Protocol for state-of-health prediction of lithium-ion batteries based on machine learning

Accurate estimates of State of Health (SoH) are critical for characterizing the aging of lithium-ion batteries. This protocol combines feature extraction and a representative machine learning algorithm (i.e., least-squares support vector machine) for SoH prediction of lithium-ion batteries. We detail the step-by-step estimation process, followed by validation of the constructed model with a maximum absolute error of 1.62%. Overall, the proposed approach can efficiently track the aging trajectory and ensure precise SoH prediction.

---

Xing Shu, Shiquan Shen, Jiangwei Shen, Yuanjian Zhang, Guang Li, Zheng Chen, YongGang Liu

chen@kust.edu.cn (Z.C.)
andyliuyg@cqu.edu.cn (Y.L.)

Highlights

Experimental design recommended for battery degradation

Guidelines for health feature extraction with incremental capacity analysis

A step-by-step protocol from feature extraction to state of health estimation

Widely applicable to state-of-health estimation using machine learning algorithms

Shu et al., STAR Protocols 3, 101272
June 17, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.xpro.2022.101272
Protocol

Protocol for state-of-health prediction of lithium-ion batteries based on machine learning

Xing Shu,1 Shiquan Shen,1 Jiangwei Shen,1 Yuanjian Zhang,2 Guang Li,3 Zheng Chen,1,5,6,* and YongGang Liu4,*

1Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming 650500, China
2Department of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU Leicestershire, UK
3School of Engineering and Materials Science, Queen Mary University of London, E1 4NS London, UK
4State Key Laboratory of Mechanical Transmissions & College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China
5Technical contact
6Lead contact
*Correspondence: chen@kust.edu.cn (Z.C.), andyliuyg@cqu.edu.cn (Y.L.)
https://doi.org/10.1016/j.xpro.2022.101272

SUMMARY

Accurate estimates of State of Health (SoH) are critical for characterizing the aging of lithium-ion batteries. This protocol combines feature extraction and a representative machine learning algorithm (i.e., least-squares support vector machine) for SoH prediction of lithium-ion batteries. We detail the step-by-step estimation process, followed by validation of the constructed model with a maximum absolute error of 1.62%. Overall, the proposed approach can efficiently track the aging trajectory and ensure precise SoH prediction.

For complete details on the use and execution of this protocol, please refer to Shu et al. (2021b).

BEFORE YOU BEGIN

The estimation precision of State of Health (SoH) is significant to guarantee operation safety and prevent latent failures of lithium-ion batteries (Roman et al., 2021), and this protocol is a systematic prediction process of SoH for lithium-ion batteries based on machine learning algorithms. An architecture of the test bench is demonstrated in Figure 1, and it mainly consists of a battery tester for charging and discharging operations, a programmable thermal chamber for ambient temperature control, and a host computer for control and data acquisition. Moreover, to acquire and analyze the experimental data, two software, including MITS Pro and MATLAB, which respectively accounts for battery tester control and data analysis, is installed in the host computer. The software setup described in this protocol is assembled on Windows 10. However, all mentioned tools and packages function can operate properly across different platforms, including Linux and Mac operating systems.

Test bench fabrication

© Timing: 2 h

1. Preparation of the BT-5HC system.
   a. Ensure that the BT-5HC is disconnected from the power supply.
   b. Connect the BT-5HC and host computer through TCP/IP.
c. Connect the positive and negative electrodes of the battery to a channel on the BT-5HC through the cable and battery clamp.
d. Make sure that the cables are correctly connected, and the power supply are connected to the power grid port.

2. Preparation of the thermal chamber.
   a. Ensure the thermal chamber is disconnected from the power supply.
   b. Place the battery in the thermal chamber with reliable electric insulation from the inside wall.
   c. Plug in the power supply and verify that the equipment runs normally.

△ CRITICAL: Ensure that the power supply is disconnected before setting up the equipment.
Moreover, and mostly important, confirm that the positive and negative terminals of the battery are connected correctly.

**Installation**

© Timing: 1–2 h

3. Install MITS Pro 7 or latest version (depending on the available version for the operation system).
   a. Download the relevant software to the host computer from https://www.arbin.com/software/.
      More information in this regard can be referred to the Arbin website: https://www.arbin.com/.

4. Install the MATLAB on the host computer.
   a. The MATLAB can be downloaded from https://uk.mathworks.com/downloads/web_downloads/. After registration, a free 30-days trial can be requested.
   b. For a permanent installation, an associated license can be purchased for use by commercial or government organizations, degree-granting institutions or individuals.

**Alternatives:** MATLAB can be substituted by other related analysis software, such as Python, but this protocol is mainly developed based on the MATLAB.

---

**Figure 1. The architecture of the test bench**

The whole test bench includes a thermal controlled chamber (manufactured by Pindun Experimental Equipment Co., Ltd, Shanghai, China), a battery tester BT-SHC (manufactured by Arbin Instruments, College Station, Texas, United States) and a host computer. The main function of the thermal chamber is to control the ambient temperature of lithium-ion batteries according to the test requirement. The BT-SHC is utilized to charge and discharge the battery based on the designated process. The host computer mainly takes charge of setting up the testing protocol, recording the measure in real-time and conducting the necessary protection.
5. Download and install the least-squares support vector machine (LS-SVM) package:
   a. Download the LS-SVM package from https://www.esat.kuleuven.be/sista/lssvmlab/.
   b. Extract the LS-SVM package and copy all files to the toolbox folder of MATLAB, e.g.,
      "C:\Program Files\MATLAB\R2021b\toolbox".
   c. Open the MATLAB desktop.
   d. In the MATLAB “Command Window” type: addpath ("fullpathToGrafeoFolder"), using the
      example above, addpath ("C:\Program Files\MATLAB\R2021b\toolbox\LSSVMlabv1_8_R2009b_R2011a"), and press “enter”.

### KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---------------------|--------|------------|
| Battery             | Lishen | LR2170SA 4.0Ah |
| Battery tester BT-SHC | Arbin   | https://www.arbin.com/ |
| Thermal controlled chamber | Pindun | http://www.021-pd.com |
| Host computer       | Dell   | https://www.dell.com/en-us |
| Software and algorithms | Matlab MathWorks | https://www.mathworks.com |
| LS-SVM MATLAB script package | (Suykens et al., 2002) | https://www.esat.kuleuven.be/sista/lssvmlab/ |
| Other               | Tianjin Lishen Battery Joint-Stock Co., Ltd | http://en.lishen.com.cn/ |

**Note:** This protocol is conducted on nickel-cobalt-manganese (NCM)/graphite cylindrical (21700) lithium-ion cells manufactured by Tianjin Lishen Battery Joint-Stock Co., Ltd., but it can be extended to other lithium-ion batteries.

### STEP-BY-STEP METHOD DETAILS

In this section, the major steps and comprehensive step-by-step protocol are laid out for SoH prediction of lithium-ion batteries. We begin by illustrating how to acquire the experimental data based on the assembled test bench. Then, the data preprocessing procedures are demonstrated, followed by feature extraction. Finally, the SoH prediction model is trained and validated.

#### Battery test

- **Timing:** ~100 days (depending on the charging and discharging rates).

1. Discharge the battery with 1C current (C denotes the rated battery capacity with the unit Ampere hour), until the terminal voltage reaches the predefined discharge cut-off voltage.
   a. According to the instructions provided by the battery manufacturer, the cut-off voltage is predefined as 2.75 V (Shu et al., 2020). For the purpose of avoiding over-discharge and prolonging the battery life, the discharge cut-off voltage can be reset to other values, such as 3 V.
2. After the discharging operation, the battery is shelved for 5 min.
3. The 0.5C current is imposed to charge the battery, until the battery terminal voltage reaches the predefined charging voltage limit, i.e., 4.2 V.
4. After the charging operation, the battery is left to relax at open-circuit voltage for 5 min.
5. Repeat the above steps until the discharge capacity of the battery drops to less than 80% of the rated capacity (4,000 mAh). This value can be set on the MITS Pro software, and end-users only need to calculate 80% of the discharge capacity, i.e., 3.2 Ah in this protocol.
During the test, the charge and discharge voltage and temperature thresholds should be set reasonably to ensure that the battery will not be overcharged and over-discharged, so as to avoid battery thermal runaway, explosion and safety accidents. In this protocol, the charge and discharge voltage thresholds are set as 2.7 V and 4.25 V, and the temperature thresholds are respectively allotted from 0°C to 45°C and -20°C–60°C. (See Troubleshooting 1).

Preprocessing

- **Timing:** 1 h

6. Create an output directory and file name, and export the raw test data to the excel document using the MITS Pro software.

7. Read data (excel document) using `xlsread()` function of MATLAB. By typing:

   ```matlab
   >> [data, txt] = xlsread(filename, Sheet{SheetNum});
   ```

8. Classify the read data as charging current, charging voltage, charging capacity, charging time, discharging current, discharging voltage, discharging capacity and discharging time according to the current and step number.

9. Eliminate the data with obvious anomalies in the original data. These anomalies include abrupt changes in current and voltage caused by acquisition noise and errors. In addition, the asynchronous collection of data will also cause data anomalies. (See Troubleshooting 2).

10. Calculate SoH using the following equation (Shu et al., 2021a):

    \[
    \text{SOH} = \frac{C_{\text{Present}}}{C_{\text{Rated}}}
    \]  

    (Equation 1)

    where \( C_{\text{Present}} \) and \( C_{\text{Rated}} \) denote the present and rated capacity.

Feature extraction

- **Timing:** 0.5 h

There exist various health features used to characterize battery degradation (Hu et al., 2021), and incremental capacity (IC) curves are deemed as an efficient manner to evaluate the capacity loss of lithium-ion batteries (Bian et al., 2022). Additionally, since the discharging curves are usually stochastic, leading to difficulty of acquiring accuracy discharge capacity. In contrast, the charging process is more regular and easier to be investigated. Thus, the charging IC curves are analyzed and the health features are extracted from these curves.

11. Derive the charging IC curve, namely \( \frac{dQ_a}{dV} \) (Schaltz et al., 2021). It is enumerated using finite difference over single time step, as:

    \[
    \frac{dQ_a}{dV} \approx \frac{\Delta Q_a(k)}{\Delta V(k)} = \frac{Q_a(k) - Q_a(k - 1)}{V(k) - V(k - 1)}
    \]  

    (Equation 2)

    where \( Q_a \) and \( V \) denote charging capacity and voltage, and \( k \) means the sampling step.

12. Filter the IC curves based on Kalman filter (KF) (Figure 2) (Tang et al., 2018) according to the following steps.
a. Initialize the model parameters. The initial value of error covariance can be set to any value and will converge after continuous iteration. In this protocol, the initial value of error covariance is set to 10.

b. Generate the prior estimate of error covariance, as:

$$P_k^c = P_{k-1}^c + Q$$  \hspace{1cm} (Equation 3)

c. Calculate the Kalman gain, as:

$$K_k = P_k^c (P_k^c + R)^{-1}$$  \hspace{1cm} (Equation 4)

d. Update the state estimation measurement, as:

$$\hat{x}_k = \hat{x}_k^c + K_k (y_k - \hat{x}_k^c)$$  \hspace{1cm} (Equation 5)

e. Deduce the measurement of error covariance, as:

$$\hat{P}_k = (1 - K_k)P_k^c$$  \hspace{1cm} (Equation 6)

where $Q$ represents the process noise matrix, $P_{k-1}^c$ and $P_k^c$ are the prior and posterior error covariance matrices, $K_k$ means the Kalman gain, $R$ is the measurement noise covariance matrix, $\hat{x}_k^c$ and $\hat{x}_k$ respectively denote the priori and posterior state matrices, and $y_k$ is the measurement variable.

> CRITICAL: If the filtering performance of KF is not ideal, we can adjust the values of $Q$ and $R$ (See Troubleshooting 3).

13. Find the peak values and peak locations of the IC curves as indicated in the m-script shown in the following function:

```matlab
> for i = 1:n
>     Peak(i) = max(IC(i).After_smooth);
>     Peak_location(i) = find(IC(i).After_smooth == Peak(i));
>     Peak_value(i) = IC(i).After_smooth(Peak_location(i));
>     Peak_voltage(i) = IC(i).After_voltage(Peak_location(i));
>     Peak_capacity(i) = IC(i).After_capacity(Peak_location(i));
> end
```

Figure 2. The evolution of IC curves
(A) The original IC curves.
(B) The IC curves after smoothing.
Model training and validation

14. Divide the health features and SoH data into two groups, i.e., the training dataset and the validation dataset. A total of 500 cycles are experimented for the test battery, of which the first 70% of the data, i.e., 350 cycles, are chosen for the model training, and the rest data are used to evaluate the prediction performance (Tan et al., 2021) (see Figure 3).

Note: In the protocol, the input and output of the constructed SoH estimation model are denoted as the peak value of IC curve and SoH, respectively. Actually, other health features can also be selected. It is difficult to say that the peak value of IC curve is always the best selection. Sometimes, the health feature selection needs to be quantitatively analyzed according to some specific algorithms, such as Pearson correlation method (Gou et al., 2021) and gray relational analysis (Xu et al., 2021).

15. To start training the SoH prediction model, simply run the following command (See Troubleshooting 4):

```matlab
> [alpha, b] = trainlssvm([trainset, trainset_label, type, gam, sig2, 'RBF_kernel']);
```

△ CRITICAL: The generalization of LS-SVM model is decided by the kernel parameter “gam”, and the stability and complexity of this algorithm are influenced by the regularization parameter “sig2”. In this protocol, “gam” and “sig2” are respectively set to 464 and 11 by iterative optimization. If the results are unsatisfactory, check hyperparameters presented in the running part. We encourage users to adjust the model hyperparameters according to their own battery test dataset (See Troubleshooting 5).

16. When the model training is finished, the model parameters will be automatically saved to `[alpha, b]`, and the following function can be executed to examine the constructed LS-SVM model, as:

```matlab
> predictlabel = simlssvm([trainset, trainset_label, type, gam, sig2, 'RBF_kernel'], (alpha, b), testset);
```
EXPECTED OUTCOMES

LS-SVM produces a list of predictions for battery degradation state. These predictions are presented in a graphical format and in a tabular format (see Figure 4 and Table 1). The figure outputs the SoH evolution trend of the predictions along with the sequence of the cycles, wherein the real SoH values are presented in blue, and the estimated values are shown in pink. It can be observed that the estimation curve meets the real data consistently, and the maximum absolute error (MAE) is lower than 1.62%. Moreover, the root mean squared error (RMSE) and average absolute error (AAE) are 0.04% and 0.31%, respectively.

The LS-SVM algorithm is a typical classification and regression method, and it features the advantages of acceptable accuracy in high dimensional systems and quick computation capability. Other classification and regression methods, such as relevance vector machine (Guo et al., 2021) and random forest (Mawanou et al., 2021), can also be leveraged to estimate battery SoH. Moreover, this protocol can also be extended to other machine learning algorithms such as neural networks (including Elman neural network (Li et al., 2019), extreme learning machine (Chen et al., 2021), long-short term memory recurrent neural network (Li et al., 2020), gated recurrent unit based neural network (Chen et al., 2022)).

LIMITATIONS

Since the battery aging test needs long-term duration, this protocol only focuses on the SoH prediction at room temperature (25°C ± 2°C). The influence of temperature variation on battery life prediction and its verification process needs to be addressed, and this will be investigated in our future research. Besides, lithium-ion battery degradation mechanism analysis and SoH prediction from the perspective of electrochemistry under different working conditions are beyond the scope of this protocol and also require further exploration.

TROUBLESHOOTING

Problem 1
Battery aging test is a long-term and time-consuming process. During this process, it is possible to encounter power failure, resulting in aging interruption and discontinuous data acquisition (step 1).

Potential solution
Short term intermittence (e.g., up to 10 h) has little impact on the quality of test data. In case of long-term power failure (such as more than 2 days), the emerging outliers can be eliminated in the process of data processing.

| Table 1. Statistical results |
|-----------------------------|
| RMSE | MAE   | AAE  |
| 0.04% | 1.63% | 0.31% |
Problem 2
The dimensions of the matrix are inconsistent. We assume the dimensions of health features used for SoH prediction agree well with SoH data. However, if the data dimension health feature and SoH in the data preprocessing process are inconsistent, the mapping error of matrix dimensions will occur (step 9).

Potential solution
When eliminating abnormal data, the SoH data corresponding to health features need to be removed at the same time to make sure the dimensions of the matrices are consistent.

Problem 3
When KF algorithm is hired to filter the IC curves, the improper settings of parameters Q and R will enable that the curve after filtering is not smooth enough or reduces the peak value of the curve (step 12).

Potential solution
Generally, increasing R or decreasing Q will make the curve smoother, but the peak point will also become inconspicuous. To make the curve smooth and the peak points easy to find, we suggest the users need to adjust the values of Q and R by trial and error according to the filtering performance.

Problem 4
LS-SVM fails to process the input dataset because the health feature and corresponding SoH take row vectors as inputs (step 15).

Potential solution
Use the "Transpose" option at the time of the data processing.

Problem 5
We encourage the users to tune the hyperparameters of the LS-SVM algorithm for better performance. Different hyperparameters may lead to large differences in estimation results (step 15).

Potential solution
The generalization ability of the LS-SVM is determined by the kernel parameter, and the stability and complexity of this algorithm are influenced by the regularization parameter. To achieve the optimal SoH results, intelligent optimization algorithms, such as grid search, genetic algorithm (GA) and particle swarm optimization (PSO), can be applied to find the optimal parameters, thereby accelerating the algorithm design process.

RESOURCE AVAILABILITY

Lead contact
Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Zheng Chen (chen@kust.edu.cn).

Materials availability
This study does not yield any new unique reagents.

Data and code availability
The datasets/code supporting the current study has not been deposited in a public repository because of the requirements of confidentiality agreement, but they are available from the corresponding author on request.
ACKNOWLEDGMENTS
This work is supported by the National Natural Science Foundation of China (Grant No. 52162051 and 52172400).

AUTHOR CONTRIBUTIONS
X.S. and S.S. designed the specification of the protocol. X.S. and J.S. conducted the experiments and performed data extraction. X.S., S.S., and Y.Z. prepared the initial draft. Z.C. and G.L. revised and polished the manuscript. Z.C. and Y.L. reviewed and edited the manuscript. All authors commented on the manuscript.

DECLARATION OF INTERESTS
The authors declare no competing interests.

REFERENCES
Bian, X., Wei, Z., Li, W., Pou, J., Sauer, D.U., and Liu, L. (2022). State-of-health estimation of lithium-ion batteries by fusing an open circuit voltage model and incremental capacity analysis. IEEE Trans. Power Electron. 37, 2226–2236. https://doi.org/10.1109/TPEL.2021.3104723.

Chen, L., Ding, Y., Wang, H., Wang, Y., Liu, B., Wu, S., and Pan, H. (2021). Online estimating state-of-health of lithium-ion batteries using hierarchical extreme learning machine. IEEE Trans. Transport. Electrif. 1. https://doi.org/10.1109/TTE.2021.3107727.

Chen, Z., Zhao, H., Zhang, Y., Shen, S., Shen, J., and Liu, Y. (2022). State of health estimation for lithium-ion batteries based on temperature prediction and gated recurrent unit neural network. J. Power Sourc. 521, 230892. https://doi.org/10.1016/j.jpowsour.2021.230892.

Gou, B., Xu, Y., and Feng, X. (2021). An ensemble learning-based data-driven method for online state-of-health estimation of lithium-ion batteries. IEEE Trans. Transport. Electrif. 7, 422–436. https://doi.org/10.1109/TTE.2020.3029295.

Guo, Y., Huang, K., and Hu, X. (2021). A state-of-health estimation method of lithium-ion batteries based on multi-feature extracted from constant current charging curve. J. Energy Storage 36, 102372. https://doi.org/10.1016/j.est.2021.102372.

Hu, X., Che, Y., Lin, X., and Onori, S. (2021). Battery health prediction using fusion-based feature selection and machine learning. IEEE Trans. Transport. Electrif. 7, 382–398. https://doi.org/10.1109/TTE.2020.3017090.

Li, P., Zhang, Z., Xiong, Q., Ding, B., Hou, J., Luo, D., Rong, Y., and Li, S. (2020). State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network. J. Power Sourc. 459, 228069. https://doi.org/10.1016/j.jpowsour.2020.228069.

Li, W., Jiao, Z., Du, L., Fan, W., and Zhu, Y. (2019). An indirect RUL prognosis for lithium-ion battery under vibration stress using Elman neural network. Int. J. Hydrogen Energy 44, 12270–12276. https://doi.org/10.1016/j.ijhydene.2019.03.101.

Mawonou, K.S.R., Eddahech, A., Dumur, D., Beauvois, D., and Godoy, E. (2021). State-of-health estimators coupled to a random forest approach for lithium-ion battery aging factor ranking. J. Power Sourc. 484, 229154. https://doi.org/10.1016/j.jpowsour.2020.229154.

Roman, D., Saxena, S., Robu, V., Pecht, M., and Flynn, D. (2021). Machine learning pipeline for battery state-of-health estimation. Nat. Machine Intelligence 3, 447–456. https://doi.org/10.1038/s42256-021-00312-3.

Schultz, E., Stroe, D.I., Nørregaard, K., Ingvardsen, L.S., and Christensen, A. (2021). Incremental capacity analysis applied on electric vehicles for battery state-of-health estimation. IEEE Trans. Indus. Appl. 57, 1810–1817. https://doi.org/10.1109/TIA.2021.3052454.

Shu, X., Li, G., Shen, J., Lei, Z., Chen, Z., and Liu, Y. (2020). A uniform estimation framework for state of health of lithium-ion batteries considering feature extraction and parameters optimization. Energy 204, 119575. https://doi.org/10.1016/j.energy.2020.119575.

Shu, X., Shen, J., Li, G., Zhang, Y., Chen, Z., and Liu, Y. (2021a). A flexible state-of-health prediction scheme for lithium-ion battery packs with long short-term memory network and transfer learning. IEEE Trans. Transport. Electrif. 7, 2238–2248. https://doi.org/10.1109/TTE.2021.3074638.

Shu, X., Shen, S., Shen, J., Zhang, Y., Li, G., Chen, Z., and Liu, Y. (2021b). State of health prediction of lithium-ion batteries based on machine learning: advances and perspectives. iScience 24, 103265. https://doi.org/10.1016/j.isci.2021.103265.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2002). Least Squares Support Vector Machines (World Scientific). https://doi.org/10.1142/5089.

Tan, X., Zhan, D., Lyu, P., Rao, J., and Fan, Y. (2021). Online state-of-health estimation of lithium-ion battery based on dynamic parameter identification at multi timescale and support vector regression. J. Power Sourc. 494, 229233. https://doi.org/10.1016/j.jpowsour.2020.229233.

Tang, X., Zou, C., Yao, K., Chen, G., Liu, B., He, Z., and Gao, F. (2018). A fast estimation algorithm for lithium-ion battery state of health. J. Power Sourc. 396, 453–458. https://doi.org/10.1016/j.jpowsour.2018.06.036.

Xu, T., Peng, Z., and Wu, L. (2021). A novel data-driven method for predicting the circulating capacity of lithium-ion battery under random variable current. Energy 218, 119530. https://doi.org/10.1016/j.energy.2020.119530.