

Model-constrained deep learning for online fault diagnosis in Li-ion batteries over stochastic conditions

Received: 7 April 2024

Accepted: 31 January 2025

Published online: 14 February 2025

 Check for updatesRui Cao, Zhengjie Zhang, Runwu Shi , Jiayi Lu, Yifan Zheng , Yefan Sun, Xinhua Liu  & Shichun Yang  

For the intricate and infrequent safety issues of batteries, online safety fault diagnosis over stochastic working conditions is indispensable. In this work, we employ deep learning methods to develop an online fault diagnosis network for lithium-ion batteries operating under unpredictable conditions. The network integrates battery model constraints and employs a framework designed to manage the evolution of stochastic systems, thereby enabling fault real-time determination. We evaluate the performance using a dataset of 18.2 million valid entries from 515 vehicles. The results demonstrate our proposed algorithm outperforms other relevant approaches, enhancing the true positive rate by over 46.5% within a false positive rate range of 0 to 0.2. Meanwhile, we identify the trigger probability for four safety fault samples, namely, electrolyte leakage, thermal runaway, internal short circuit, and excessive aging. The proposed network is adaptable to packs of varying structures, thereby reducing the cost of implementation. Our work explores the application of deep learning for real-state prediction and diagnosis of batteries, demonstrating potential improvements in battery safety and economic benefits.

Lithium-ion batteries (LiBs) are currently the preferred choice for electric vehicles (EVs) known for their high specific energy, long life-span, minimal self-discharge, and wide temperature range applicability^{1–5}. However, as the market share of EVs grows, safety failures of battery packs have become a significant hurdle to their development. Recent years have seen frequent battery failures, leading to numerous market recalls, increasing user safety concerns, and economic burdens^{6–9}. Early prediction and real-time monitoring of battery failures, particularly under complex stochastic operating conditions, remain technological challenges^{10,11}. Both in market development and scientific research, an efficient online fault diagnosis solution for batteries under stochastic operating conditions is urgently needed.

Under normal conditions, the microscopic electrochemical composition of LiBs degrades irreversibly with increasing charge/discharge cycles, leading to increased internal resistance and decreased capacity until end-of-life recycling^{12,13}. However, due to unpredictable operating

conditions and abuse tolerance, the battery's charging and discharging process may occasionally lead to abnormal failures such as leakage, abnormal aging, and even severe safety incidents like short circuits, fires, and explosions^{14–16}. Battery failure is a process that evolves from microscopic reaction abnormalities to macroscopic characterization abnormalities, and then to an uncontrolled system^{17,18}. However, the unpredictable nature of failure in actual operating conditions results in significant variability in failure duration and the randomness of critical trigger points¹⁹. Coupled with the limited type and quality of monitorable data during battery operation, this greatly increases the difficulty of fault diagnosis.

Research on lithium-ion battery (LiB) fault diagnosis typically involves analyzing the monitoring object for specific fault types and designing corresponding diagnostic methods. Previous research has primarily detected faults such as external short circuit, internal short circuit (ISC), electrolyte leakage (EL), excessive aging (EA) (Capacity <80%), and thermal runaway (TR)^{20–25}. The main monitoring

parameters include voltage, temperature, current, gas production, and battery appearance images²⁶. It's worth noting that due to the limitations of obtainable macroscopic characterization data, different fault phenomena may exhibit similar macroscopic characteristics, such as temperature rise and voltage drop²⁷. The diagnostic methods used for final decision-making can be categorized into three types²⁸: ① Knowledge-based: establish judgment methods such as numerical comparison, fuzzy logic, graph theory, etc., based on experience, and formulate judgment thresholds for decision-making^{29–31}. ② Model-based: construct a battery model, obtain model parameters through parameter identification, state estimation, etc., and make judgments based on the residual difference between actual and model parameters^{32–35}. ③ Data-based: make judgments through signal processing or machine learning methods, based on extensive data analysis and training^{36–38}.

Most existing battery fault diagnosis methods are explored in experimental simulation scenarios. However, real-world monitoring must consider various factors such as the number of sensors, data quality, and random operating conditions, which often render laboratory research methods inapplicable. The research of fault diagnosis algorithms requires in-depth analysis of extensive real-world full-working condition data. Zhang et al. utilized deep learning to detect anomalies in real charging scenarios and provided the collected charging segment characterization data for future research³⁹. However, data across all operating conditions is still missing. In addition, severe faults such as ISC and TR can deteriorate rapidly within a short timeframe, making real-time diagnostic functionality essential. Therefore, the input samples for diagnostic algorithms must be compatible with random operating conditions and small sequential segments, which poses a significant challenge for current diagnostic methods. Furthermore, the online battery fault diagnosis system requires enhancement to enable not only risk identification but also precise fault type diagnosis, thereby greatly improving subsequent fault maintenance and response strategies.

In this study, we design a model-constrained neural network (MCNN) capable of diagnosing battery faults under stochastic working conditions and evaluating the trigger probabilities of four typical safety faults. The algorithm's development, training, and testing are carried out using real vehicle datasets, with data sampled at 30-s intervals per frame as input to ensure the algorithm's real-time performance and broad applicability. We publicly release a dataset of 18.2 million valid entries from 515 vehicles collected by a cloud-based data center, which we believe will contribute to future battery-related data mining work. The framework design can be summarized into four parts: constraint model construction, data prediction, feature encoding, and diagnostic output. During this process, we employ innovative detail designs such as distribution-corrected pack modeling, model-constrained autoencoder (MC-AE), and monitoring feature extraction to enhance the algorithm's performance for actual deployment. Our validation process covers the entire dataset. The algorithm's pack prediction results outperform the classical optimal scheme by 23.69%, while the cell prediction results show a performance advantage of 0.80% without specific training on the cell state. The results on fault point identification and fault type diagnosis also show a remarkable improvement, with an area under the receiver operating characteristic (AUROC) improvement of 49.0% to 63.7% in true positive rate (TPR) results over false positive rate (FPR) $\in (0, 0.2]$ compared to current typical methods. We also discuss model transferring to ensure efficient application to different pack types. In summary, our proposed method overcomes the application limitations of LiB fault diagnosis and enables online diagnosis of multiple fault types under full life cycle and full working conditions, offering substantial real-world benefits in power battery safety applications.

Results

Framework overview

Besides coping with the effects of online data fluctuations, the development of this work requires overcoming two additional challenges. The first is ensuring the real-time performance of the algorithm's decision-making, which means that the identified samples need to meet the requirements of being independent of operating conditions and time intervals. To address this, the input for our algorithm is data sampled at every individual time point, without being restricted by operating conditions. The second constraint is the lack of fault samples for algorithm training. To tackle this challenge, we divide the entire algorithm framework into two objectives. The first is to predict the state of each battery cell in normal conditions and achieve quantifiable fault identification based on prediction residual; and the second is to estimate the probability of different fault types at abnormal moments. Of these, the first objectives only require data from normal vehicle operation, which is relatively abundant, while the second step estimates probabilities based on features at abnormal moments, excluding regular operation moments. This approach significantly reduces the reliance on fault samples for training.

To achieve the objectives, we utilize a model-constrained deep learning approach to supplement the missing information regarding the battery performance's evolution. The MCNN embeds the relationship between individual model evolution and overall system coordination into the network's structural design, and the input and loss function also incorporate the model state. This enables the neural network to integrate physical model constraints during the learning process for optimization. Stochastic oscillations during training and application of the network are effectively limited to achieve more stable and accurate performance. The MCNN could be divided into three key stages: ① modeling and state resolution of individual and overall control models; ② overall state sequence prediction; and ③ encoder reconstruction based on the coupled relationship of the model to predict future states. These correspond to (a), (b), and (c) in Fig. 1.

As shown in Fig. 1a, a#, the state prediction of the battery pack and individual deviation is conducted by a distributionally corrected battery pack model with a filtering processor. The prediction results are then fed into the feature encoding module. To leverage the timing characteristics of the data for more accurate feature prediction, we design a timing prediction network module (Fig. 1b, b#) to predict the pack state and output to the feature encoding module. The feature encoding process is depicted in Fig. 1c, c#. We innovatively design the MC-AE, which embeds the calculation of hidden variables in the principle that the cell state is the sum of the pack state and the deviation state. Initially, the pack state and vehicle state are compressed by a compression layer. The low-dimensional pack state, vehicle state, and deviation state are then fused in the hidden variables. The predicted state of each cell is generated by the feature transformation fully connected layer (FTL). The design of the encoder integrates the actual physical modeling implications to ensure the physical interpretability of the regression results.

As depicted in Fig. 1d, d#, we extract monitoring features from the encoder output by using a directional linear layer to monitor these features and perform statistical regression on the samples to obtain quantifiable safety fault detection results. Simultaneously, based on the supplemented features of fault samples at abnormal moments, we regress the trigger probability of the four safety faults through four terminal fully connected layers, each directly corresponding to one type of safety fault. Ultimately, the designed deep learning network can detect and classify battery faults under online stochastic operating conditions.

Dataset analysis

In this study, we utilize data uploaded from vehicle onboard BMS for network design. We have released the real vehicle dataset of 18.2

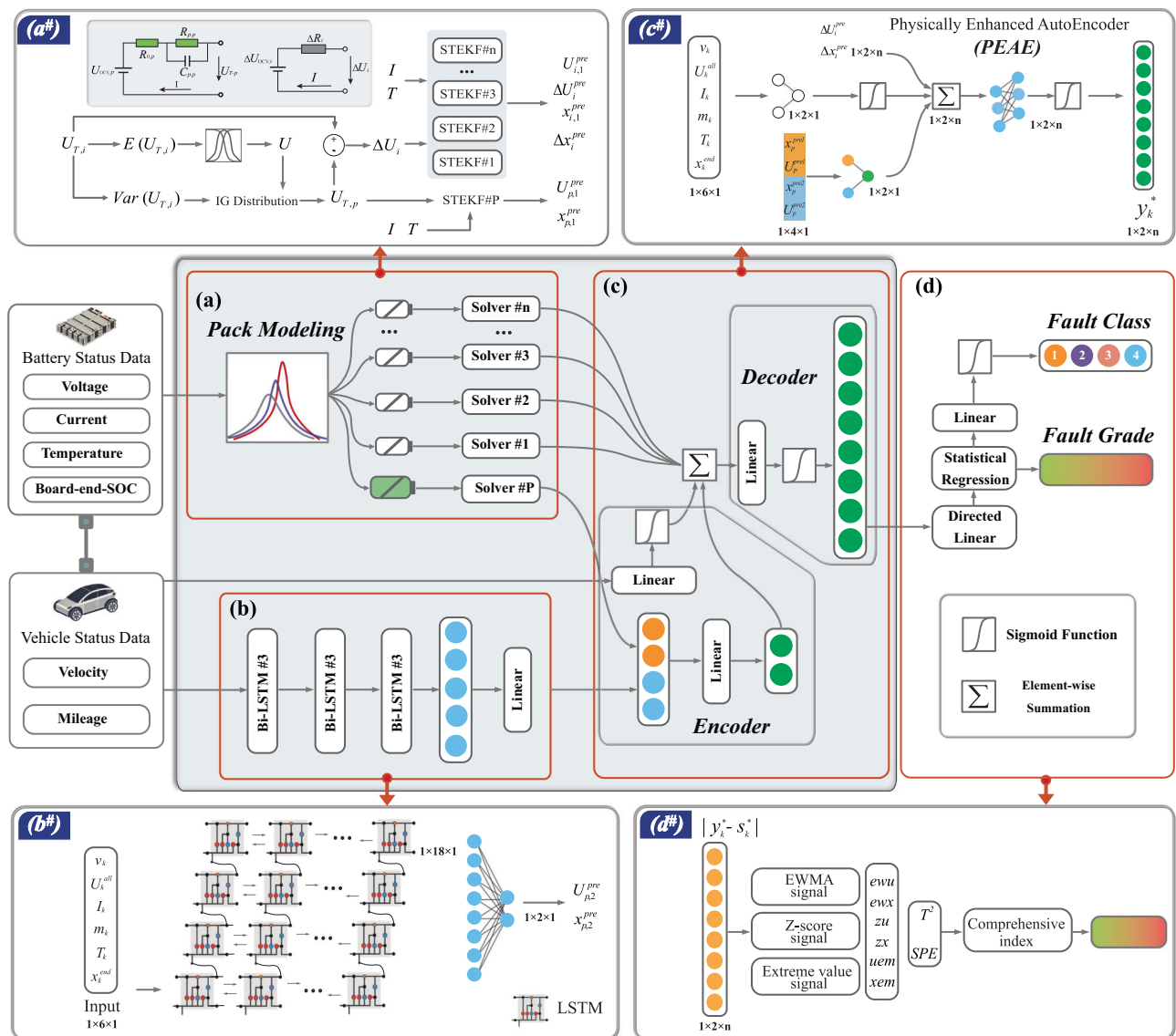


Fig. 1 | Detection framework based on the model-constrained deep learning network. The light blue part of the diagram constitutes the MCNN; **a** Modeling and state resolution of individual and overall control models: this model identifies real-time states through a specially designed pack control model and state estimator. The output state prediction parameters constrain the regression process of the network. **b** Overall state sequence prediction: this network combines battery state and vehicle state data to predict the pack state through a Bi-directional Long Short-Term Memory (BiLSTM)-based network. **c** Encoder based on the model coupling

relationship: this encoder takes the outputs of (a) and (b), along with vehicle state data, as network inputs. It compresses and fuses hidden variables based on the physical significance of the model as a whole and deviations and obtains the single predicted state through decomposition. **d** Residual monitoring module: this module extracts features from the outputs of (c), combining a directional linear layer and statistical regression for fault detection and fault classification. (#) Detailed presentation: this represents the corresponding network parts, including corresponding data flow, processing algorithm, network composition, etc.

million valid entries from 515 vehicles collected by the BMS data center on the Beihang cloud, as shown in Fig. 2a. The dataset primarily includes data from three battery manufacturers, which due to confidentiality restrictions, are referred to as DTI, QAS, and GIS in this paper. Figure 2b presents the statistics of valid data frames and the number of vehicles. In addition to normal data samples, the dataset also contains four types of hard-to-collect safety failure samples: TR, EL, ISC, and EA, as shown in Fig. 2c. The labels of the failure samples are determined by product engineers after disassembling the battery pack samples and characterizing failures retroactively. The sample set consists of multiple actual cases. The validation of the dataset allows us to evaluate the feasibility of the algorithm's fault detection in terms of the ROC effect. For specific categories of fault samples, we can determine whether it is possible to predict the faults in advance and

the probability of occurrence of the fault category. The types of online data that are generalizable are very limited. As depicted in Fig. 2d, we only use six types of feature data related to battery status and vehicle status (optional) as inputs, including voltage, temperature, current, onboard SOC, speed, and mileage. The sampling intervals meet the minimum data quality requirement of 30 s, ensuring the applicability of the network. The most direct physical quantities in the online data that reflect the battery state are voltage, current, and temperature. Comparing the distribution of values between faulty and normal samples, there is no significant difference in voltage, current, and temperature. However, the battery failure mechanism is very complex, and detecting the failure through limited types of data is challenging. This underscores the value of realizing battery fault detection in online data scenarios.

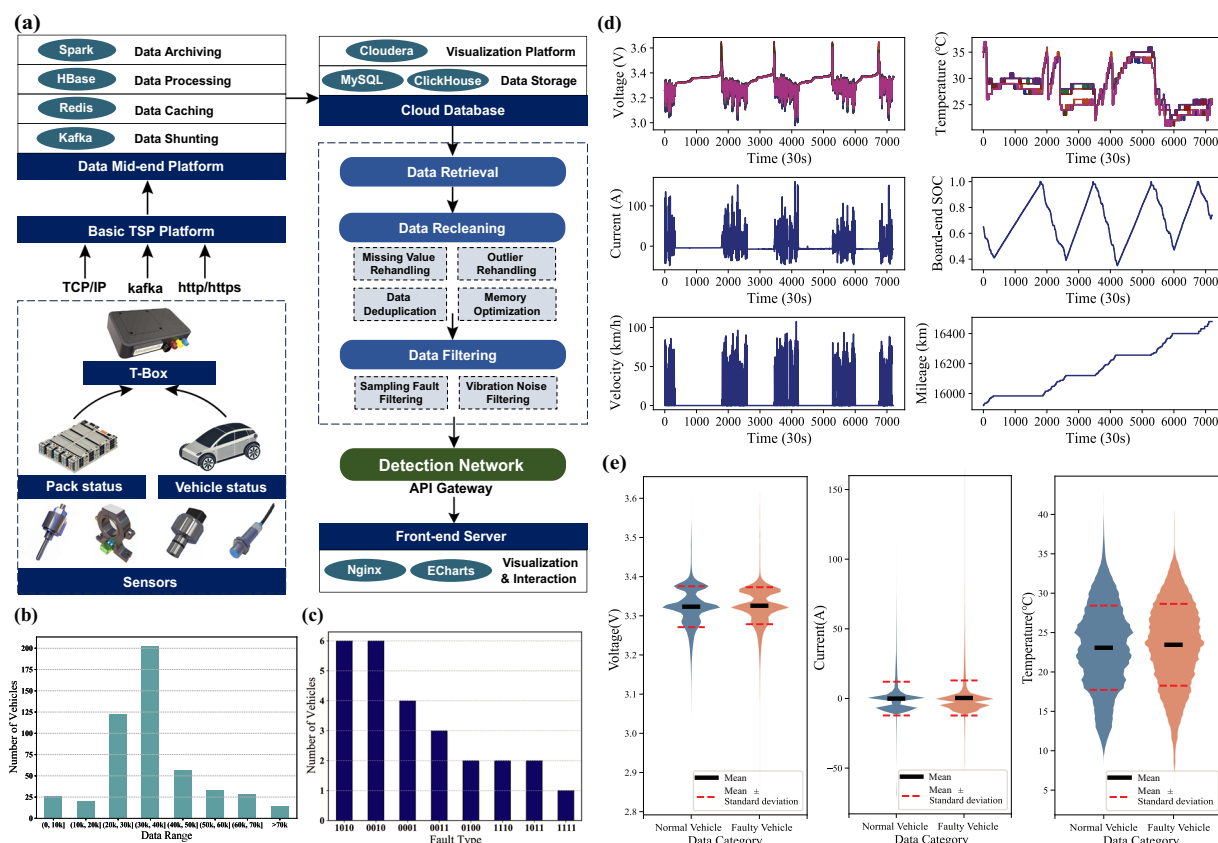


Fig. 2 | Online data acquisition and dataset analysis. **a** EV cloud data flow. Vehicle status data and battery status data are collected by multiple sensors and uploaded to the T-box. These data are then transmitted to the vehicle manufacturer's basic Telematics Service Provider (TSP) platform via Transmission Control Protocol (TCP)/Internet Protocol (IP), Kafka, and Hypertext Transfer Protocol (HTTP)/HTTPS protocols. It is subsequently sent to the data middle-end platform, where operations such as triage (Kafka), caching (Redis), pre-cleaning (Spark), and archiving (HBase) are performed. The data then flows into a cloud database for visualization (Cloudera) and storage (MySQL/Clickhouse). The proposed network retrieves the resulting data from the cloud database, outputs the results, and interacts with the front-end server through the corresponding API gateway. **b** Data frame range and vehicle number distribution. This dataset, which includes normal

and faulty samples, is derived from three manufacturers: DTI, QAS, and GIS. The graph shows the range of data frames and the distribution of the corresponding number of vehicles. **c** Statistics on the sample count of different fault types in the database. The four digits of the fault type number represent TR, EL, ISC, and EA, with 1 indicating the presence of a fault. **d** Network input sample sequences. These include voltage, temperature, and current with on-board SOC for the battery state, and speed and mileage for the vehicle state. The horizontal axis represents the number of sampled frames at 30-s intervals. **e** Battery state data distribution comparison. This comparison is between normal and faulty samples, including battery voltage, current, and temperature. The samples are taken from a random set of 100,000 data points from 10 normal/faulty vehicles. Source data are provided as a Source Data file.

Fault diagnosis

Battery degradation under real-world operating conditions takes various forms, with random triggering causes. Often, there is a certain delay in the data's ability to characterize obvious anomalies, so most safety failure samples do not reflect a clear outlier state from the data characterization. In Fig. 3a, we visualize the initial sample data of a faulty vehicle input into the network. There is no relative separation between faulty and normal sample points in the figure, making it impossible to distinguish the faulty sample points. To address these challenges, the network we design exhibits significant performance advantages. In Fig. 3b, we visualize the network's intermediate variable (the monitoring feature in Fig. 1d). The faulty sample points of the vehicle exhibit obvious clustering, creating clear spatial differences from the normal sample points. Our proposed model-constrained deep learning network identifies safety faults by outputting one-dimensional time-series comprehensive determination indicators. The time-series variation of these indicators also reflects the trend of the battery's state of health. Figure 3c–f sequentially shows the vehicle identification results for four safety fault types: EL, EA, TR, and ISC. In the fault identification process, three levels of alarm thresholds are defined based on the degree of deviation from the model

comprehensive index. When the first-level alarm threshold is triggered, the probability of fault types will be calculated for the alarm points. The t-SNE visualization plots of the case samples for the four fault types, along with the corresponding battery state data and resultant data, are visualized in Figs. S10–S13. The comprehensive determination indicators of the network outputs in the final results all show a clear rise. Additionally, by delimiting the distribution range of the indicators, we can also adjust the warning advance of the fault samples.

Performance testing

To validate the progressiveness of our proposed algorithms, we compare them with typical algorithms in terms of receiver operating characteristics (ROC), and the prediction error of pack state versus cell state on the validated dataset. Firstly, we compare the ROC performance of four popular machine learning fault diagnosis algorithms on the test samples, as shown in Fig. 4a. These include variational autoencoder (VAE), sparse autoencoder (SAE), principal component analysis (PCA), and AE (implementation details provided in ST3). The results show that existing methods often fail to meet application requirements due to a high FPR in fault detection of full-condition

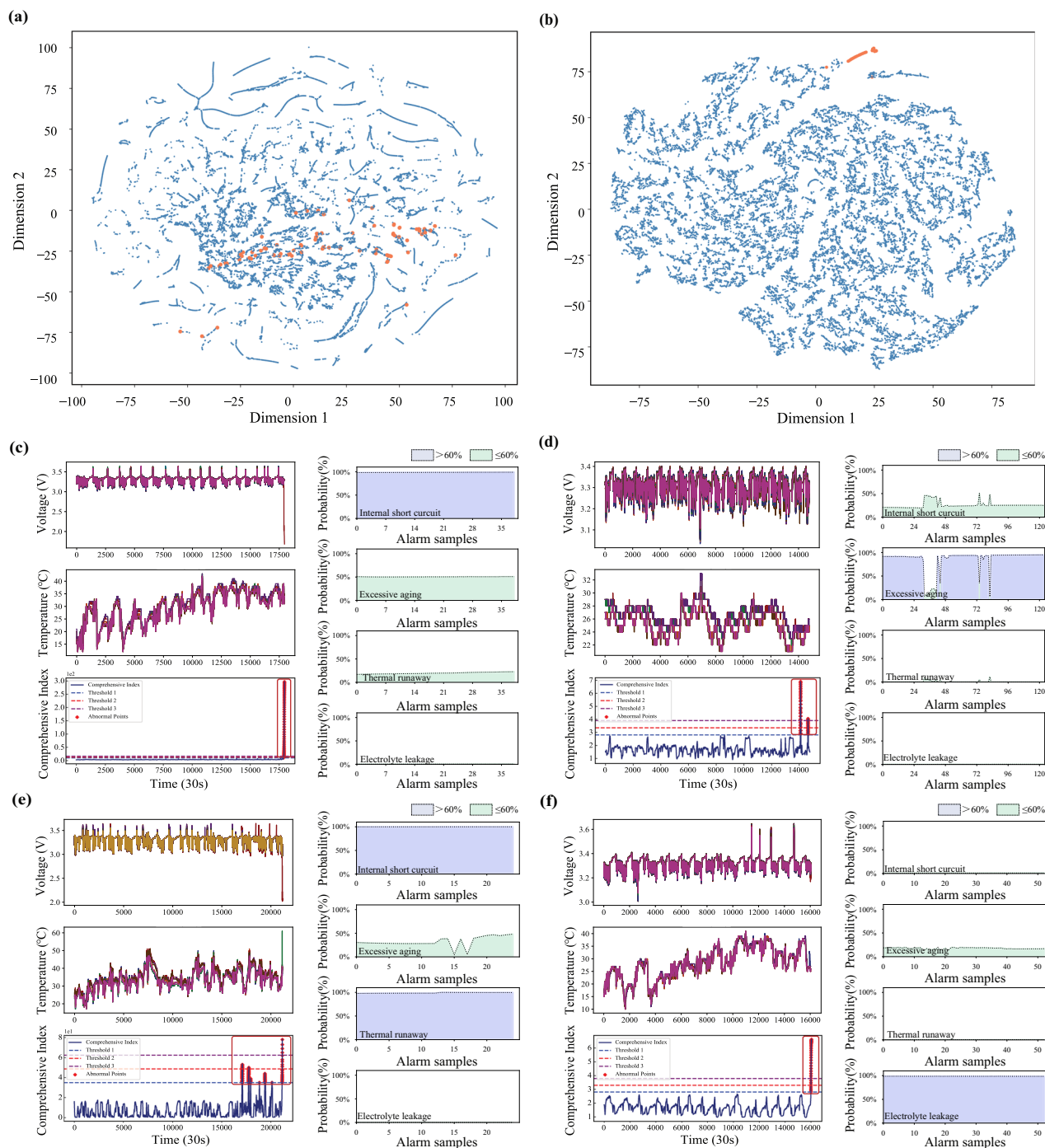


Fig. 3 | Effectiveness of safety fault detection. **a, b** present the t-SNE visualization of the inputs and outputs of the proposed network. The axes represent the two-dimensional space into which the data is compressed. Blue dots represent normal moment samples, while red dots represent fault moment samples. The left side of **c–f** display the time-series data of battery voltage and temperature variations, as well as the output results of the model comprehensive index for four safety-faulted vehicles, namely, ISC, EA, TR, and EL, respectively. The output results of threshold 1,

threshold 2, and threshold 3 correspond to sequences of 3σ , 4.5σ , and 6σ , respectively. The red point indicates the triggering threshold fault point. The four types of safety faults are recognized in advance to varying degrees. The right side of **c–f** show the real-time probability estimation results for four types of battery safety faults. Fill with blue when the probability is greater than 60%, and fill with green when the probability is less than or equal to 60%. Source data are provided as a Source Data file.

online data. In contrast, our proposed method shows a 46.5%–63.7% TPR enhancement compared to the comparison method for $FPR \in (0, 0.2]$, demonstrating significant performance advantages and stronger applicability. Second, we evaluate the prediction error of the battery pack state, as shown in Fig. 4b, comparing four typical multi-dimensional data encoder methods: VAE, SAE, and AE (specific scheme in SN5 and ST3). Due to actual data fluctuations, existing methods

cannot achieve reliable pack/cell state prediction, with their average Root Mean Square Error (RMSE) above 22.42% and the maximum RMSE error exceeding 35.28%. In contrast, our proposed method reduces the maximum error to 9.81% and the average error to less than 6.3 times, significantly improving prediction accuracy and robustness. We also discuss the cell results in the pack prediction of our proposed method by comparing the performance of four typical temporal prediction

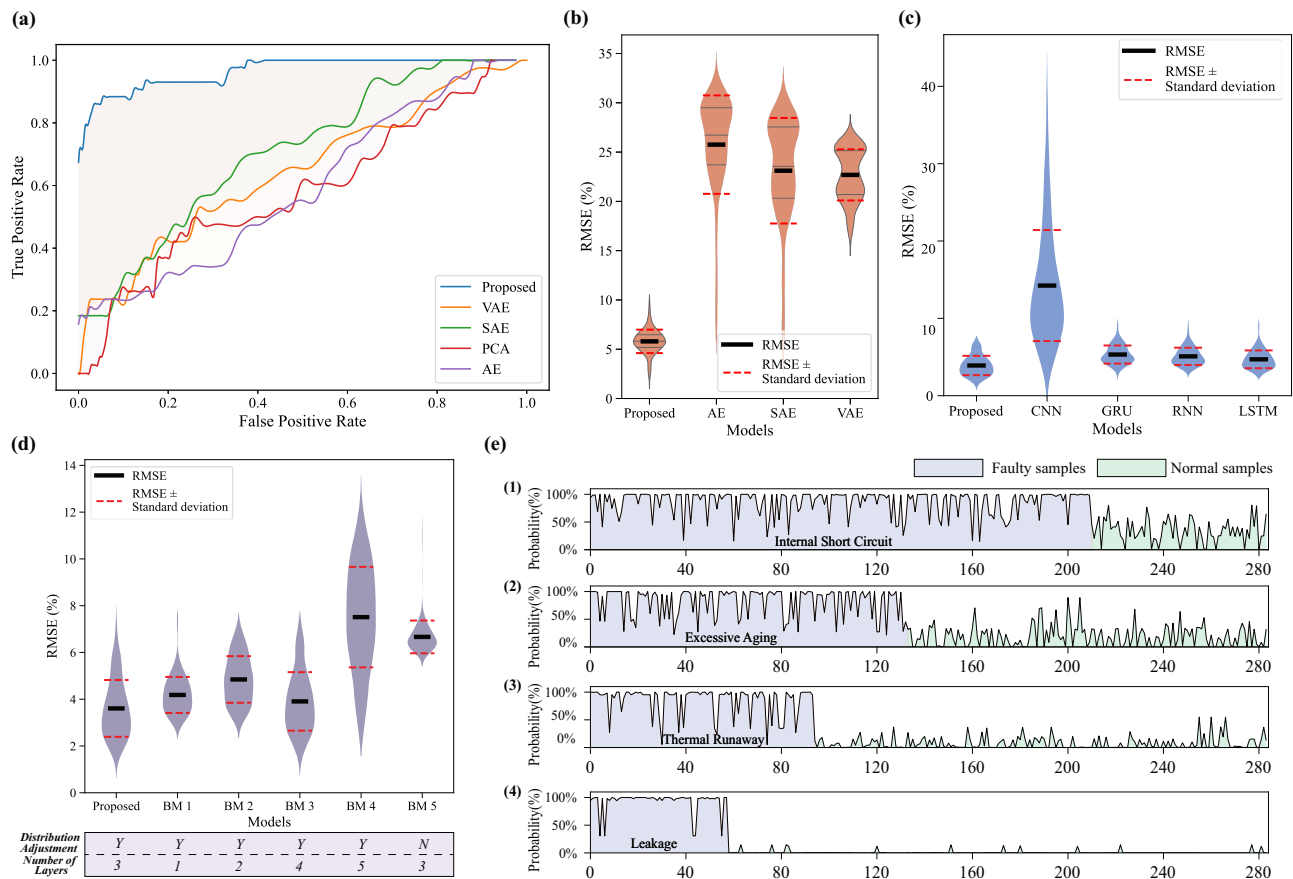


Fig. 4 | Evaluation of fault detection effectiveness and prediction accuracy.

a ROC curves of five methods on the test dataset, where the shadow represents the performance gap between methods. VAE, SAE, PCA, and AE are representative fault diagnosis methods currently in use, and the proposed method shows significant performance advantages. **b** Comparison of RMSE distribution in pack state prediction for four methods, where AE, SAE, and VAE are typical neural networks used for multidimensional reconstruction. The samples are sourced from 392 random vehicles, each providing one result. **c** Comparison of RMSE distribution in cell state prediction for five methods, where CNN, GRU, RNN, and LSTM are typical neural networks used for time-series prediction. The samples are sourced from 60 random vehicles, each providing one result. **d** Comparison of six structures in the proposed network framework, discussing the statistical correction of constraint model and

the number of layers in the BiLSTM timing network. BM1, BM2, Proposed, BM3, and BM4 perform statistical correction, and the timing network corresponds to one to five layers of BiLSTM in sequence. BM5 does not perform statistical correction, and the timing network has three layers of BiLSTM. The samples are sourced from 60 random vehicles, each providing one result. **e** Probability estimation results for four types of battery safety faults: (1), (2), (3), (4) represent the probability estimation for ISC, EA, TR, and EL faults, respectively. The x-axis represents the samples identified as triggering faults. The blue part indicates the presence of the current fault type, and the green part indicates the absence of the current fault type. The y-axis represents the probability of the fault. The comparison of probabilities between blue and green parts can verify the network's recognition performance for the four types of safety faults. Source data are provided as a Source Data file.

neural networks (Fig. 4c), including Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) (specific scheme in ST4). Without targeted training in a single state, the average RMSE (3.90%) and the maximum RMSE (6.64%) of our proposed scheme are still lower than the other schemes (the lowest average RMSE is 4.70% and the lowest maximum RMSE is 8.86%). In summary, our proposed deep learning method shows advantages in terms of pack/cell state prediction accuracy and ROC performance at the application level. This performance is attributed to the optimization of the distribution of constraint model, the network design for timing prediction, the design of the physically augmented encoder, and the overall framework of the network. To demonstrate this, we discuss the distributional optimization of constraint model and the design of the network for timing prediction in Fig. 4d, which presents the ablation results with different numbers of timing prediction network layers and inverse Gaussian distribution correction. The comparison of the RMSE of the pack state prediction shows that the optimal timing network structure with statistical correction is crucial for the performance of the algorithm (the RMSE can be optimized from 0.32% to 4.01%). It is only by combining

all the design details that the performance advantage of the current algorithm is achieved.

Fault type probability estimation

The comprehensive index can effectively present the overall potential risk, which can be regarded as the generalized index for diverse fault alerts. Based on the identified anomaly points, we perform probabilistic identification of the four most typical types (ISC, EA, TR and EL) of safety failures at the moments of anomaly. The detection results are shown in Fig. 4e. For a clearer demonstration, we present each type of fault separately. For each fault class, the left part of the figure in blue represents the fault sample, while the green means the sample without the current fault. It should be noted that all of the data sample used for the training of the detector has at least one fault, and some samples may even have three faults simultaneously. Furthermore, the occurrence probability of different fault types is different, among all data samples, the amount of ISC fault is the most, while the EL samples are the scarcest. For the most common fault, the ISC, in the dataset, most of the samples have been successfully detected, while some of the normal samples are wrongly given a higher probability. The detection

results are obviously better when it comes to the other following samples, the probability of accurately detecting the fault class is higher overall. The accuracy of the fault class detection is 80.50%, 80.83%, 89.60%, and 97.76%, for ISC, EA, TR and EL, respectively.

Discussion

The scarcity of EV fault data and the random complexity of real-world operating conditions make online battery safety monitoring a significant application challenge. Although there are many advanced fault diagnosis algorithms available, there are still two major limitations in the real-time fault diagnosis of EV batteries: the lack of performance adaptation of the algorithms in real-world stochastic scenarios, and the high input requirements of the algorithms, which hinder real-time diagnosis. These issues highlight a research gap in the field of real-time fault diagnosis for EV batteries. When dealing with these constrained online scenarios, the model-constrained deep learning method proposed in this paper plays a crucial role. In the MCNN, the integration of input-output with the loss function incorporates the state of the cell model, while the latent variable resampling process is closely aligned with the overall model's synergy. The network fully leverages the constraint relationships of the battery's physical model, enabling training and decision-making without operational limitations and with minimal data requirements. The result effectively clusters fault samples and achieves fault detection in different categories via one-dimensional indicators. The superiority of the network is validated in terms of ROC, pack state prediction error, and cell state prediction error. Compared with typical methods, the proposed method shows a 46.5–63.7% TPR improvement for FPR ∈ (0, 0.2], demonstrating significant performance advantages.

The algorithm can also adjust the output threshold to achieve hierarchical fault determination and meet different application needs, such as early warning and high recall rate. Furthermore, the occurrence probabilities of four typical safety fault types (ISC, TR, EL, EA) can be innovatively regressed from process data, highlighting the capability to identify fault types with limited data. Besides, the network can be efficiently transferred across different battery packs. This proves that the proposed network can be quickly applied to different battery packs, significantly reducing the development cycle during the application process.

Additionally, the dataset of 18.2 million items from 3 manufacturers and 515 vehicles, with 9 types of fault tags, collected by our data platform is disclosed online. This dataset will pave the way for developing larger and more powerful ML models for real-world LiB application scenarios.

The proposed method is based on the model relationship between the evolution of individual cells and the overall synergy within a battery pack. This network's application scope is not limited to battery modules but can also be adapted to similar industrial components where multiple units operate in coordination, such as generators, pumps, and other critical industrial products. In terms of functionality, under real-world constraints such as data quality and parameter quantity, it can not only be applied to battery fault detection but also be more effectively utilized in other engineering applications, such as mechanical equipment state prediction, energy system optimization, etc. This will undoubtedly promote the application of deep learning in real industrial scenarios, paving the way for broader performance optimization and safer use of industrial equipment.

Methods

Model-constrained neural network

To address the issue of missing labeled data, fault identification is performed using an unsupervised learning approach. The goal of the MCNN is to reconstruct the future state of individual cells within the monitoring system. Based on the residuals from the network's results and the information from latent variables, the steps for fault detection

and fault type probability calculation are designed and implemented sequentially, achieving fault diagnosis functionality.

Modeling and state resolution of individual and overall control models

We select two equivalent circuit control models to model the overall and individual deviations of the battery pack. During the pack modeling process, the voltage distribution often skews due to influences from different battery production batches or local cells within the circuit. We opt for the inverse Gaussian distribution to fit the voltage probability distribution⁴⁰. The distribution probability of the obtained cell voltage is as follows:

$$f(U_i|\mu, \lambda) = [\lambda/(2\pi U_i^3)]^{1/2} * \exp[-\lambda(U_i - \mu)^2/(2\mu^2 U_i)] \quad (1)$$

Where $\mu = U_m$, $\lambda = \mu^3/\text{Var}(U_i)$, U_i is the parallel module (or single) voltage, U_m is the mean value of U_i . Based on the inverse Gaussian distribution, the model is modified to obtain the pack model terminal voltage $U_{T,p}$ and the cell model terminal voltage ΔU_i , as illustrated below. where k represents the sampling time.

$$U_{T,p,k} = (f(U_{i,k}) * U_{i,k}) / \sum f(U_{i,k}) \quad (2)$$

$$\Delta U_{i,k} = U_{i,k} - U_{T,p,k} \quad (3)$$

Thus, we obtain the output of the battery pack as a whole and for each individual cell. Given the input variable current, we can construct control models for both the overall system and individual cells. On this basis, the strong tracking Kalman filter algorithm is used to solve the current state (including voltage $U_{i,k}$ and SOC $x_{i,k}$) of the model and predict the next time step⁴¹. The modeling method we adopt significantly reduces the number of parameters for cell modeling, while emphasizing the deviation comparison of different cells in the pack. We provide a detailed demonstration of the general method's specifics in SN1.

Overall state sequence prediction

By combining the battery pack and vehicle status time series data, we perform predictions for the overall state of the battery. LSTM networks are widely used in battery state prediction^{42–45}. In this part, BiLSTM is chosen as the core unit for predicting the overall state. The input data is a 7-dimensional matrix composed of total voltage U_k^{all} , current I_k , vehicle speed v_k , vehicle mileage m_k , average battery temperature T_k , board end SOC x_k^{end} and vehicle operating status x_k^{end} . This matrix is processed through a three-layer BiLSTM network to output an 18-dimensional prediction matrix p_k . The output results are dimensionally reduced through a linear layer. The prediction result at the current moment is a two-dimensional state matrix p_k^* , which serves as the input for the feature encoding part. The overall battery model state of the next frame, represented by a two-dimensional matrix consisting of voltage and SOC, is set as the p_k^* regression target. More details are shown in SN2.

Autoencoder based on model coupling relationships

The overall prediction state compensates and corrects the overall model state, and the overall and deviation model states are superimposed to map the individual cell states. The MC-AE design is based on this coupling relationship and consists of three parts: encoder, latent space sampling, and decoder⁴⁶. The process is as follows, where W represents model coefficient vector, b represents network bias, and σ represents sigmoid activation function:

- (1) The encoder integrates the overall model state with the state prediction results, which can be formulated as follows, where $\tilde{x}_k = [x_p^{pre1}, U_p^{pre1}, x_p^{pre2}, U_p^{pre2}]$, x_p^{pre1} and U_p^{pre1} is the predicted

value of SOC and voltage in the overall control model, x_p^{pre2} and U_p^{pre2} is the overall SOC and voltage prediction obtained from the sequence prediction:

$$f_k^e = W_1^e x_k^{\sim} + b_1^e \quad (4)$$

$$c_k^e = \sigma_e(W_2^e x_k^{\sim} + b_2^e) \quad (5)$$

This aims to effectively circumvent issues related to low model accuracy and poor neural network robustness, thereby achieving higher accuracy. Simultaneously, the current battery status data and vehicle status data are compressed and used as a correction term in the latent space, serving to correct weakly related physical quantities.

- (2) In the latent space sampling, we combine the encoder output latent variable with the battery deviation state to obtain the sampling points corresponding to the number of cells. The calculation process is as follows, where $\Delta s_k^{\sim} = \begin{bmatrix} \Delta x_{1,k}^{\sim} & \Delta x_{2,k}^{\sim} & \cdots & \Delta x_{n,k}^{\sim} \\ \Delta U_{1,k}^{\sim} & \Delta U_{2,k}^{\sim} & \cdots & \Delta U_{n,k}^{\sim} \end{bmatrix}_{2 \times n}$, $\Delta x_{i,k}^{\sim}$ and $\Delta U_{i,k}^{\sim}$ are the deviation prediction SOC and voltage in cell control model respectively. \mathbf{I} is an all-ones vector of the same dimension as Δs_k^{\sim} :

$$y_k^{\sim} = \Delta s_k^{\sim} + (f_k^e + c_k^e) \cdot \mathbf{I}^T \quad (6)$$

The calculation of hidden variable sampling points incorporates the logic that the single state is the sum of the pack state and the deviation state. These sampling points will be utilized by the decoder to generate new data points.

- (3) The decoder's task is to map the sample points from the latent space to the data space of the system's future state. The design of the decompression network is as follows:

$$y_k^* = \sigma_d(W_1^d y_k^{\sim} + b_1^d) \quad (7)$$

In this process, σ_d is an activation function designed based on battery operating conditions to prevent gradient explosion during training.

Diagnostic output

We carry out feature extraction on the single prediction state output by MC-AE. Taking $r_k^* = [y_k^* - s_k^*]$ as input, the extracted feature matrix is $[ewu, ewx, zu_1, zx_1, zu_2, zx_2]$. Where $s_k^* = \begin{bmatrix} x_{1,k} & x_{2,k} & \cdots & x_{n,k} \\ U_{1,k} & U_{2,k} & \cdots & U_{n,k} \end{bmatrix}_{2 \times n}$.

The calculating process of each dimension feature is as follows:

$$zu_1(k) = \max[(u_k^* - \mu_k^u)/\sigma_k^u] \quad (8)$$

$$zx_1(k) = \max[(x_k^* - \mu_k^x)/\sigma_k^x] \quad (9)$$

$$zu_2(k) = (u_{\max,k}^* - u_{\sec\max,k}^*)/\sigma_k^u \quad (10)$$

$$zx_2(k) = (x_{\max,k}^* - x_{\sec\max,k}^*)/\sigma_k^x \quad (11)$$

Where, $u_{\max,k}^*$ and $x_{\max,k}^*$ are the maximum values of $r_k^*[1, :]$ and $r_k^*[2, :]$ respectively, while $u_{\sec\max,k}^*$ and $x_{\sec\max,k}^*$ are the second largest values of $r_k^*[1, :]$ and $r_k^*[2, :]$ respectively. ewu and ewx are the EWMA signals of $u_{\max,k}^*$ and $x_{\max,k}^*$ respectively⁴⁷. For directional dimensionality reduction of features, we refer to the principal component analysis method. We select feature vectors that contain 95% of the eigenvalue proportions to form a new feature space. The Hotelling's T^2 statistics and SPE statistics of the feature data mapping results are used to obtain the final comprehensive evaluation⁴⁸.

We quantify the normal/abnormal degree of the sample through a one-dimensional comprehensive evaluation index and can output the grading results of abnormal monitoring. The calculation process of indicators is shown in SN3.

Subsequently, we extracted the intermediate parameters in the statistical index calculation process to form a supplementary classification feature matrix Θ (More details in SN4). To facilitate the handling of emergencies, considering the different levels of urgency and troubleshooting methods, we construct a fault class detector that can output the probability of the fault class. Because of the informative feature vector of the decoder layer, the detector contains just one linear layer with 10 trainable weights and biases for each fault type, by which the detector can separately estimate each fault type probability.

Model training

The proposed framework is trained in three modular networks, corresponding to the sections in Fig. 1: (b) sequence network correction, (c) MC-AE, and (d) diagnostic output. The training dataset used in (b) and (c) consists of 200,000 normal vehicle operating data points with no anomalies. The sample distribution of the training set is shown in SF2. During the training process for (b), we set the batch size to 100 and select the Adam algorithm as the training optimizer. In the sequential network, we set the learning rate to $1e-4$. The training input data is normalized by $x_n = \frac{x - \mu}{\sigma}$, where x_n and x represent the physical parameters before and after normalization, and μ and σ represent the mean and standard deviation of the parameters in the training dataset. The loss function is the mean squared error, which is shown as follows:

$$loss_1 = 1/t \sum_{k=1}^t (p_k^* - u_k^*)^2 \quad (12)$$

In (12), $u_k^* = [U_{T,p,k+1}, x_{T,p,k+1}]$ and p_k^* represent output, where $U_{T,p,k+1}$ is the terminal voltage at time $k+1$ of the pack model, and $x_{T,p,k+1}$ is the pack model SOC at time $k+1$. During the training of (d) part, the learning rate is set to $5e-4$, and the network training loss uses the MSE function shown as follows:

$$loss_2 = 1/t \sum_{k=1}^t (y_k^* - s_k^*)^2 \quad (13)$$

In (13), y_k^* represent network output. After training, we use the data from all 515 vehicles in the dataset as the test set to validate the performance of the MCNN.

For the training of the fault type probability estimation network in section (d), the training set is composed of feature samples from abnormal moments with known fault type labels. The fault label consists of four digits, with each digit representing a type of fault corresponding to separate trainable linear layer. The dataset consists of 1135 samples with ground truth labels. We split the dataset into training and testing sets with a 3:1 ratio, resulting in 851 samples for training and 285 samples for testing. The final learning rate is set to $1e-4$, and the Binary Cross Entropy Loss (BCE) is adopted loss function. All training processes of the model are completed on an NVIDIA GeForce RTX 3090.

Transfer learning

During the application process, battery cells of the same type are often integrated into power cells with different pack structures. In such cases, reselecting the training set and retraining can be costly and may lack precision. To address this, we discuss the network transferring process under different pack structures to significantly shorten the model development cycle. During the process, the module state prediction and fault type probability estimation parts are not affected by the number of cells and can be generalized to

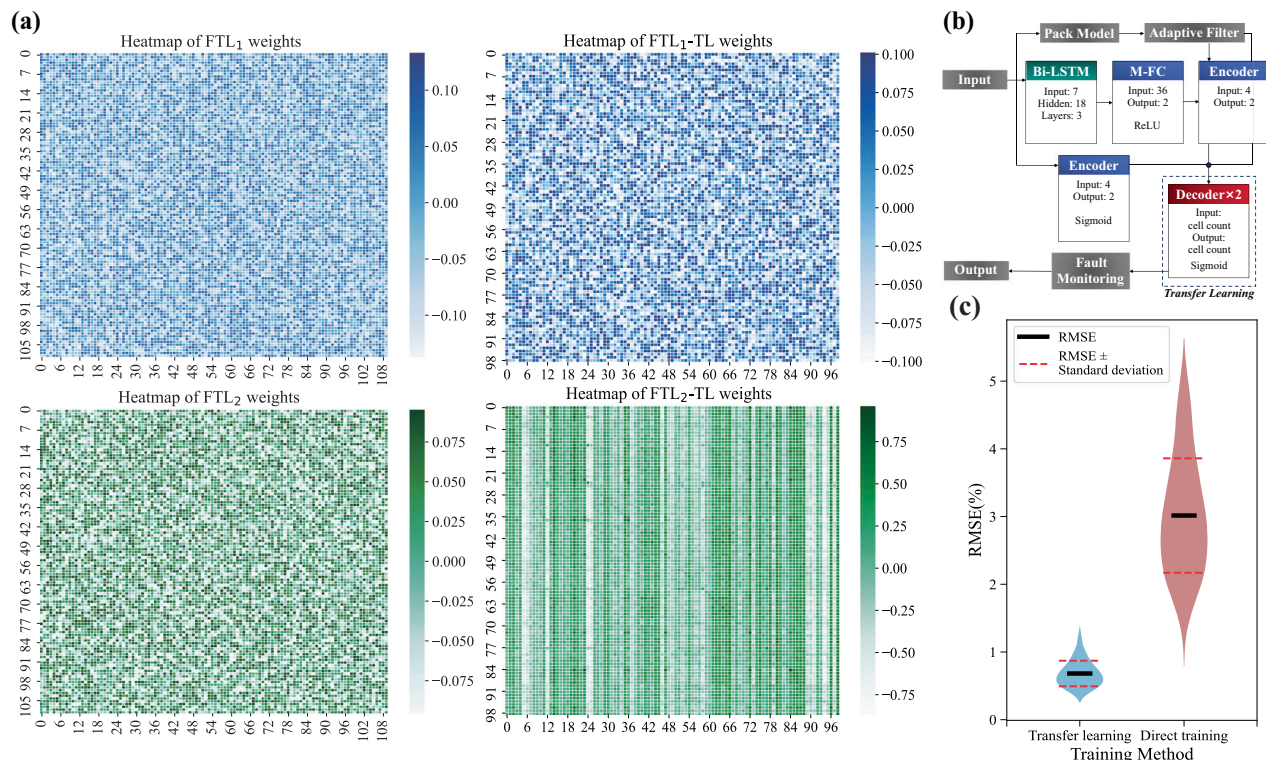


Fig. 5 | Discussion on network transferring between different packs. a The weight distribution of the FTLs before and after transferring. The pack dimension of the pre-trained model is 110, while the dimension of the transferred pack is 100. **b** The architecture of the developed network. The dashed lines indicate

components eligible for network transfer. **c** Comparison of the RMSE error distribution for pack status prediction between model transfer and direct training, both with the same 50 training cycles. The samples are sourced from 60 vehicles, each providing one result. Source data are provided as a Source Data file.

modules with different structures. For the MC-AE part, we replace the feature conversion layer of the pre-trained network's model-constrained encoder with a new fully connected layer corresponding to the pack dimension. When implementing the network, only the new FTL layer is trained. We experiment with transferring a model trained on a module with 110 cells to a module with 99 cells. Using the network trained on the 110-cell module as the pre-trained model, we train for an additional 50 epochs and achieve the desired results. The weight distribution of the FTLs before and after transferring is shown in Fig. 5a. The architecture of the network and the position of the transferred layer is shown in Fig. 5b. By comparing the pack state prediction performance before and after transferring, it is evident that the accuracy of transferred network accuracy has significant advantages in terms of accuracy and robustness compared to direct training, as shown in Fig. 5c.

Data availability

The processed EV data in this study have been deposited in the Zenodo database under accession code⁴⁹. Source data are provided with this paper.

Code availability

The main code are available in the Zenodo database⁵⁰.

References

- Harper, G. et al. Recycling lithium-ion batteries from electric vehicles. *Nature* **575**, 75–86 (2019).
- Liu, K., Liu, Y., Lin, D., Pei, A. & Cui, Y. Materials for lithium-ion battery safety. *Sci. Adv.* **4**, eaas9820 (2018).
- Grey, C. P. & Hall, D. S. Prospects for lithium-ion batteries and beyond—a 2030 vision. *Nat. Commun.* **11**, 2–5 (2020).
- Tao, Y., Rahn, C. D., Archer, L. A. & You, F. Second life and recycling: energy and environmental sustainability perspectives for high-performance lithium-ion batteries. *Sci. Adv.* **7**, 1–16 (2021).
- Shaffer, B., Auffhammer, M. & Samaras, C. Make electric vehicles lighter to maximize climate and safety benefits. *Nature* **598**, 254–256 (2021).
- Che, Y., Hu, X., Lin, X., Guo, J. & Teodorescu, R. Health prognostics for lithium-ion batteries: mechanisms, methods, and prospects. *Energy Environ. Sci.* **16**, 338–371 (2023).
- Zhang, Y. et al. A smart risk-responding polymer membrane for safer batteries. *Sci. Adv.* **9**, 1–9 (2023).
- Hu, X., Xu, L., Lin, X. & Pecht, M. Battery lifetime prognostics. *Joule* **4**, 310–346 (2020).
- Castelvecchi, D. Electric cars and batteries: how will the world produce enough? *Nature* **596**, 336–339 (2021).
- Sulzer, V. et al. The challenge and opportunity of battery lifetime prediction from field data. *Joule* **5**, 1934–1955 (2021).
- Li, W., Zhu, J., Xia, Y., Gorji, M. B. & Wierzbicki, T. Data-driven safety envelope of lithium-ion batteries for electric vehicles. *Joule* **3**, 2703–2715 (2019).
- Severson, K. A. et al. Capacity degradation. *Nat. Energy* **4**, 383–391 (2019).
- Han, X. et al. A review on the key issues of the lithium ion battery degradation among the whole life cycle. *eTransportation* **1**, 100005 (2019).
- Liu, X. et al. Thermal runaway of lithium-ion batteries without internal short circuit. *Joule* **2**, 2047–2064 (2018).
- Naguib, M. et al. Limiting internal short-circuit damage by electrode partition for impact-tolerant Li-ion batteries. *Joule* **2**, 155–167 (2018).
- Feng, X. et al. Thermal runaway mechanism of lithium ion battery for electric vehicles: a review. *Energy Storage Mater.* **10**, 246–267 (2018).

17. Roman, D., Saxena, S., Robu, V., Pecht, M. & Flynn, D. Machine learning pipeline for battery state-of-health estimation. *Nat. Mach. Intell.* **3**, 447–456 (2021).
18. Chen, B. R. et al. Battery aging mode identification across NMC compositions and designs using machine learning. *Joule* **6**, 2776–2793 (2022).
19. Kang, Y., Duan, B., Zhou, Z., Shang, Y. & Zhang, C. Online multi-fault detection and diagnosis for battery packs in electric vehicles. *Appl. Energy* **259**, 114170 (2020).
20. Xiong, R., Yang, R., Chen, Z., Shen, W. & Sun, F. Online fault diagnosis of external short circuit for lithium-ion battery pack. *IEEE Trans. Ind. Electron.* **67**, 1081–1091 (2020).
21. Tian, J. P., Xiong, R., Shen, W. X. & Sun, F. C. Fractional order battery modelling methodologies for electric vehicle applications: recent advances and perspectives. *Sci. China Technol. Sci.* **63**, 2211–2230 (2020).
22. Lu, Y. et al. Ultrasensitive detection of electrolyte leakage from lithium-ion batteries by ionically conductive metal-organic frameworks. *Matter* **3**, 904–919 (2020).
23. Lu, J., Xiong, R., Tian, J., Wang, C. & Sun, F. Deep learning to estimate lithium-ion battery state of health without additional degradation experiments. *Nat. Commun.* **14**, 1–13 (2023).
24. Ren, D. et al. Model-based thermal runaway prediction of lithium-ion batteries from kinetics analysis of cell components. *Appl. Energy* **228**, 633–644 (2018).
25. Yang, R., Xiong, R., Ma, S. & Lin, X. Characterization of external short circuit faults in electric vehicle Li-ion battery packs and prediction using artificial neural networks. *Appl. Energy* **260**, 114253 (2020).
26. Feng, X., Ren, D., He, X. & Ouyang, M. Mitigating thermal runaway of lithium-ion batteries. *Joule* **4**, 743–770 (2020).
27. Fei, Z., Huang, Z. & Zhang, X. Voltage and temperature information ensembled probabilistic battery health evaluation via deep Gaussian mixture density network. *J. Energy Storage* **73**, 108587 (2023).
28. Hu, X. et al. Advanced fault diagnosis for lithium-ion battery systems: a review of fault mechanisms, fault features, and diagnosis procedures. *IEEE Ind. Electron. Mag.* **14**, 65–91 (2020).
29. Ma, G., Xu, S. & Cheng, C. Fault detection of lithium-ion battery packs with a graph-based method. *J. Energy Storage* **43**, 103209 (2021).
30. Xiong, R., Sun, W., Yu, Q. & Sun, F. Research progress, challenges and prospects of fault diagnosis on battery system of electric vehicles. *Appl. Energy* **279**, 115855 (2020).
31. Zhang, Y. et al. A multi-level early warning strategy for the LiFePO₄ battery thermal runaway induced by overcharge. *Appl. Energy* **347**, 121375 (2023).
32. Jin, H. et al. A combined model-based and data-driven fault diagnosis scheme for lithium-ion batteries. *IEEE Trans. Ind. Electron.* **PP**, 1–11 (2023).
33. Yu, Q., Wang, C., Li, J., Xiong, R. & Pecht, M. Challenges and outlook for lithium-ion battery fault diagnosis methods from the laboratory to real world applications. *eTransportation* **17**, 100254 (2023).
34. Xu, C., Li, L., Xu, Y., Han, X. & Zheng, Y. A vehicle-cloud collaborative method for multi-type fault diagnosis of lithium-ion batteries. *eTransportation* **12**, 100172 (2022).
35. Yang, R., Xiong, R., Shen, W. & Lin, X. Extreme learning machine-based thermal model for lithium-ion batteries of electric vehicles under external short circuit. *Engineering* **7**, 395–405 (2021).
36. Zhao, J., Ling, H., Wang, J., Burke, A. F. & Lian, Y. Data-driven prediction of battery failure for electric vehicles. *iScience* **25**, 104172 (2022).
37. Jiang, L. et al. Data-driven fault diagnosis and thermal runaway warning for battery packs using real-world vehicle data. *Energy* **234**, 121266 (2021).
38. Zhao, J. et al. Battery fault diagnosis and failure prognosis for electric vehicles using spatio-temporal transformer networks. *Appl. Energy* **352**, 121949 (2023).
39. Zhang, J. et al. Realistic fault detection of Li-ion battery via dynamical deep learning. *Nat. Commun.* **14**, 5940 (2023).
40. Barndorff-Nielsen, O. E. Normal inverse Gaussian distributions and stochastic volatility modelling. *Scand. J. Stat.* **24**, 1–13 (1997).
41. Jwo, D. J. & Wang, S. H. Adaptive fuzzy strong tracking extended Kalman filtering for GPS navigation. *IEEE Sens. J.* **7**, 778–789 (2007).
42. Yun, S. T. & Kong, S. H. Data-driven in-orbit current and voltage prediction using Bi-LSTM for LEO satellite lithium-ion battery SOC estimation. *IEEE Trans. Aerosp. Electron. Syst.* **58**, 5292–5306 (2022).
43. Chemali, E., Kollmeyer, P. J., Preindl, M., Ahmed, R. & Emadi, A. Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries. *IEEE Trans. Ind. Electron.* **65**, 6730–6739 (2018).
44. Ren, L. et al. A data-driven Auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life. *IEEE Trans. Ind. Inform.* **17**, 3478–3487 (2021).
45. Zraïbi, B., Okar, C., Chaoui, H. & Mansouri, M. Remaining useful life assessment for lithium-ion batteries using CNN-LSTM-DNN hybrid method. *IEEE Trans. Veh. Technol.* **70**, 4252–4261 (2021).
46. Phan, N. H., Wang, Y., Wu, X. & Dou, D. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In *30th AAAI Conference on Artificial Intelligence 2016* 1309–1316 (AAAI Press, 2016). <https://doi.org/10.1609/aaai.v30i1.10165>.
47. Lu, C. W. & Reynolds, M. R. EWMA control charts for monitoring the mean of autocorrelated processes. *J. Qual. Technol.* **31**, 166–188 (1999).
48. Sun, X., Marquez, H. J., Chen, T. & Riaz, M. An improved PCA method with application to boiler leak detection. *ISA Trans.* **44**, 379–397 (2005).
49. Cao, R. et al. Model-constrained deep learning for online fault diagnosis in Li-ion batteries over stochastic conditions. <https://doi.org/10.5281/zenodo.10656500> (2024).
50. Cao, R. et al. Model-constrained deep learning for online fault diagnosis in Li-ion batteries over stochastic conditions. Model-Constrained-Deep-Learning-for-Online-Fault-Diagnosis. <https://doi.org/10.5281/zenodo.14555916> (2024).

Acknowledgements

This work has been supported in part by the National Key R&D Program of China under Grant 2021YFB2501300 and 2023YFB2504100 (S.Y.), National Natural Science Foundation of China Regional Innovation and Development Joint Fund under Grant U22A2042 (X.L.), and the Fundamental Research Funds for the Central Universities under Grant YWF-23-JC-08 (S.Y.).

Author contributions

R.C. conceived the idea of online fault diagnosis, led the project, participated in algorithm development paper writing and revision. S.Y. supervised and led this project. Z.Z., R.S., and Y.Z. generated the data and performed the computational studies. J.L., X.L., and Y.S. conceived, wrote, and revised the manuscript. All the authors have revised the manuscript and agreed with its content.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-56832-8>.

Correspondence and requests for materials should be addressed to Shichun Yang.

Peer review information *Nature Communications* thanks Jingyuan Zhao and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025