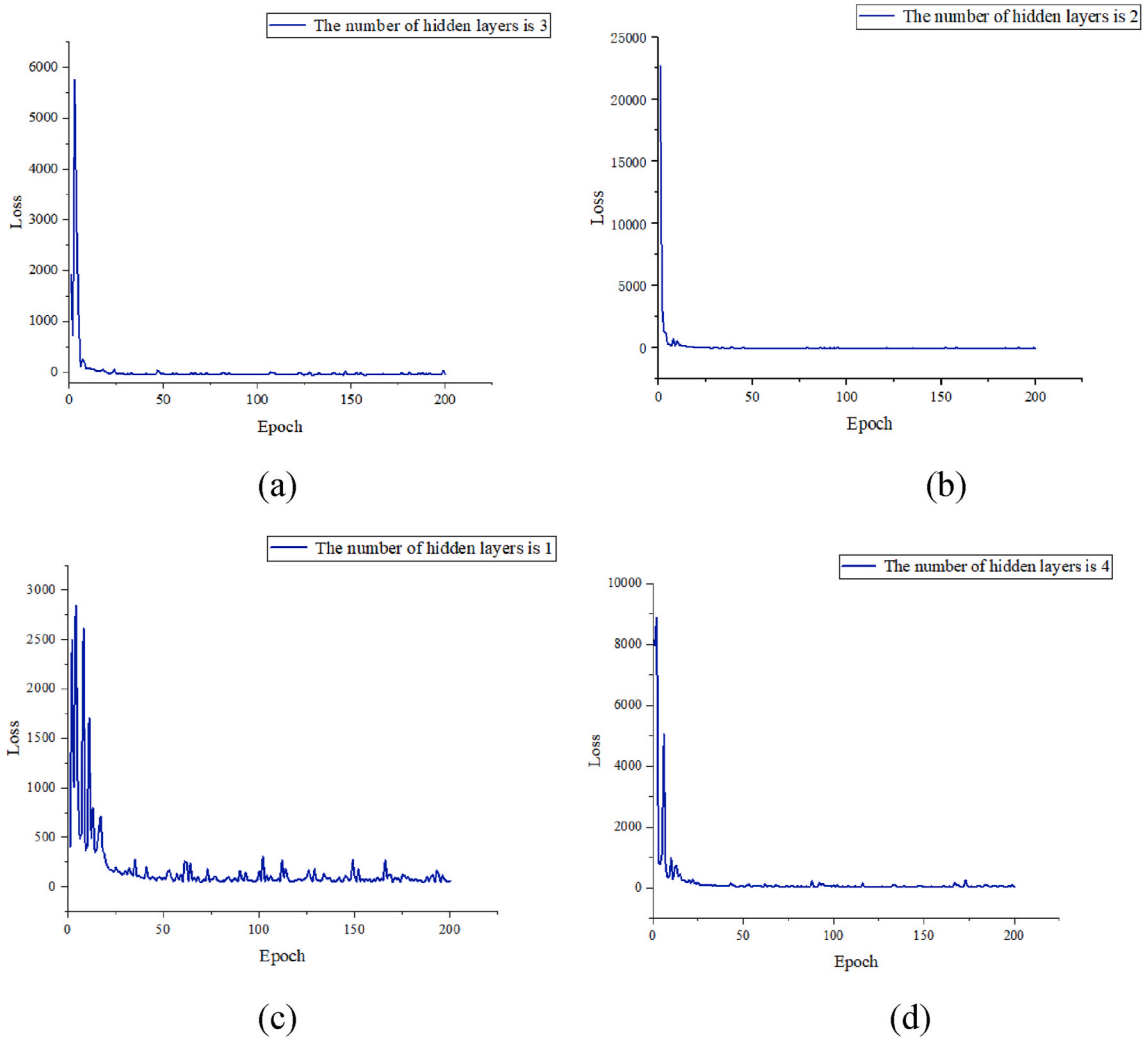

Fig. A.1. Variation of the loss with epoch under different number of hidden layers.
The number of hidden layers is (a)1; (b)2; (c)3; (d)4.

After determining the best hidden layer, under the same training data, changing the number of neurons in the second hidden layer to obtain the variation of the loss. The number of neurons varied from 5 to 30 were studied, when the number of neurons in the second hidden layer is 20, the loss can be achieved to the minimum. In order to save the length of the article, The number of neurons varied from 16 to 23 is shown in Figure A.2.

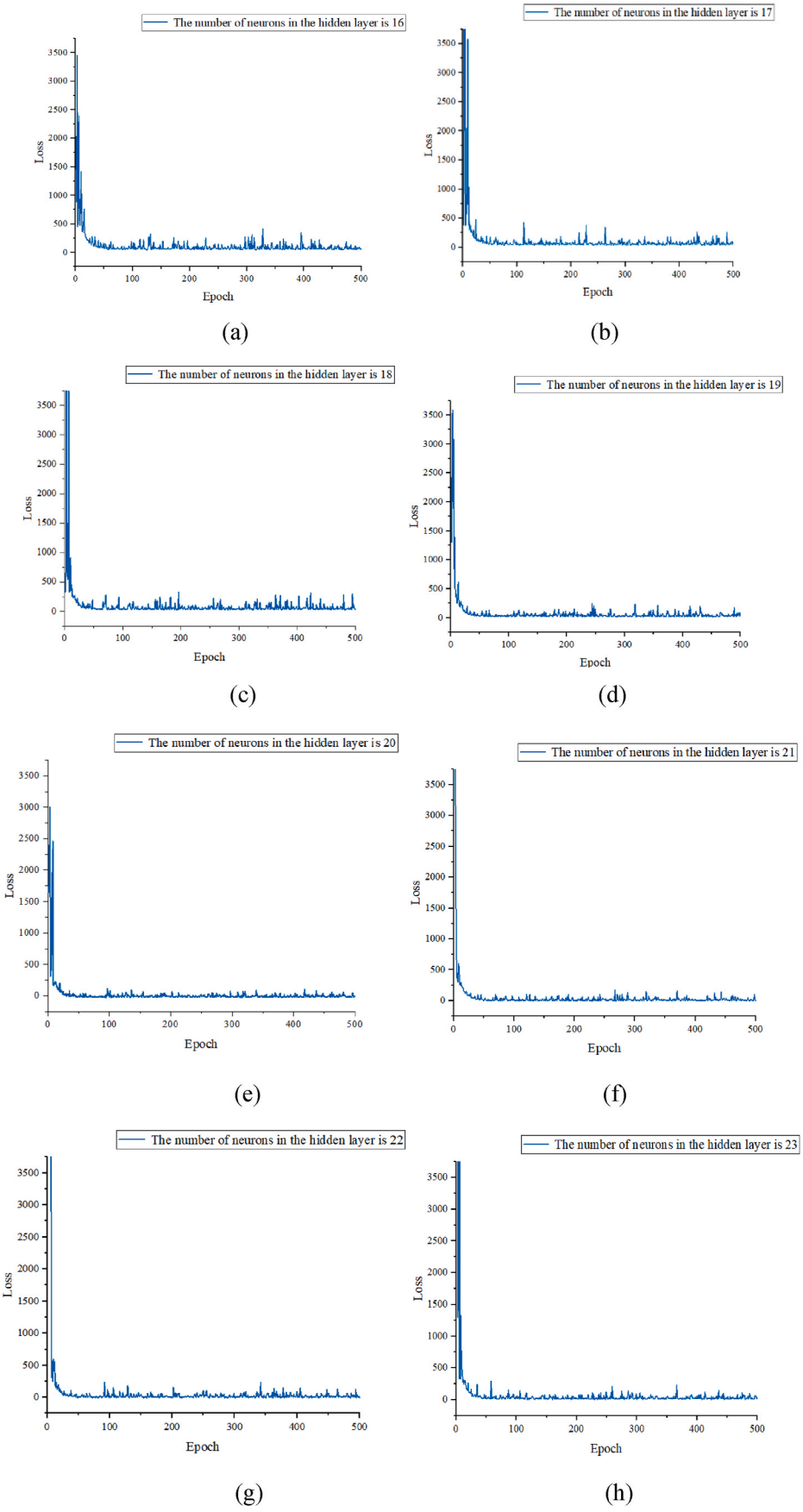

Fig. A.2. variation of the loss with epoch under different number of neurons.

Appendix B

In order to further reflect the performance of the B-GRU, Appendix B predicts the point estimation and interval estimation of the voltage decay of the fuel cell under dynamic conditions. The data under dynamic conditions fluctuates widely and is inconsistent with previous and subsequent data around 35h, 181h, 342h, 505h, 666h, and 830h, which can be considered as "failure data". Before applying these data, the data of these failure points were preprocessed using the averaging filtering (AF) method, and the preprocessed data was used for prediction.

The results of the point estimation and interval estimation are shown in Tables B.1 and B.2. Among them, Table B2 is the interval estimation based on the prediction result of 380 h in Table B1. The interval estimation obtained by using the training data with the data length of 380h is shown in Figure B.1:

Table B.1
Performance of point estimation under different length training data

Model	Training data	MSE	RMSE	MAE
B-GRU	80h	1. 657E-3	4.071E-2	1.926E-2
	180h	1. 005E-3	3.171E-2	1.330E-2
	280h	6.98E-4	2.641E-2	1.001E-3
	380h	**5.23E-4**	**2.287E-2**	**2.842E-3**
	480h	6.53E-4	2.556E-2	3.296E-3
	580h	8.21E-4	2.866E-2	3.802E-3

Table B.2
Interval estimation results at different times

Time/h	Actual value	Point estimation	95%CI	Length of CI
400	3.184	3.181	[3.164, 3.207]	0.043
450	3.200	3.205	[3.186, 3.213]	0.027
500	3.236	3.232	[3.218, 3.243]	0.025
550	3.209	3.213	[3.197, 3.221]	0.024
600	3.215	3.219	[3.186, 3.227]	0.041
650	3.213	3.220	[3.193, 3.234]	0.041
700	3.200	3.208	[3.186, 3.219]	0.033
800	3.183	3.187	[3.165, 3.206]	0.041

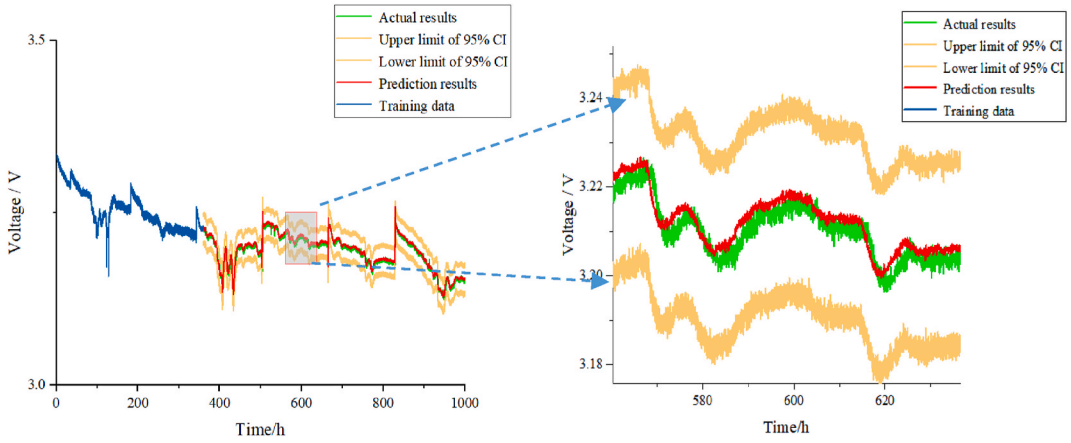

Fig. B.1. Interval estimation of B-GRU under dynamic conditions and its partial magnification

It can be seen from the prediction results that B-GRU still shows good performance under dynamic conditions. Table B1 shows that the prediction performance first decreases and then increases with the increase of training data and reaches the minimum at 380h, which is highly consistent with the performance of the algorithm on static data. In Table B2, the prediction performance of BNN over time is depicted. Point estimation results are very close to actual results, while interval estimation is closely distributed around the point estimation. Although the length of the interval estimation ranges from 0.02 to 0.04, there is an excellent quantitative performance of uncertainty.

Appendix C:

This part briefly describes the structure and parameters of the neural network used.

1) DNN: Deep neural networks have multiple hidden layers and are suitable for classification and regression problems. The DNN structure used in this paper is as follows: the number of input, hidden and output layers is 1, 2, 1; the input layer has a total of 18 neurons, and each neuron has 50 wt parameters and 50 bias parameters; the first hidden layer has a total of 50 neurons, and each neuron has 25 wt parameters and 25 bias parameters; the second hidden layer has a total of 25 neurons, and each neuron has 10 wt parameters and 10 Bias parameter; the output layer which contains 1 wt parameter and 1 bias parameter has 10 neurons.
2) RNN: RNN is a basic recurrent neural network, which has an advantage over DNN in processing time series sequences. In this paper, RNN is used to predict the data of the first 60 moments to obtain the data of the 61st moment, and so on. The structure of RNN is as follows: the number of input, hidden and output layers is 1, 2, 1; the input layer has a total of 60 neurons, and each neuron has 100 wt parameters and 100 bias parameters; the first hidden layer has a total of 100 neurons, and each neuron has 80 wt parameters and 80 bias parameters; the second hidden layer has a total of 80 neurons, and each neuron has 10 wt parameters and 10 bias parameter; the output layer which contains 1 wt parameter and 1 bias parameter has 10 neurons.
3) LSTM: The structure of LSTM is as follows: the number of input, hidden and output layers is 1, 2, 1; the input layer has a total of 60 neurons, each with 100 wt parameters and 100 bias parameters; the first hidden layer has a total of 100 neurons, and each neuron has 80 wt parameters and 80 bias parameters; the second hidden layer has a total of 80 neurons, and each neuron has 10 wt parameters and 10 bias parameter; the output layer which contains 1 wt parameter and 1 bias parameter has 10 neurons.
4) GRU: the GRU is used to predict the data of the first 60 moments, and the data of the 61st moment is obtained, and so on. The structure of the GRU is as follows: the number of input, hidden and output layers is 1, 2, 1; the input layer has a total of 60 neurons, and each neuron has 100 wt parameters and 100 bias parameters; the first hidden layer has a total of 100 neurons, and each neuron has 80 wt parameters and 80 bias parameters; the second hidden layer has a total of 80 neurons, and each neuron has 10 wt parameters and 10 bias parameter; the output layer which contains 1 wt parameter and 1 bias parameter has 10 neurons.
5) BNN: Bayesian neural network is the product of the combination of Bayesian theory and ANN. Unlike ordinary neural networks, BNN regard weights as Gaussian distributions with mean and variance. Ordinary neural networks optimize the weights, while BNN optimizes the mean and variance of the weights. The structure of the BNN is as follows: the number of input, hidden and output layers is 1, 2, 1; the input layer has a total of 8 neurons, and each neuron has 30 wt parameters and 30 bias parameters; the first hidden layer has a total of 30 neurons, and each neuron has 20 wt parameters and 20 bias parameters; the second hidden layer has a total of 20 neurons, and each neuron has 10 wt parameters and 10 bias parameter; the output layer has 10 neurons, which contains 1 wt parameter and 1 bias parameter. All weight parameters and bias parameters in BNN obey their respective Gaussian distributions. All parameters in the network have their corresponding mean and variance.
6) B-LSTM: Bayesian long short-term memory network is the combination of Bayesian theory and LSTM, which can quantify the uncertainty of time series data. The structure of the B-LSTM is as follows: the number of input, hidden and output layers is 1, 2, 1; the input layer has a total of 8 neurons, and each neuron has 50 wt parameters and 50 bias parameters; the first hidden layer has a total of 50 neurons, and each neuron has 30 wt parameters and 30 bias parameters; the second hidden layer has a total of 30 neurons, and each neuron has 10 wt parameters and 10 bias parameter; the output layer has 10 neurons, which contains 1 wt parameter and 1 bias parameter. All weight parameters and bias parameters in B-LSTM obey their respective Gaussian distributions. All parameters in the network have their corresponding mean and variance.

Appendix D:

Appendix D uses the average filtering method to smooth the 8 sets of characteristic data of the fuel cell. Due to the large number of features, Appendix D selects four features with higher Gini coefficients for visualization and shown in Figure D.1.

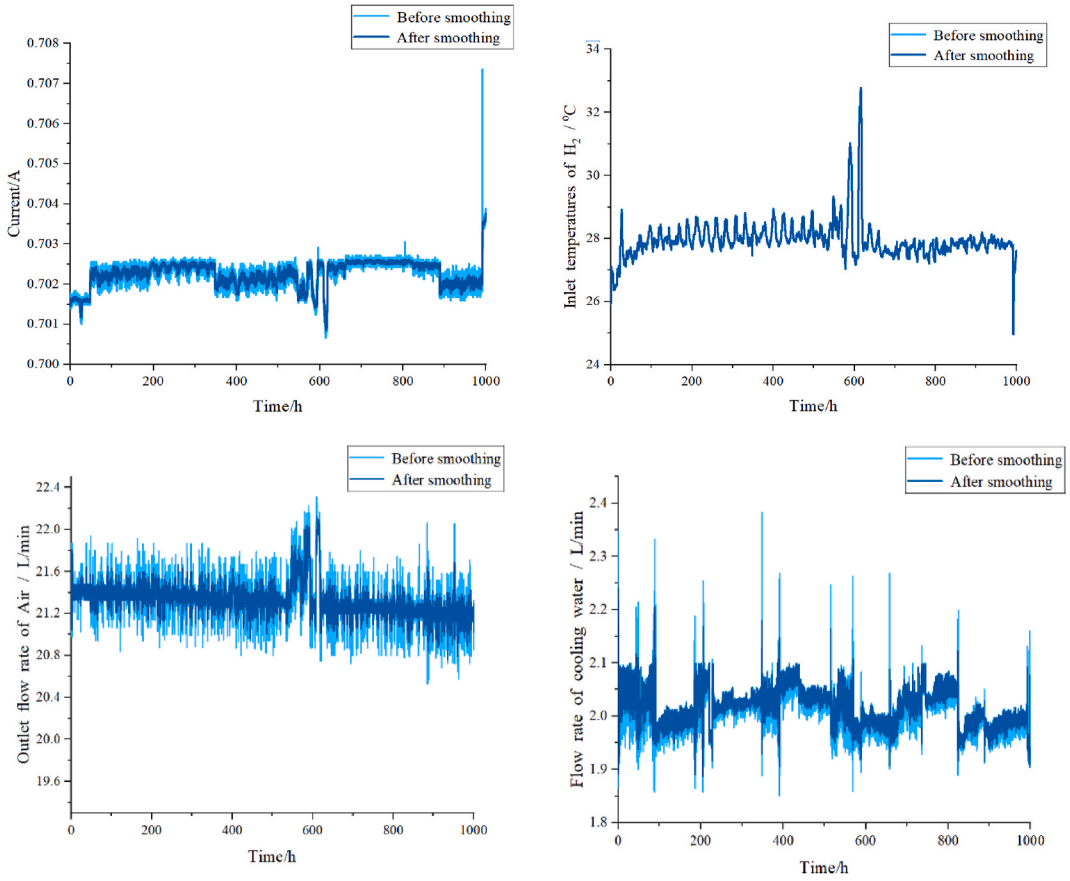

Fig. D.1. Comparison before and after the average filtering method

References

[1] Yue M, Jemei S, Gouriveau R, Zerhouni N. Review on health-conscious energy management strategies for fuel cell hybrid electric vehicles: degradation models and strategies. Int J Hydro Energy 2019;44(13):6844e61.

[2] Hao X, Yuan Y, Wang H, Ouyang M. Plug-in hybrid electric vehicle utility factor in China cities: influencing factors, empirical research, and energy and environmental application. eTransportation 2021;10:100138.

[3] Liu Jianxing, Gao Yabin, Su Xiaojie, et al. Disturbance-observer-based control for air management of PEM fuel cell systems via sliding mode technique. IEEE Trans Control Syst Technol May 2018;27(3):1129–38.

[4] Hu Z, Xu L, Li J, Ouyang M, Song Z, Huang H. A reconstructed fuel cell lifeprediction model for a fuel cell hybrid city bus. Energy Convers Manag 2018; 156:723e32.

[5] Peng H, Chen Z, Deng K, Dirkes S, Ünlübayir C, Thul A, et al. A comparison of various universally applicable power distribution strategies for fuel cell hybrid trains utilizing component modeling at different levels of detail: from simulation to test bench measurement. eTransportation 2021;9:100120.

[6] Pfeifer A, Prebeg P, Duic N. Challenges and opportunities of zero emission shipping in smart islands: a study of zero emission ferry lines. eTransportation 2020;3: 100048.

[7] Liu Jiawei, Qi Li, Chen Weirong, et al. A discrete hidden Markov model fault diagnosis strategy based on K-means clustering dedicated to PEM fuel cell systems of tramways. Int J Hydrogen Energy Jul 2018;43(27):12428–41.

[8] Wang Chu, Dou Manfeng, Li Zhongliang, et al. A fusion prognostics strategy for fuel cells operating under dynamic conditions. Etransportation May 2022:12.

[9] Yang Hao, Wang Penglei, An Yabin, et al. Remaining useful life prediction based on denoising technique and deep neural network for lithium-ion capacitors. Etransportation Jul 2021;5.

[10] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature May 2015;521(7553):436–44.

[11] Khan S, Yairi T. A review on the application of deep learning in system health management. Mech Syst Signal Process Jul 2018;107:241–65.

[12] Zhao R, an RY, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Signal Process Jan 2019;115:213–37.

[13] Ding Rui, Zhang Shiqiao, Chen Yawen, et al. Application of machine learning in optimizing proton exchange membrane fuel cells: a review. Energy and AI 2022;9: 100170.

[14] Tong Nin, Huang Weifeng, Zhang Caizhi, et al. Study of degradation of fuel cell stack based on the collected high-dimensional data and clustering algorithms calculations. Energy ang AI 2022;10:10014.

[15] Zuo Jian, Lv Hong, Zhou Daming, et al. Deep learning based prognostic framework towards proton exchange membrane fuel cell for automotive application. Appl Energy Dec 2020;281:115937.

[16] Li Zhongliang, Zheng Zhixue, Rachid Outbib. Adaptive prognostic of fuel cells by implementing ensemble echo state networks in time-varying model space. IEEE Trans Ind Electron Jan 2020;67(1):378–89.

[17] Yue Meili, Li Zhongliang, Robin Roche, et al. A feature-based prognostics strategy for PEM fuel cell operated under dynamic conditions. Prognost Syst Health Manag Conf (PHM-Besancon) Feb 2021:122–7.

[18] Li Zhongliang, Jemei S, Gouriveau R, et al. Remaining useful life estimation for PEMFC in dynamic operating conditions. IEEE Vehicle Power Propuls Conf Feb 2016:1–6. https://doi.org/10.1109/VPPC.2016.7791762.

[19] Wang Chu, Dou Manfeng, Li Zhongliang, et al. A fusion prognostics strategy for fuel cells operating under dynamic conditions. eTransportion May 2022;12:10016.

[20] Wang Chu, Li Zhongliang, Rachid Outbib, et al. Symbolic deep learning based prognostics for dynamic operating proton exchange membrane fuel cells. Appl Energy Jan 2022;305:117918.

[21] Ghahramani Zoubin. Probabilistic machine learning and artificial intelligence. Nature May 2015;521(7553):452–9.

[22] Baraldi P, Mangili F, Zio E. Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data. Reliab Eng Syst Saf Apr. 2013;112:94–108.

[23] Sankararaman S. Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. Mech Syst Signal Process Feb. 2015;52–53:228–47.

[24] Wang Ruihan, Chen Hui, Guan Cong. A Bayesian inference-based approach for performance prognostics towards uncertainty quantification and its applications on the marine diesel engine. ISA (Instrum Soc Am) Trans Nov 2020;118:159–73.

[25] Peng Weiwen, Ye Zhi-Sheng, Chen Nan. Bayesian deep learning based health prognostics towards prognostics uncertainty. IEEE Trans Ind Electron 2019;2019 (67):2283–93.

[26] Cheng Yujie, Zerhouni Noureddine, Lu Chen. A hybrid remaining useful life prognostic method for proton exchange membrane fuel cel. Int J Hydrogen Energy Dec 2018;43(27):12314–27.

[27] Scornet Erwan. Random forests and kernel methods. IEEE Trans Inf Theor 2016;62 (3):1485–500.
[28] Han Shipeng, Meng Zhen, Zhang Xingcheng, Yan Yuepeng. Hybrid deep recurrent neural networks for noise reduction of MEMS-IMU with static and dynamic conditions. Micromachines Mar 2021;12(2):214.
[29] Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: a review for statisticians. J Am Stat Assoc Feb 2017;518:859–77.
[30] Kingma DP, Welling M. Auto-encoding variational Bayes. Int Conf Learn Represent Apr. 2014.
[31] Ni Y, Jones D, Wang ZY. Consensus variational and Monte Carlo algorithms for bayesian nonparametric clustering. IEEE International Conference on Big Data; Jul 2020. p. 204–9.
[32] Žnidarič Luka, Nusev Gjorgji, Morel Bertrand, Mougin Julie, Juričić Đani, Boškoski Pavle. Evaluating uncertainties in electrochemical impedance spectra of solid oxide fuel cells. Appl Energy Jul 2021:298.
[33] Jin Jiashu, Chen Yuepeng, Xie Changjun, Zhu Wenchao, Wu Fen. Remaining useful life prediction of PEMFC based on cycle reservoir with jump model. Int J Hydrogen Energy Nov 2021;46(80):40001–13.