Multi-Step Ahead Battery SOC Estimation Using Data-Driven Prognostics and Health Management

Juliano Pimentel
j.pimentel1@unimail.derby.ac.uk
University of Derby
Derby, UK

Alistair A. McEwan
A.McEwan@derby.ac.uk
University of Derby
Derby, UK

Hong Qing Yu
H.Yu@derby.ac.uk
University of Derby
Derby, UK

Abstract

This paper proposes a data-driven multi-step ahead battery state of charge (SOC) forecasting system that can be used for prognostics and health management (PHM) of a battery management system (BMS). Two long short-term memory (LSTM) recurrent neural networks (RNN) are implemented and tested, due to their unique capability to, at the same time, use a high number of past time steps and forecast a horizon of interest at real-time. This online inference is then capable to provide an advisory window in case the BMS needs to take any preventive action. The LSTM models, a stacked-LSTM and a Bidi-LSTM, are compared to a statistical-based algorithm which uses a combination of autoregressive integrated moving average (ARIMA) with a polynomial regressor that fits measured variables into the battery SOC. The three methods are tested and validated against a wealthy battery dataset and results demonstrate the feasibility of using the Bidi-LSTM RNN as multi-step ahead SOC forecast estimator.

CCS Concepts

• **Computing methodologies → Control methods**; **Artificial intelligence**; *Computational control theory*.

Keywords

PHM, SOC Estimation, Time Series Forecasting, Multi-Step Ahead, Bidirectional LSTM, Data-Driven

ACM Reference Format:
Juliano Pimentel, Alistair A. McEwan, and Hong Qing Yu. 2018. Multi-Step Ahead Battery SOC Estimation Using Data-Driven Prognostics and Health Management. In *Proceedings of 13th International Conference on Software and Information Engineering (ICSIE 2024).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Electric vehicles (EVs) have been widely adopted aiming to reduce transportation emissions and accelerate decarbonization across the transportation sector. EVs are almost exclusively powered by lithium-ion (Li-ion) batteries in today's global automotive market [1]. Accurate state of charge (SOC) estimation of Li-ion batteries is crucial in prolonging cell lifespan and ensuring its safe operation for EV applications [2].

Modelling a battery considering all possible factors is difficult due to the complex non-linearity and time-variability of the system. Different models have been proposed: electrochemical model (EM), equivalent circuit model (ECM), and electrochemical impedance model (EIM) [3]. In many works, the model-based methods are used in conjunction with adaptive filters and state estimation algorithms. The most prominent algorithms for battery' state estimation include Kalman filters and its variants [3], particle filter [4], H∞ filter [5],[6] and sliding-mode observer [7], [8].

Herein, as in [9] we propose a fully data-driven framework to estimate the current battery SOC which is also capable of providing a multi-step ahead (MSA) advisory window for the system status to prevent faults that can lead to system mishaps.

This paper proposes a novel approach for real-time PHM of embedded battery management system (BMS), which is based on an optimized bidirectional-LSTM (Bidi-LSTM) model used for MSA forecast. The proposed model is then compared with a MSA stacked-LSTM and a regression model that uses ARIMA to forecast the measured variables and then uses a polynomial regressor to estimate the SOC. The models implementation as well as the results are presented and discussed herein. Two important research contributions are presented herein: it proposes a real-time framework capable of providing an advisory window for the system status and implements a robust Bidi-LSTM that provides consistent results tested to a forecast horizon of up to thirty time steps ahead.

This paper is structured as follows: Section 2 provides an overview of similar applications, their implementation and results, Section 3 provides a comprehensive background of the methodology and the proposed system; Section 4 presents the experimental setup and the methodology, while Section 5 the main results obtained, including the proposed framework and its results. Finally, Section 6 expands the study findings and outlines the future work.

2 Related Work

Several models and architectures have been proposed for time series forecasting. From state space model to neural networks, many approaches have been used. Seasonal ARIMA (SARIMA) is used as a state space model to compare with deep learning methods in [10]. Sequence-to-sequence RNNs and LSTMs have become popular due to their ability to learn temporal patterns and long range memory.

A multi-output (MIMO) RNN forecasting architecture, where explicit temporal dependencies between outputs capture the relationship between the predictions is proposed in [11]. This work introduces and differentiates two forecasting methods, the recursive and the multi-output. Recursive forecasting is the primary form

ICSIE 2024, December 2-4, 2024, Derby, UK

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

of multi-step forecasting [12], where forecasting is done by the factorization of previous values, amplifying potential errors and leading to lower quality predictions as the time horizon increases. The MIMO forecasting aims to estimate the future values in one step. Multi-output approaches sidestep the issue of error feedback by jointly estimating over the prediction window [11].

A review of single-output versus MIMO approaches showed direct strategy more promising choice over recursive strategy [13], and a study on the properties shows that direct strategy provides prediction values that are relatively robust and the benefits increases with the prediction horizon [14].

The applications of MSA forecasting for real-world are several [15], including models for predicting critical levels of abnormality in physiological signals, flood forecasting using RNNs, Nitrogen oxide emission forecast, electric power load forecasting and even earthquake and seismic response prediction. Yet for electrical power dispatching forecast a bidirectional LSTM (Bidi-LSTM) was used to schedule the power demand and dispatching with one day in advance window in [16]. Wind power forecasting was also proposed using LSTM in [17].

Not many works have been found for a MSA prediction of the Li-ion batteries SOC though. In [18], an ARIMA-NARX model, which is capable of predicting SOC for higher charging/discharging rates (C-rates), is proposed. NARX stands for nonlinear auto-regressive network with exogenous inputs. The use of different multi-step prediction techniques for long-term prognosis of the Lithium-ion batteries condition is presented in [19]. The paper proposed the use of adaptive neuro-fuzzy inference systems, random forest, and group method of data handling, along with various MSA strategies: iterative, direct and DirRec (which is a combination of the former ones). These methods were then evaluated for prediction of capacity over the long horizon. A novel machine-learning enabled method to perform real-time multi-forward-step SOC prediction for battery systems using a recurrent neural network with LSTM is presented in [20]. Most of the SOC predictions in published studies are basically single-step predictions/estimates based on experimental data. By using a multi-forward-step SOC prediction for battery systems, battery anomalies/faults caused by SOC anomalies (such as low SOC, SOC jumping, etc.) can also be diagnosed in advance, thereby avoiding more serious battery faults/failures or even battery thermal runaway.

3 Multi-Step Ahead Forecasting

A multi-step ahead (also called long-term) time series forecasting task consists of predicting the next H values of $(\mathbf{x}, \mathbf{y})$ as in (1) of a historical time series $[(x, y)_1 \ \ldots \ (x, y)_N]$ composed of N observations, where $H > 1$ denotes the forecasting horizon [12].

$$(\mathbf{x}, \mathbf{y}) = [(x_{k+1}, y_{k+1}) \ \ldots \ (x_{k+H}, y_{k+H})] \in \mathbb{R}^{m\times n} \tag{1}$$

3.1 Auto-regressive Integrated Moving Average (ARIMA)

In this paper, ARIMA is selected to be compared with both stacked-LSTM and Bidi-LSTM. The most well-known method called Univariate Auto-Regressive Moving Average (ARMA) for single time series' data in which Auto-Regressive (AR) and Moving Average (MA) models are combined. Univariate Auto-Regressive Integrated Moving Average (ARIMA) is a special type of ARMA where non-stationarity is taken into account in the model [21]. ARIMA is a well-used approach for time series forecasting [22].

Non-stationarity is related to time series that have means, variances and covariances that change over time. Non-stationary behaviours can be trends, cycles, random walks, or combinations of the three. Non-stationary data, as a rule, are unpredictable and cannot be modelled or forecasted. The results obtained by using non-stationary time may indicate a relationship between two variables that does not exist. To obtain reliable results, the non-stationary data needs to be transformed into stationary data [23].

ARIMA holds a relationship between the current observation and past observations (Auto-regression - AR) with a capability of differencing of actual observations in order to make the time series stationary (Integrated - I) with lags of the forecast errors of the moving average mode (Moving Average - MA).

These components are included in an ARIMA model as a set of parameters. The standard notation for the ARIMA model is usually given as $ARIMA(p, d, q)$; where p is the number of lag observations, d is the degree of differencing and q is the size of the moving average window.

Initially the system shall be assessed for non-stationarity, so the measured values $x(k)$ are replaced by the results of a recursive differencing process $\nabla^d x$, where d is the number of times the differencing process has been applied. The first order differencing is shown in (2):

$$\nabla^d x^*(k) = \nabla^{d-1}x(k) - \nabla^{d-1}x(k-1) \tag{2}$$

Finally, the ARIMA model to estimate $\hat{x}$ is given below:

$$x^*(k) = \mu + \sum_{i=1}^{p} \phi_i x^*(k-i) + \sum_{i=1}^{q} \theta_i \epsilon(k-i) + \epsilon(k) \tag{3}$$

Where , $x^*(k)$ is the current estimated value at time sequence k; $\epsilon(k)$ is the random error at time k; ϕ and θ are parameters for the AR and MA addends; and p and q are the auto-regressive and moving average specific parameters respectively.

3.2 Multivariate Multi-step Bidirectional LSTM

One solution that addresses the vanishing gradients issue for time-dependent and sequential data series is a method called long short-term memory (LSTM) [24], [25]. Unlike traditional neural networks, the recurrent neural network LSTM is an extremely efficient tool when the information is sequential [26]. This model replaces the traditional neuron of the perceptron with a memory block [27]. LSTM can learn how to bridge minimal time lags of more than 1,000 discrete time steps.

The LSTM structure calculation process, shown in Fig. 1 is such that, at each time iteration k, the hidden layer maintains a hidden state h_k, and updates it based on the layer input x_k, and previous hidden state h_{k-1}.

In Fig. 1, $x(k)$ and $y(k)$ are measurable scalar input and output, respectively. LSTM has three gates to protect and control the cell state: forget gate $F(k)$, input gate $I(k)$, and output gate $O(k)$. Additionally, $\tilde{c}(k)$ and $I(k)$ are inner states.

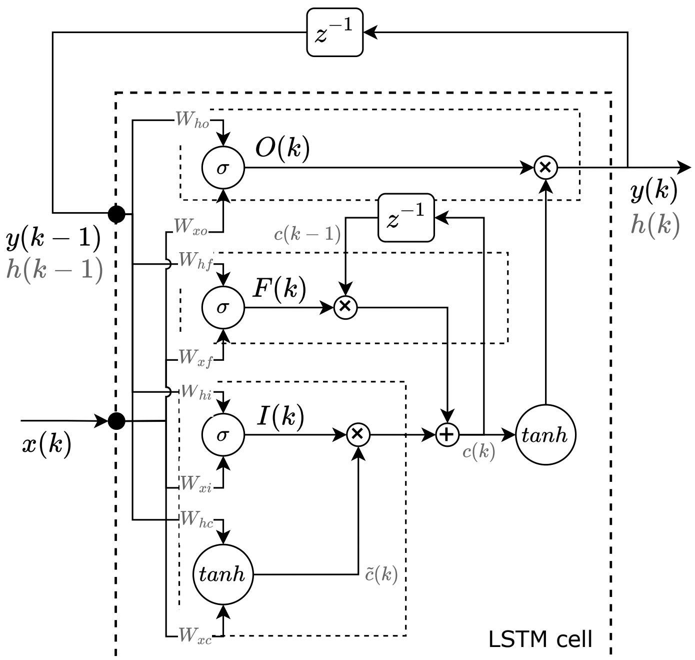

Figure 1: Long Short-Term Memory Cell

Below are the governing equations referring to the gates, states and output:

Gates:

$$f(k) = \sigma(\mathbf{W}_{hf} \cdot h(k-1) + \mathbf{W}_{uf} \cdot x(k) + b_f) \tag{4}$$

$$i(k) = \sigma(\mathbf{W}_{hi} \cdot h(k-1) + \mathbf{W}_{ui} \cdot x(k)) + b_i) \tag{5}$$

$$o(k) = \sigma(\mathbf{W}_{ho} \cdot h(k-1) + \mathbf{W}_{uo} \cdot x(k) + b_o) \tag{6}$$

States:

$$\tilde{c}(k) = \tanh(\mathbf{W}_{hc} \cdot h(k-1) + \mathbf{W}_{xc} \cdot x(k) + b_c) \tag{7}$$

$$c(k) = f(k)c(k-1) + i(k)\tilde{c}(k) \tag{8}$$

Output:

$$y(k) = h(k) = o(k)\tanh(c(k)) \tag{9}$$

Traditional RNNs including unidirectional LSTM (also called forward-pass) are suitable for processing sequential data but are trained only in the forward path. Bidirectional learning trains on both forward and reverse paths with two separate hidden layers [28] and uses the output for prediction as well. Bidi-LSTM uses the information by independently calculating both the forward path and the reverse path [9]. The output, which results from the flow of information, is also used for learning so that features are better extracted and have higher accuracy than the existing LSTM.
A RNN computes only the forward hidden sequence $\overrightarrow{h}$. By implementing the backward hidden sequence, $\overleftarrow{h}$, the output sequence y is obtained by iterating layers from $k = \{1 \dots n\}$ in the forward direction and $k = \{n \dots 1\}$ in the reverse direction [29], [30]. The formulation of the bidirectional LSTM forward direction is given in (10), the backward direction is given in (11) and the output function is finally achieved in (12):

$$\overrightarrow{h}_k = \mathcal{H}(\mathbf{W}_{x\overrightarrow{h}} \cdot x_k + \mathbf{W}_{\overrightarrow{h}\overrightarrow{h}} \cdot \overrightarrow{h}_{k-1} + b_{\overrightarrow{h}}) \tag{10}$$

$$\overleftarrow{h}_k = \mathcal{H}(\mathbf{W}_{x\overleftarrow{h}} \cdot x_k + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}} \cdot \overleftarrow{h}_{k-1} + b_{\overleftarrow{h}}) \tag{11}$$

$$\hat{y}_k = \sigma_y(\mathbf{W}_{\overrightarrow{h}y} \cdot \overrightarrow{h}_k + \mathbf{W}_{\overleftarrow{h}y} \cdot \overleftarrow{h}_k + b_k) \tag{12}$$

Fig. 2 shows an example of MSA Bidi-LSTM architecture, containing forward and backward LSTM layers. This architecture is trained to provide at each time the forecast of next H values of $(\mathbf{y}) = (y_{k+1}, \dots, y_{k+H})$, using N past time steps of the input $(\mathbf{x}) = (x_{k-N}, \dots, x_k)$.

4 Experimental Setup and Methodology

A comprehensive review of publicly available batteries datasets is presented in [31]. The dataset selected herein is the one available in [32] (under 'CC BY 4.0'), which has been cited in multiple publications, such as in [33], [34] and [35].
This battery testing dataset covers four typical driving cycles, US06, HWFET, UDDS and LA92 and a mix of other cycles. It contains data from a single 2.9 Ah NCA (Lithium Nickel Cobalt Aluminium Oxide) Panasonic 18650PF cell. A brand new 2.9 Ah Panasonic 18650PF cell was tested in an 8 $cu.ft$. thermal chamber with a 25 A, 18 V Digatron Firing Circuits Universal Battery Tester channel. The cell was cycled according to the above driving cycles and an additional "neural network driving cycle" systematically through a range of temperatures (25 °C, 10 °C, 0 °C, -10 °C, and -20 °C, in that order).

The dataset, presented in '.mat' and '.csv' files contains the voltage, current, capacity, energy and temperature from the driving cycles, sorted by temperature, test type and drive cycle.
A series of nine drive cycles were performed in the following order: Cycle 1, Cycle 2, Cycle 3, Cycle 4, US06, HWFET, UDDS, LA92, Neural Network (NN). Cycles 1-4 consist of random mix of US06, HWFET, UDDS, LA92, and Neural Network drive cycles (these been emission test cycles regulated by American authorities [36]). Neural Network drive cycle consists of combination of portions of US06 and LA92 drive cycle, and was designed to have some additional dynamics which may be useful for training neural networks. The drive cycle power profile is calculated for an electric Ford F150 truck with a 35 kWh battery pack scaled down for a single 18650PF cell. The drive cycle tests are terminated when voltage first hits 2.5 V for 25 °C.

4.1 Battery SOC Estimation

Herein, the Neural Networks (NN) drive cycle dataset at 25 °C provided in [32] is used to estimate the SOC of the proposed methods.

The widely used SOC estimation method is the Ah (Ampère-hour) Coulomb-counting method, as used in [37], and shown in (13).

$$SOC_k = SOC_{k-1} + \frac{\eta}{Q_n} \int_{k-1}^{k} \mathrm{I}_k \, \mathrm{d}k \tag{13}$$

where Q_n is the rated battery capacity, η is the charge-discharge efficiency, which is related to the battery working temperature, charge and discharge rate and other aspects; I_k is the measured

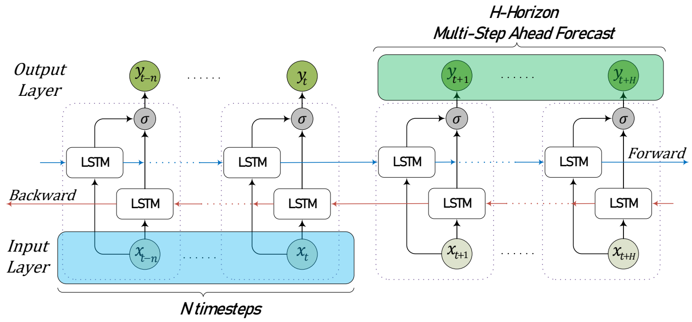

Figure 2: Bidirectional LSTM Architecture

instant current across the battery terminals.

The Coulomb-counting (CC) keeps tracking the Ampère-hour (*Ah*) and determines, therefore the battery remaining capacity. It is a time consuming process and requires a high storage capacity. Also, if the initial value of Ampère-hour is given wrong, then all estimation tends to be incorrect. So, frequent calibration are often needed to prevent accumulated errors in charge integration [38]. For the SOC estimation, The Coulomb-counting (13) is compared with three data-driven proposed methods:

- A statistical-based algorithm composed of a combination of ARIMA and polynomial regression, as explained in subsection 4.2.
- Stacked-LSTM method as elucidated in [9].
- Bidi-LSTM, as described in [9].

4.2 ARIMA and Polynomial Regression

Based on the direct relationship between the battery instant capacity and its SOC, we propose utilising ARIMA to forecast different horizons H for the capacity, given in Ah.

Then the obtained horizon is combined with recent measurements as the data for an estimator of a regression function that computes the SOC based on the forecasted capacity Ah. The regression function has the form of (14) and it is obtained using the curve fitting tool from and Simulink R2023b [39].

$$\hat{y}(k) = \beta_0 x(k) + \beta_1 + \varepsilon(k) \tag{14}$$

Where ε is a random error component.

4.3 Forecast Evaluation

Several performance metrics have been proposed for the evaluation of multi-step forecasting models, like in [40], [41], [10] and [42] . Many accuracy measures are reviewed in [41] and [10]. For datasets depending on the scale of their data, absolute and squared errors are used and known as scale-dependent measures. An option is using percentage-based measures to become more scale-independent. The use of scaled-independent data metrics is proposed in [10].

In this work, we propose the use of both scale-dependent and percentage-based metrics due to the uniformity of data and its dependency on the scale. Four metrics are selected, which are the mean absolute error (MAE), the mean squared error (MSE) and its squared-root (RMSE), and the mean absolute percentage error (MAPE). The equations are as follows:

$$MAE = \frac{1}{n}\sum_{k=1}^{n} |e_k| \tag{15}$$

where n is the number of data points, y_k are the observed (or true) values and $\hat{y}_k$ the predicted ones. The scale-dependent error e_k is defined as:

$$e_k = y_k - \hat{y}_k \tag{16}$$

The MSE and RMSE are:

$$MSE = \frac{1}{n}\sum_{k=1}^{n} e_k^2 \tag{17}$$

$$RMSE = \sqrt{MSE} \tag{18}$$

Finally, the MAPE definition is given below:

$$MAPE = \frac{1}{n}\sum_{k=1}^{n} \left(\frac{|e_k|}{|y_k|}\right) \tag{19}$$

4.4 Hardware and Software

The offline training for the data-driven PHM methods was performed on a 1xQuadro RTX8000 - 48GB GDDR6 GPU (graphical card).
The SOC estimator using ARIMA is computed using Matlab and Simulink R2023b. The data-driven PHM SOC estimator is computed using Keras [43] with a Tensorflow-backend, with Python version 3.9.7.

5 Results and Analysis

This section summarizes the main results obtained in the evaluation and comparison of the different proposed methods. Initially the CC SOC is obtained directly from the application of (13) to the available dataset, and the data-driven methods are obtained according to the previous sections.

The prediction results are related to one single run of the battery data when the SOC was slightly above 90%. For comparison, the different horizons were tested within $H = \{10, 20, 30\}$ for the three different methods.

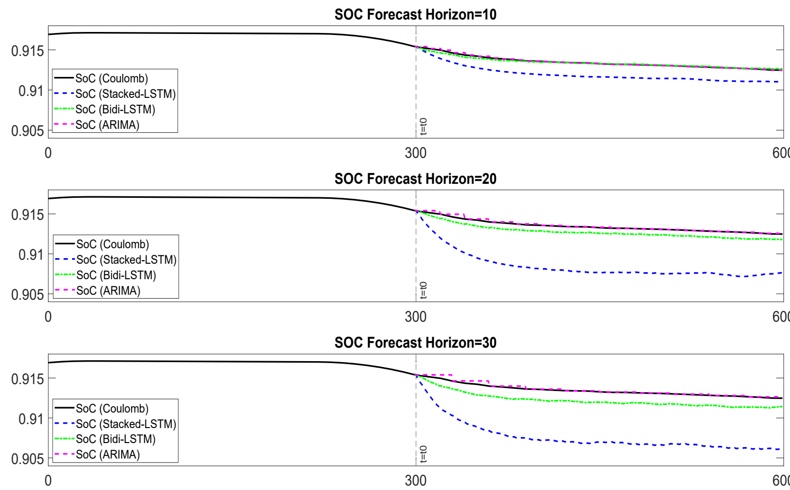

Figure 3: SOC Forecast for Different Horizons

The results can be seen in Fig. 3. It is noticeable that both ARIMA and Bidi-LSTM have good fitting to the expected values. The stacked-LSTM, despite fitting within the close range of the expected values does not perform as the other methods. Overall, in the figure, it is seen that the ARIMA outperforms the other methods.

The quantitative analysis, using the metrics detailed in section 4.3 are found in Table 1. All indicators show that statistical method ARIMA outperforms the machine learning ones, however the Bidi-LSTM has consistently better results than the stacked-LSTM. As the forecast window increases, the Bidi-LSTM falls under the same index scales of the ARIMA.

Table 1: SOC Forecast Estimation Comparison

Horizon	Method	MAE	MSE	RMSE	MAPE
10	ARIMA	1.27e−4	3.61e−8	1.90e−4	1.39e−4
	Bidi-LSTM	1.16e−3	1.76e−6	1.33e−3	1.27e−3
	Stacked-LSTM	1.94e−3	3.89e−6	1.97e−3	2.12e−4
20	ARIMA	6.62e−4	5.03e−7	7.10e−4	7.25e−4
	Bidi-LSTM	3.10e−3	1.15e−5	3.39e−3	3.41e−3
	Stacked-LSTM	5.76e−3	3.38e−5	5.81e−3	6.31e−3
30	ARIMA	1.17e−3	1.61e−6	1.27e−3	1.29e−3
	Bidi-LSTM	4.18e−3	2.14e−5	4.63e−3	4.61e−3
	Stacked-LSTM	6.77e−3	4.75e−5	6.89e−3	7.41e−3

6 Conclusions and Future Works

The presented work compares one statistical-based algorithm for the determination of the multi-step ahead forecast of a battery SOC with two data-driven machine-learning PHM methodologies.

Despite of the best results and fitting to the proposed dataset, the statistical-based algorithm, a combination of ARIMA with polynomial regression, is based on the BMS Coulomb-counting, which as stated before, is more computationally intensive and requires frequent calibration to prevent accumulated errors in charge integration [38].

Therefore, as presented in this paper, the use of a bidirectional LSTM stands as an effective data-driven option for this application, which agrees with [9], [28], [29], [44], and [15]. Besides, the Bidi-LSTM requires to be trained only once and provides a MIMO response at each time sequence with excellent fitting to the dataset. For the continuation of this work, it is envisaged to:

- Expand the use of the PHM framework for real-time forecasting models, using multi-steps ahead implementation, to allow future predictions beyond $\{k+1\}$, important aspects for critical safety applications.
- Use the proposed framework with other applications' datasets, to continue proving its generalization and robustness across different use cases.
- Implement the PHM framework on embedded devices to explore real-time edge applications where time constraints are relevant; and
- The whole system design to evolve towards the full PHM capability and remaining useful life (RUL) determination for embedded systems.

7 Acknowledgments

The authors acknowledge the University of Derby for the support of this research.

References

[1] J. Zhao, X. Feng, Q. Pang, J. Wang, Y. Lian, M. Ouyang, and A. F. Burke, "Battery prognostics and health management from a machine learning perspective," *Journal of Power Sources*, vol. 581, p. 233474, Oct. 2023.

[2] M. A. Hannan, D. N. T. How, M. S. H. Lipu, M. Mansor, P. J. Ker, Z. Y. Dong, K. S. M. Sahari, S. K. Tiong, K. M. Muttaqi, T. M. I. Mahlia, and F. Blaabjerg, "Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model," *Scientific reports*, vol. 11, no. 1, pp. 19 541–19 541, 2021.

[3] D. N. T. How, M. A. Hannan, M. S. Hossain Lipu, and P. J. Ker, "State of charge estimation for lithium-ion batteries using model-based and data-driven methods:

A review," *IEEE Access*, vol. 7, pp. 136 116–136 136, 2019.
[4] X. Liu, X. Fan, L. Wang, and J. Wu, "State of charge estimation for power battery base on improved particle filter," *World Electric Vehicle Journal*, vol. 14, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2032-6653/14/1/8
[5] S. Zhang, Y. Wan, J. Ding, and Y. Da, "State of charge (soc) estimation based on extended exponential weighted moving average h-infinite filtering," *Energies*, vol. 14, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/6/1655
[6] C. Chen, R. Xiong, and W. Shen, "A lithium-ion battery-in-the-loop approach to test and validate multiscale dual h infinity filters for state-of-charge and capacity estimation," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 332–342, 2018.
[7] Q. Chen, J. Jiang, H. Ruan, and C. Zhang, "Simply designed and universal sliding mode observer for the soc estimation of lithium-ion batteries," *IET Power Electronics*, vol. 10, no. 6, pp. 697–705, 2017. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-pel.2016.0095
[8] B. Ning, J. Xu, B. Cao, B. Wang, and G. Xu, "A sliding mode observer soc estimation method based on parameter adaptive battery model," *Energy Procedia*, vol. 88, pp. 619–626, 2016, cUE 2015 - Applied Energy Symposium and Summit 2015: Low carbon cities and urban energy systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876610216301527
[9] J. Pimentel, A. A. McEwan, and H. Q. Yu, "A novel real-time framework for embedded systems health monitoring," in *2023 26th Euromicro Conference on Digital System Design (DSD)*, 2023, pp. 309–316.
[10] E. Strøm, "Evaluation of multi-step forecasting models - an empirical deep learning study," NTNU Open Access, MSc Thesis, Aug. 21, 2021. [Online]. Available: https://hdl.handle.net/11250/2831558
[11] I. Fox, L. Ang, M. Jaiswal, R. Pop-Busui, and J. Wiens, "Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1387–1395. [Online]. Available: https://doi.org/10.1145/3219819.3220102
[12] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067–7083, 2012.
[13] S. Ben Taieb, A. Sorjamaa, and G. Bontempi, "Multiple-output modeling for multi-step-ahead time series forecasting," *Neurocomputing*, vol. 73, no. 10, pp. 1950–1957, 2010, subspace Learning / Selected papers from the European Symposium on Time Series Prediction. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231210001013
[14] G. Chevillon, "Multistep forecasting in the presence of location shifts," *International Journal of Forecasting*, vol. 32, no. 1, pp. 121–137, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207015000801
[15] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *IEEE Access*, vol. 9, pp. 83 105–83 123, 2021.
[16] J.-F. Toubeau *et al.*, "Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1203–1215, 2019.
[17] Y. Fu, W. Hu, M. Tang, R. Yu, and B. Liu, "Multi-step ahead wind power forecasting based on recurrent neural networks," in *2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2018, pp. 217–222.
[18] A. Khalid, A. Sundararajan, and A. I. Sarwat, "A multi-step predictive model to estimate li-ion state of charge for higher c-rates," in *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2019, pp. 1–6.
[19] R. Razavi-Far, S. Chakrabarti, and M. Saif, "Multi-step-ahead prediction techniques for lithium-ion batteries condition prognosis," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 004 675–004 680.
[20] J. Hong, Z. Wang, W. Chen, L.-Y. Wang, and C. Qu, "Online joint-prediction of multi-forward-step battery soc using lstm neural networks and multiple linear regression for real-world electric vehicles," *Journal of Energy Storage*, vol. 30, p. 101459, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X20300396
[21] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401.
[22] S. Masum, Y. Liu, and J. Chiverton, "Multi-step time series forecasting of electric load using machine learning models," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, Eds. Cham: Springer International Publishing, 2018, pp. 148–159.
[23] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
[24] R. C. Staudemeyer and E. R. Morris, "Understanding lstm – a tutorial into long short-term memory recurrent neural networks," 2019.
[25] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 115–143, mar 2003. [Online]. Available: https://doi.org/10.1162/153244303768966139
[26] J. Gonzalez and W. Yu, "Non-linear system modeling using lstm neural networks," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 485–489, 2018, 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318310814
[27] E. López, C. Valle, H. Allende, E. Gil, and H. Madsen, "Wind power forecasting based on echo state networks and long short-term memory," *Energies*, vol. 11, no. 3, 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/3/526
[28] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," 2019.
[29] C. W. Hong, C. Lee, K. Lee, M.-S. Ko, D. E. Kim, and K. Hur, "Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction," *Sensors*, vol. 20, no. 22, 2020.
[30] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.
[31] G. dos Reis, C. Strange, M. Yadav, and S. Li, "Lithium-ion battery data and where to find it," *Energy and AI*, vol. 5, pp. 100 081–, 2021.
[32] P. Kollmeyer, "Panasonic 18650pf li-ion battery data," Jun 2018. [Online]. Available: https://data.mendeley.com/datasets/wykht8y7tg/1
[33] 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 (1982)*, vol. 65, no. 8, pp. 6730–6739, 2018.
[34] C. Li, F. Xiao, and Y. Fan, "An approach to state of charge estimation of lithium-ion batteries based on recurrent neural networks with gated recurrent unit," *Energies*, vol. 12, no. 9, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/9/1592
[35] K. L. Wong, M. Bosello, R. Tse, C. Falcomer, C. Rossi, and G. Pau, "Li-ion batteries state-of-charge estimation using deep lstm at various battery specifications and discharge cycles," in *Proceedings of the Conference on Information Technology for Social Good*, ser. GoodIT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 85–90. [Online]. Available: https://doi.org/10.1145/3462203.3475878
[36] S. Singh, M. J. Kulshrestha, N. Rani, K. Kumar, C. Sharma, and D. Aswal, "An overview of vehicular emission standards," *Mapan*, vol. 38, no. 1, pp. 241–263, 2023.
[37] Y. Chen, D. Huang, Q. Zhu, W. Liu, C. Liu, and N. Xiong, "A new state of charge estimation algorithm for lithium-ion batteries based on the fractional unscented kalman filter," *Energies*, vol. 10, no. 9, 2017. [Online]. Available: https://www.mdpi.com/1996-1073/10/9/1313
[38] A. Basia, Z. Simeu-Abazi, E. Gascard, and P. Zwolinski, "Review on state of health estimation methodologies for lithium-ion batteries in the context of circular economy," *CIRP Journal of Manufacturing Science and Technology*, vol. 32, pp. 517–528, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1755581721000249
[39] The MathWorks Inc., *Curve Fitting Toolbox*, Natick, Massachusetts, United State, 2023. [Online]. Available: https://www.mathworks.com/products/curvefitting.html
[40] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[41] C. Chen, J. Twycross, and J. M. Garibaldi, "A new accuracy measure based on bounded relative error for time series forecasting," *PLOS ONE*, vol. 12, no. 3, pp. 1–23, 03 2017.
[42] H. Hewamalage, K. Ackermann, and C. Bergmeir, "Forecast evaluation for data scientists: Common pitfalls and best practices," 2022. [Online]. Available: https://arxiv.org/abs/2203.10716
[43] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.
[44] Z. Cui *et al.*, "Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values," *Transportation Research Part C: Emerging Technologies*, vol. 118, p. 102674, 2020.

Received 03 August 2024; revised 15 September 2024; accepted 15 September 2024