Energy Advances

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Energy Adv.*, 2025, 4, 1295

A sampling fault diagnosis method for power battery data in cloud platforms based on a ResNet–BiLSTM neural network

Yuntao Jin, Zhengjie Zhang,* Baitong Chang, Rui Cao, Hanqing Yu, Yefan Sun, Xinhua Liu and Shichun Yang

As the basis for many functions of the battery management system (BMS) such as state estimation and thermal runaway warning, stable sampling data are crucial for the safe operation of electric vehicles (EVs). In this paper, a sampling fault diagnosis method for power battery data in cloud platforms is proposed based on a residual network (ResNet) and bi-directional long short-term memory (BiLSTM) neural network, which can effectively identify the abnormalities of the battery sampling data and recognize the failure modes. Firstly, through the analysis of fault data and sampling circuits for real EVs, four typical failure modes are selected to complete the fault injection experiments. The physical simulation model of the fault circuit is established, and the corresponding mathematical empirical model is condensed. Then, based on the understanding of the abnormal data distribution pattern, the fault diagnosis algorithms based on a threshold and the ResNet–BiLSTM neural network are developed, respectively. Finally, the algorithms are introduced into the simulation dataset and real-vehicle dataset for testing. The results show that both algorithms have high effectiveness and accuracy, with the latter exhibiting strong fault diagnosis capability. In summary, the proposed sampling fault diagnosis method is feasible and provides a theoretical basis for future multi-type fault diagnosis of BMSs.

Received 5th April 2025,
Accepted 11th August 2025

DOI: 10.1039/d5ya00093a

rsc.li/energy-advances

1. Introduction

Due to global warming and the continuous consumption of fossil fuels, electric vehicles (EVs) have become an ideal means of transportation. As the key component of the EV power source, the performance of the power battery is affected by the environment and evolves over time; thus, refined battery management systems (BMSs) have become a research hotspot.[1] Traditional vehicle-side BMSs are limited by computing and storage capacities and cannot run complex artificial intelligence (AI) models to analyze the full lifespan data, while cloud-based big data platforms have the advantages of easy expansion, strong stability and integration of storage and computing. Therefore, end-cloud collaboration is also considered to be the future development trend of BMS technology.[2]

The cloud platform of power battery data normally accomplishes functions such as the state of health (SoH) estimation, fault diagnosis and remaining useful life (RUL) prediction, based on information uploaded from the vehicle, which require accurate and real-time battery data as support.[3] Among them, in terms of the fault diagnosis of power battery systems, current main research focuses on the fault types that occur for the battery itself, such as internal short circuit, overcharge, over-discharge and capacity loss.[4] Li *et al.* developed a thermal runaway warning algorithm for abnormal heat production using vehicle state, driving behavior and local weather as inputs.[5] Jiang *et al.* used a state representation methodology (SRM) for battery fault diagnosis based on original cell voltages, which captured subtle changes in the battery and enabled rapid fault identification.[6] Zhang *et al.* built a deep learning fault identification model based on a dynamical autoencoder using voltage, current, temperature, state of charge and other signals as inputs, which comprised over 690 000 charging snippets from 347 vehicles' battery packs.[7] These studies have used popular data-driven techniques as methodologies, while defaulting to the authenticity and reliability of the data collected by sensors.[8]

However, in actual use, various factors may cause data to be missing, damaged and misplaced such as vehicle/cloud-end communication interruption, sampling signal failure and cloud platform component abnormality.[9] Due to the long-term operation and severe working environment, sensor faults are inevitable within the lifespan of a BMS, even though the probability of fault happening is on the order of one part per million.[10]

School of Transportation Science and Engineering, Beihang University, Beijing 102206, China. E-mail: zhengjie_zhang@buaa.edu.cn, yt_jin@buaa.edu.cn, changbaitong@buaa.edu.cn, crcaorui@buaa.edu.cn, hanqingyu@buaa.edu.cn, zy2213111@buaa.edu.cn, liuxinhua19@buaa.edu.cn, yangshichun@buaa.edu.cn

Therefore, it is essential to have a better understanding of failure mechanisms of power electronic components and to explore innovative approaches to increase the reliability of power electronic circuits and systems.[11] Cloud platforms can usually interpolate and infill or directly reject the missing and damaged data. However, misaligned data are often difficult to directly distinguish from normal data, which can easily lead to false alarms and distorted state estimation.[12] In the field of engineering, identifying signal errors caused by sampling faults has become a critically needed function for cloud-side data platforms. Currently, sensor fault diagnosis research is divided into three stages: detection, identification and quantification. All three stages are based on the establishment of a high-fidelity model of the battery sampling circuit to explore the failure mechanism of the system.[13,14] Meanwhile, in order to address the issue of fewer samples caused by the small probability of the failure event itself, researchers have already generated customized failure sample datasets by establishing physical simulation models or data-driven models of the research object. It reached a balance between the number of positive and negative samples and realized the development of diagnosis algorithms.[15,16]

Throughout the lifespan of EVs, onboard sampling circuit failures caused by bumpy driving, humid environments or aging devices are extremely common, with specific failure modes such as short or break circuits in wiring harnesses, components or connectors. The most serious scenario such as the direct breakdown of the sampling chip usually demonstrates that the data cannot be collected on the vehicle or in the cloud, and the failure would be directly detected by the system at that time. It is out of the scope of research in this paper. This paper mainly focuses on the abnormalities of sampling circuits, aiming to reduce the impact on the BMS functions caused by data acquisition errors. In Section 2 of this paper, different fault injection simulation experiments are performed according to a sampling circuit board of real-vehicle BMSs. The corresponding failure physical models are established using MATLAB/Simscape software, and the mathematical empirical models for different failure modes are summarized. The experiments and models provide a theoretical basis for the subsequent development of sampling fault diagnosis algorithms. In Section 3, the threshold algorithm for identifying anomaly patterns of data and the deep learning algorithm for detecting anomalies in temporal data are developed, respectively, and the algorithms are tested and validated in a 30-vehicle-scale dataset for real EVs. Section 4 presents the results and discussion, and the conclusion is summarized in Section 5.

2. Experiments and models for sampling fault injection

BMSs mainly monitor the voltage, current, temperature and other signals of the battery. It can accomplish the functions such as state estimation, charging/discharging control and fault warning, improving the safety and life performance of the battery system. The hardware of BMSs is mainly composed of a main board, sampling board, battery disconnect unit (BDU) and high voltage control board. Among them, the main board is responsible for collecting the sampling data from the sampling board and communicating with the vehicle through the low-voltage electrical interface to achieve the energy management and control of the battery system. The sampling board is responsible for monitoring the sensor's information such as voltage, current and temperature, which are transmitted to the main board. Advanced sampling boards are also equipped with the function of battery equalization. As shown in Fig. 1, the green line represents the flow of data transmission, the battery data are collected and processed by the BMS, and then transmitted to the cloud-based big data platform through the T-Box on the vehicle-end in real time. The fault recognition algorithm to be studied in this paper is deployed on the cloud-end, and the goal of the algorithm is to online monitor for the existence of similar sampling anomalies and issue an early warning.

The research object of this paper is a sampling board equipped with a LTC6811 sampling chip, which is also widely used onboard currently. It can simultaneously measure the voltage of up to 12 cells, with a measurement range of 0–5.5 V

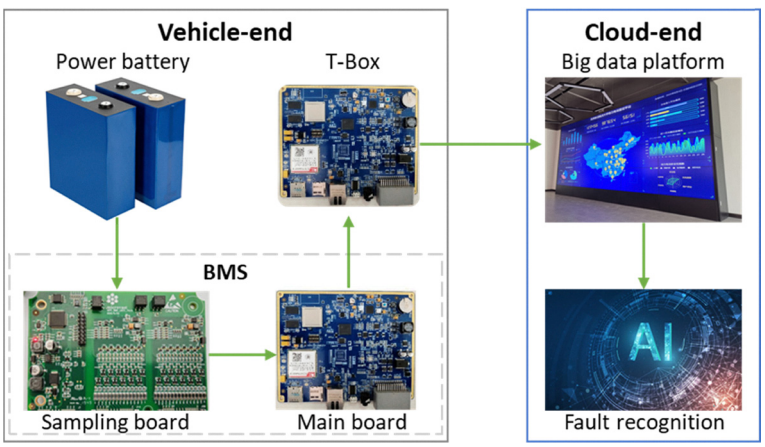

Fig. 1 Vehicle-end to cloud-end battery data transfer flow.

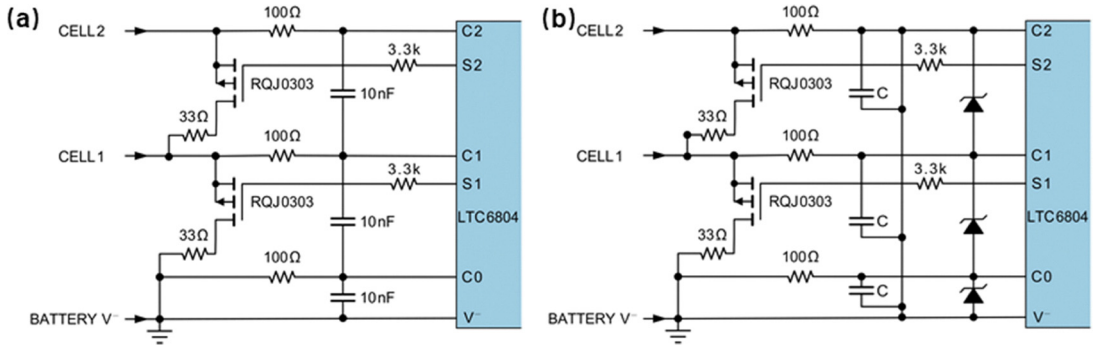

Fig. 2 Sampling board hardware circuit topology: (a) differential filter circuit and (b) ground filter circuit.

per channel and an error of less than 1.2 mV. When the sampling chip is in operation, the voltage difference between each branch (*i.e.*, C0, C1 and C2) and the reference ground, as shown in Fig. 2(a), is equal to $U_{\text{sample},i}$ in order of distance from the reference ground from near to far. The difference calculation of adjacent measurements can be obtained as the cell sampling voltage U_i as in eqn (1). In addition, $U_{\text{bat},i}$ is used as the true terminal voltage value for each cell in this paper. With normal sampling circuits, the terminal voltage of the battery $U_{\text{bat},i}$ should be equal to the sampling voltage U_i. When a fault occurs, the sampling voltage would be offset from the true battery voltage.

$$U_i = U_{\text{sample},i} - U_{\text{sample},i-1}, \quad (i = 1, 2, 3, \ldots, 12) \qquad (1)$$

The sampling fault studied in this paper is mainly related to the sampling board, and the present hardware topology of the sampling board mainly has two kinds of ground filtering and differential filtering, as shown in Fig. 2. The ground filtering circuit connects all the filter capacitor branches of the cell to ground, which reduces the crosstalk of different branch currents in the differential filtering circuit and increases the voltage regulator diode in each branch to avoid the energy impact when large current passes through. In practice, the ground filtering is widely used because it can achieve better voltage ripple suppression, even if the cost is higher.

This section mainly focuses on the more complex ground filtering circuit to carry out fault injection experiments. The experimental setup for simulating sampling faults is depicted in Fig. 3, encompassing many components such as the constant voltage power source (utilized for simulating battery inputs), the sampling host computer and the BMS sampling board. When employing the battery module for experimental purposes, there is a potential risk to induce short circuits, leading to fire or explosion. Therefore, this study initially employs a constant voltage power source equipped with a safety circuit for the simulation. Subsequently, utilizing a calibrated fault model, this paper investigates the behavior of abnormal data under real vehicle conditions. The experimental setup uses a sampling board equipped with two sampling chips capable of simultaneously collecting 24 channels of cell voltage data. In this study, we specifically focus on conducting fault injection experiments on 12 selected channels of voltage associated with one of the chips, while the remaining 12 channels of grounding are not subjected to investigation.

Based on the statistics and analysis of real-vehicle faults and different device failure modes, the injection and evaluation of four representative fault types, namely, sampling harness breakage, equalization loop closure, filter capacitor breakdown and voltage regulator diode breakdown, were completed, respectively. Furthermore, a simplified physical model was constructed according to the topology of the circuit using MATLAB/Simscape software, and the circuit analysis and modeling part of this paper can be found in the Appendix. To streamline computations in the model, only the first six cells' voltages that were affected by faults were retained. Building upon this foundation, a substantial dataset comprising both fault and normal scenarios was generated through simulations. This dataset serves as a valuable resource for training and validating deep learning algorithms of anomaly detection. In the following, we outline the simulation experiments of each type of fault injection and elucidate the method used for establishing the corresponding fault modes. The fault injection locations were strategically chosen, focusing on the middle cells of the circuit or selecting the first and last cells.

2.1 Sampling harness breakage experiment

Sampling harness breakage is the most common type of hardware failure, which can be induced by failure modes (*e.g.*, harness disconnection and connector malfunction) and erroneous welding practices (*e.g.*, false welding and void welding). In this paper, the experimental injection of faults for sampling harness breakage is depicted in Fig. 4(a) at the marked location, and in subplots (a) of Fig. 4–7, the fault injection positions are illustrated using the scenario in which the fault occurs in the middle cell as an example. The fault simulation can be achieved by removing the chip ferrite bead situated on the second branch of the sampling circuit. It is noteworthy that the chip ferrite bead serves the purpose of suppressing high-frequency noise and spikes on both signals and power lines, functioning to absorb electrostatic pulses. The experimental results are shown in Fig. 4(c). It is evident that, after disconnecting the sampling harness, the voltage suspension of faulty branch results in the equivalent of an unstable voltage source

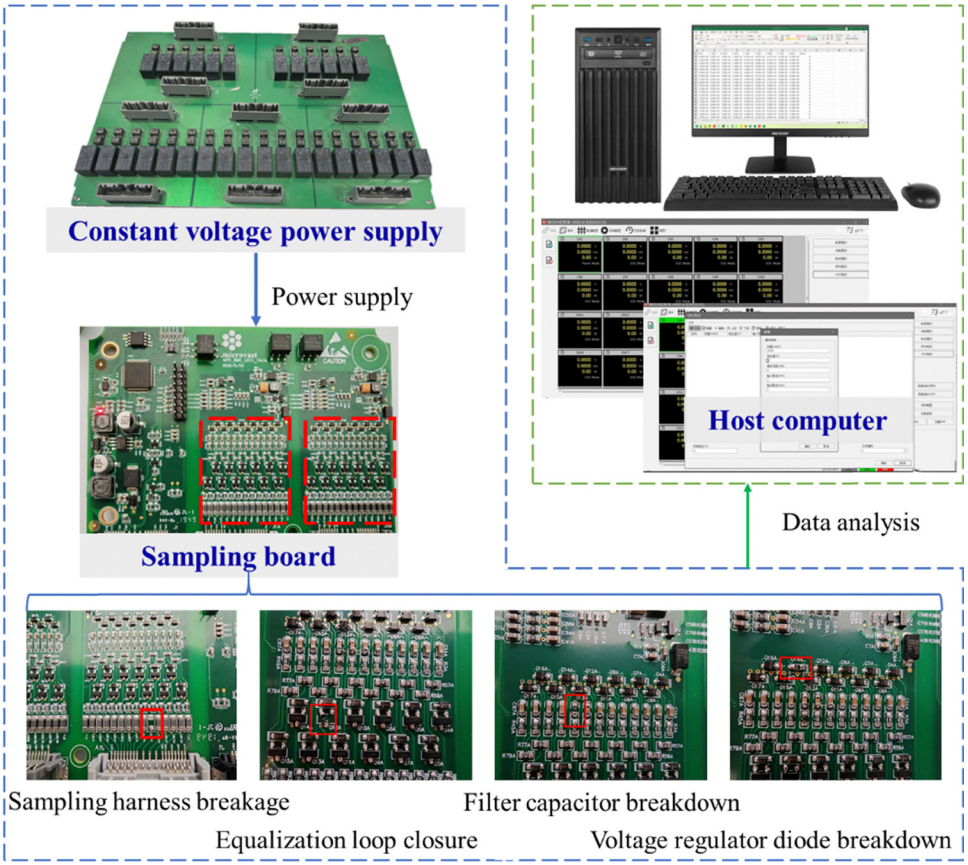

Fig. 3 Test bench of sampling fault injection.

connected in series with the branch, leading to an increase in the measured voltage. During this period, the measured voltages across the first three branches are illustrated in eqn (2):

$$\begin{cases} U_{\text{sample},1} = U_{\text{bat},1} \\ U_{\text{sample},2} = U_{\text{bat},1} + U_{\text{bat},2} + U_{\text{overhang}} \\ U_{\text{sample},3} = U_{\text{bat},1} + U_{\text{bat},2} + U_{\text{bat},3} \end{cases} \quad (2)$$

where $U_{\text{sample},i}$ is the measured voltage of the ith branch, $U_{\text{bat},i}$ is the true terminal voltage of the ith cell, and U_{overhang} is the overhang voltage at the open circuit. The sampled voltage value of the corresponding cell is determined by the difference between the voltages of two adjacent circuits, as shown in the following equation.

$$\begin{cases} U_1 = U_{\text{bat},1} \\ U_2 = U_{\text{sample},2} - U_{\text{sample},1} = U_{\text{bat},2} + U_{\text{overhang}} \\ U_3 = U_{\text{sample},3} - U_{\text{sample},2} = U_{\text{bat},3} - U_{\text{overhang}} \end{cases} \quad (3)$$

Assuming that the cell voltage collected from the faulty branch at this time is U_n, where $n = 2, 3, \ldots, N-1$. The physical model of fault injection is built using MATLAB/Simscape software according to the hardware failure mechanism. The simulation results can be aligned with experimental data, leading to the derivation of a more universally applicable mathematical empirical model, as shown in eqn (4).

$$U_n - U_{\text{bat},n} = U_{\text{bat},n+1} - U_{n+1} = U_{\text{overhang}} \quad (4)$$

Similarly, the cell voltages at the time of disconnecting the first or last branches can be obtained through simulation, as shown in Fig. 4(b) and (d), respectively. The first and last cell sampling voltages in the above two disconnected cases are respectively shown as follows:

$$U_1 = U_{\text{bat},1} - U_{\text{overhang}} \quad (5)$$

$$U_{\text{last}} = U_{\text{bat,last}} + U_{\text{overhang}} \quad (6)$$

where U_1 is the sampled voltage displayed by the first cell and U_{last} is the sampled voltage displayed by the last cell.

2.2 Equalization loop closure experiments

In this paper, the sampling board is equipped with a passive equalization circuit featuring a resistor shunt configuration. Specifically, each cell is connected in parallel with a MOSFET switch and a resistor in series branch. When the cell voltage surpasses the threshold, the subsequent charging energy is dissipated through the resistor. This method is characterized by being cost-effective, easy to implement and particularly suitable for low-power applications, albeit with the consideration of residual heat generation. Transient overvoltage and

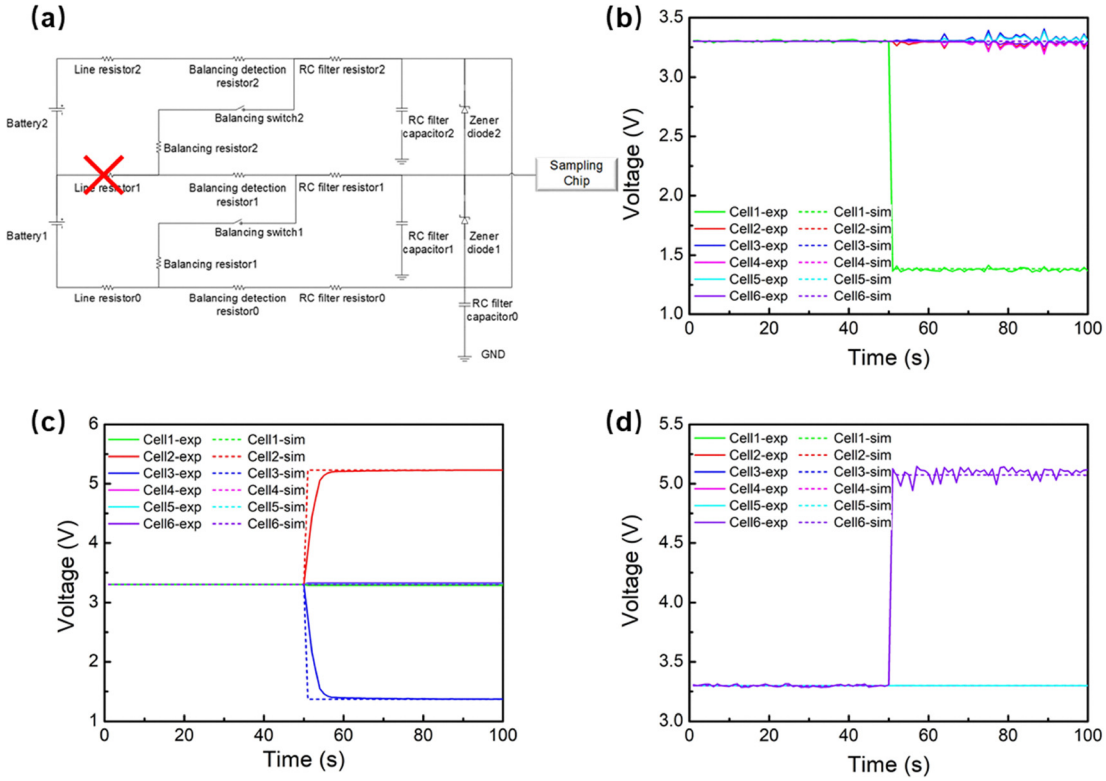

Fig. 4 Voltage values obtained from the fault injection experiment and physical model simulation for the sampling harness breakage: (a) location of the fault injection for sampling harness breakage. (b) Experimental and simulated voltage values of the sampling harness breakage fault corresponding to the first cell. (c) Experimental and simulated voltage values of the sampling harness breakage fault corresponding to the middle (*i.e.*, second) cell. (d) Experimental and simulated voltage values of the sampling harness breakage fault corresponding to the last cell.

overcurrent may occur during the charging and discharging of batteries, leading to potential sticking of MOSFET contacts due to transient arc ablation. It has been indicated that contact sticking can occur during both contact closure and contact breakage, resulting in the persistent closure of the equalization loop.[17] During fault simulation, the MOSFET on the sampling board can be replaced by a 0 Ω resistor. At this time, the equalization dissipation resistor R_B, the equalization detection resistor R_D and the third cell of the fault branch constitute the circuit, and it is also necessary to take into account that the wire resistance under the actual conditions (*i.e.*, $R_{L,2}$ and $R_{L,3}$) cannot be ignored. According to Ohm's law, the current in the equalization circuit can be obtained as follows:

$$I_B = \frac{U_{bat,3}}{R_B + R_D + R_{L,2} + R_{L,3}} \tag{7}$$

Kirchhoff's voltage law is then used to calculate the measured voltages of the first four affected branches, respectively, as shown in eqn (8).

$$\begin{cases} U_{sample,1} = U_{bat,1} \\ U_{sample,2} = U_{bat,1} + U_{bat,2} + I_B \cdot R_{L,2} \\ U_{sample,3} = U_{bat,1} + U_{bat,2} + I_B \cdot (R_{L,2} + R_B) \\ U_{sample,4} = U_{bat,1} + U_{bat,2} + U_{bat,3} + U_{bat,4} \end{cases} \tag{8}$$

The sampled voltage value of the corresponding cell is determined by the difference between the voltages of two adjacent circuits, as shown in the following equation.

$$\begin{cases} U_1 = U_{bat,1} \\ U_2 = U_{bat,2} + I_B \cdot R_{L,2} \\ U_3 = I_B \cdot R_B \\ U_4 = U_{bat,3} + U_{bat,4} - I_B \cdot (R_{L,2} + R_B) \end{cases} \tag{9}$$

From the calculation results of eqn (9) and the experimental results, it can be found that the voltage of the faulty cell is lower than the reference voltage. Simultaneously, as illustrated in Fig. 5(c), the voltages of the two adjacent cells surpass the reference voltage, with the latter cell exhibiting a more notable voltage shift. From eqn (9), a line resistance $R_{L,2}$ of 0.1915 Ω for branch 2 and $R_{L,3}$ of 0.2533 Ω for branch 3 can also be calculated. The simulation results derived from the fault physical model align with the experimental results, enabling the extension to establish the mathematical empirical model for equalization loop closure faults, as illustrated in eqn (10).

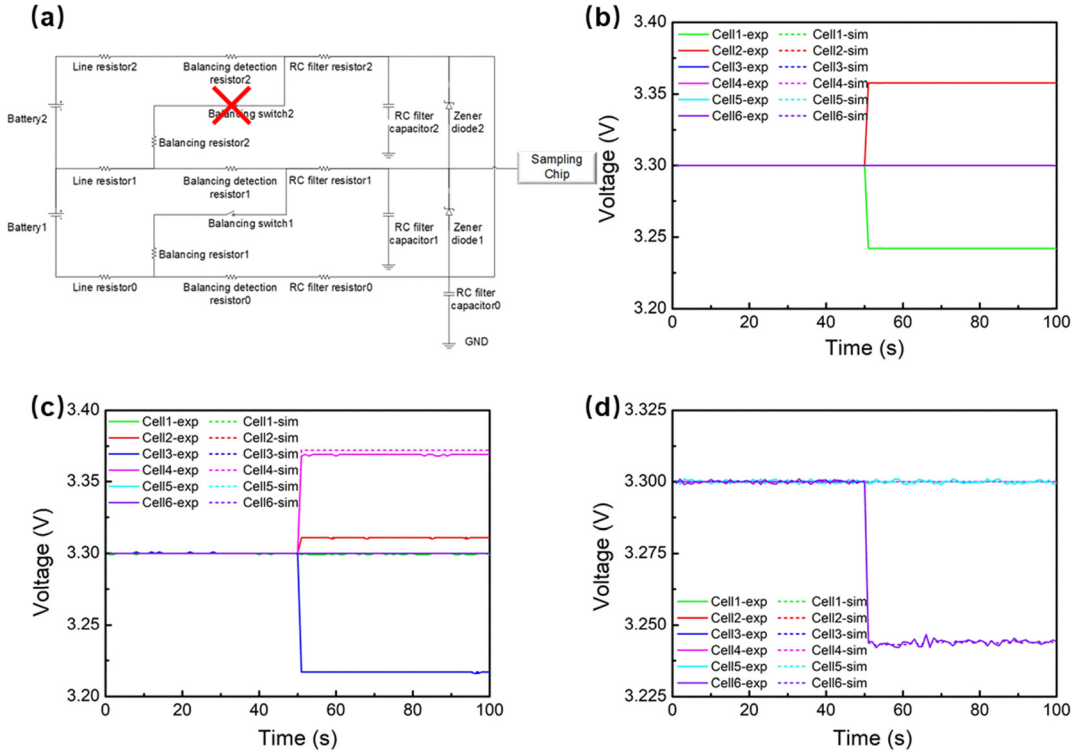

Fig. 5 Voltage values obtained from the fault injection experiment and physical model simulation for the equalization loop closure: (a) location of the fault injection for equalization loop closure. (b) Experimental and simulated voltage values of the equalization loop closure fault corresponding to the first cell. (c) Experimental and simulated voltage values of the equalization loop closure fault corresponding to the middle (*i.e.*, third) cell. (d) Experimental and simulated voltage values of the equalization loop closure fault corresponding to the last cell.

$$\begin{cases} U_n = \dfrac{U_{\mathrm{bat},n} \cdot R_{\mathrm{B}}}{R_{\mathrm{B}} + R_{\mathrm{D}} + R_{\mathrm{L},n-1} + R_{\mathrm{L},n}} \\ U_{n-1} = U_{\mathrm{bat},n-1} + \dfrac{U_{\mathrm{bat},n} \cdot R_{\mathrm{L},n-1}}{R_{\mathrm{B}} + R_{\mathrm{D}} + R_{\mathrm{L},n-1} + R_{\mathrm{L},n}} \\ U_{n+1} = U_{\mathrm{bat},n} + U_{\mathrm{bat},n+1} - \dfrac{U_{\mathrm{bat},n} \cdot \left(R_{\mathrm{L},n-1} + R_{\mathrm{B}}\right)}{R_{\mathrm{B}} + R_{\mathrm{D}} + R_{\mathrm{L},n-1} + R_{\mathrm{L},n}} \end{cases} \tag{10}$$

The same principle is deduced for the case of an equalization fault in the first cell, as shown in Fig. 5(b):

$$\begin{cases} U_{\mathrm{sample},1} = \dfrac{U_{\mathrm{bat},1} \cdot R_{\mathrm{B}}}{R_{\mathrm{B}} + R_{\mathrm{D}} + R_{\mathrm{L},0} + R_{\mathrm{L},1}} \\ U_{\mathrm{sample},2} = U_{\mathrm{bat},1} + U_{\mathrm{bat},2} \end{cases} \tag{11}$$

The sampled voltage values of the abnormal cell 1 and cell 2 can be obtained as shown in the following equation:

$$\begin{cases} U_1 = \dfrac{U_{\mathrm{bat},1} \cdot R_{\mathrm{B}}}{R_{\mathrm{B}} + R_{\mathrm{D}} + R_{\mathrm{L},0} + R_{\mathrm{L},1}} \\ U_2 = U_{\mathrm{bat},1} + U_{\mathrm{bat},2} - \dfrac{U_{\mathrm{bat},1} \cdot R_{\mathrm{B}}}{R_{\mathrm{B}} + R_{\mathrm{D}} + R_{\mathrm{L},0} + R_{\mathrm{L},1}} \end{cases} \tag{12}$$

As shown in Fig. 5(d), when an equalization loop closure fault occurs in the last battery cell, only its own voltage measurement is affected, and the voltage value is as follows:

$$U_{\mathrm{last}} = \frac{U_{\mathrm{bat,last}} \cdot R_{\mathrm{B}}}{R_{\mathrm{B}} + R_{\mathrm{F}} + R_{\mathrm{L,last}} + R_{\mathrm{L,last}-1}} \tag{13}$$

2.3 Filter capacitor breakdown experiments

Sensors can encounter hardware failures. For example, a sensor with poor connection is often short-circuited, and disconnection of the signal harness may trigger open-circuit faults in the sensor. In addition, some special sensors may be subject to external interference, resulting in inaccurate data when deployed in complex environments.[18] These issues are part of a hardware failure of the sensor, which are not difficult to notice when they occur.[19] In the course of developing BMS hardware, test engineers reported several instances under conditions of rapid plugging/unplugging and excessive loads, such as filter capacitor breakdown, voltage regulator diode failure and other device malfunctions. Breakdown is a term used in the field of electronic engineering, which mainly refers to a permanent damage to the device when the external voltage exceeds its nominal voltage. It usually results in a short circuit or disconnection of the device.

The RC filter in the sampling board is mainly to reduce the interference of environmental noise on the voltage signal. When the circuit is under normal operating conditions, the filter capacitor is directly grounded equivalent to disconnection. Therefore, this paper mainly simulates the filter capacitor breakdown into a short-circuit failure mode. Similarly, a 0 Ω resistor is used to replace the filter capacitor of the corresponding branch of cell 2, which is grounded at this time; thus, the measured voltage is 0 V. When two adjacent regulator

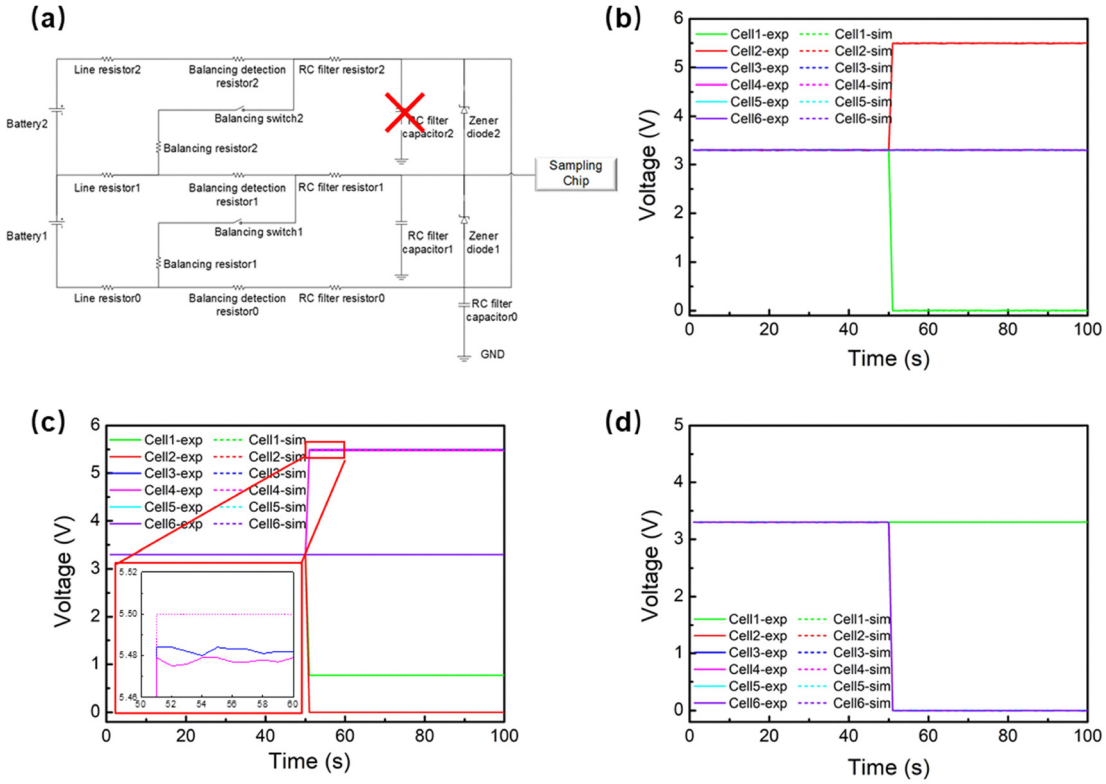

Fig. 6 Voltage values obtained from the fault injection experiment and physical model simulation for filter capacitor breakdown: (a) location of the filter capacitor breakdown fault injection. (b) Experimental and simulated voltage values of the filter capacitor breakdown fault corresponding to the first cell. (c) Experimental and simulated voltage values of the filter capacitor breakdown fault corresponding to the middle (*i.e.*, third) cell. (d) Experimental and simulated voltage values of the filter capacitor breakdown fault corresponding to the last cell.

diodes are in the conduction state, the voltage difference between the first branch and 0 V exceeds the forward conduction voltage of the diode, denoted as $U_{\text{pos\_lead}}$, and the measured voltage is about $U_{\text{pos\_lead}} = 0.76$ V. The voltage difference between the voltage of the third branch and 0 V is greater than the reverse conduction voltage of the diode, denoted as $U_{\text{neg\_lead}}$, and the measured voltage is about $U_{\text{neg\_lead}} = 7.5$ V. The measured voltages of the first four branches affected are shown in eqn (14), respectively.

$$\begin{cases} U_{\text{sample},1} = U_{\text{pos\_lead}} \\ U_{\text{sample},2} = 0 \\ U_{\text{sample},3} = U_{\text{neg\_lead}} \\ U_{\text{sample},4} = U_{\text{bat},1} + U_{\text{bat},2} + U_{\text{bat},3} + U_{\text{bat},4} \end{cases} \tag{14}$$

The sampled voltage value of the corresponding cell is determined by the difference between the voltages of two adjacent circuits, as shown in the following equation:

$$\begin{cases} U_1 = U_{\text{pos\_lead}} \\ U_2 = -U_{\text{pos\_lead}} \\ U_3 = U_{\text{neg\_lead}} \\ U_4 = U_{\text{bat},1} + U_{\text{bat},2} + U_{\text{bat},3} + U_{\text{bat},4} - U_{\text{neg\_lead}} \end{cases} \tag{15}$$

Specifically, despite the inability of the voltage difference between the fourth branch and the third branch to induce reverse conduction in the diode, the sampling voltage surpasses the upper and lower protection limits configured for the chip. Consequently, it is constrained to a range between 0 and 5.5 V. Similarly, the sampling voltages obtained from the other branches subject to these limitations are as follows:

$$\begin{cases} U_1 = 0.76 \text{ V} \\ U_2 = 0 \text{ V} \\ U_3 = 5.5 \text{ V} \\ U_4 = 5.5 \text{ V} \end{cases} \tag{16}$$

As shown in Fig. 6(c), the simulation results are consistent with the experimental results. The physical model of the filter capacitor fault is expressed as a mathematical empirical model. The sampling voltage of the faulty cell remains at 0 V, while the voltage of the preceding kth cell deviates by a magnitude of k forward conduction voltages. Normal voltage levels are reinstated when the measured voltage falls below the cumulative forward conduction voltages. Conversely, the voltage of the subsequent kth cell deviates by k reverse conduction voltages, until the normal value is restored when the measured voltage is less than the cumulative reverse conduction voltages, as

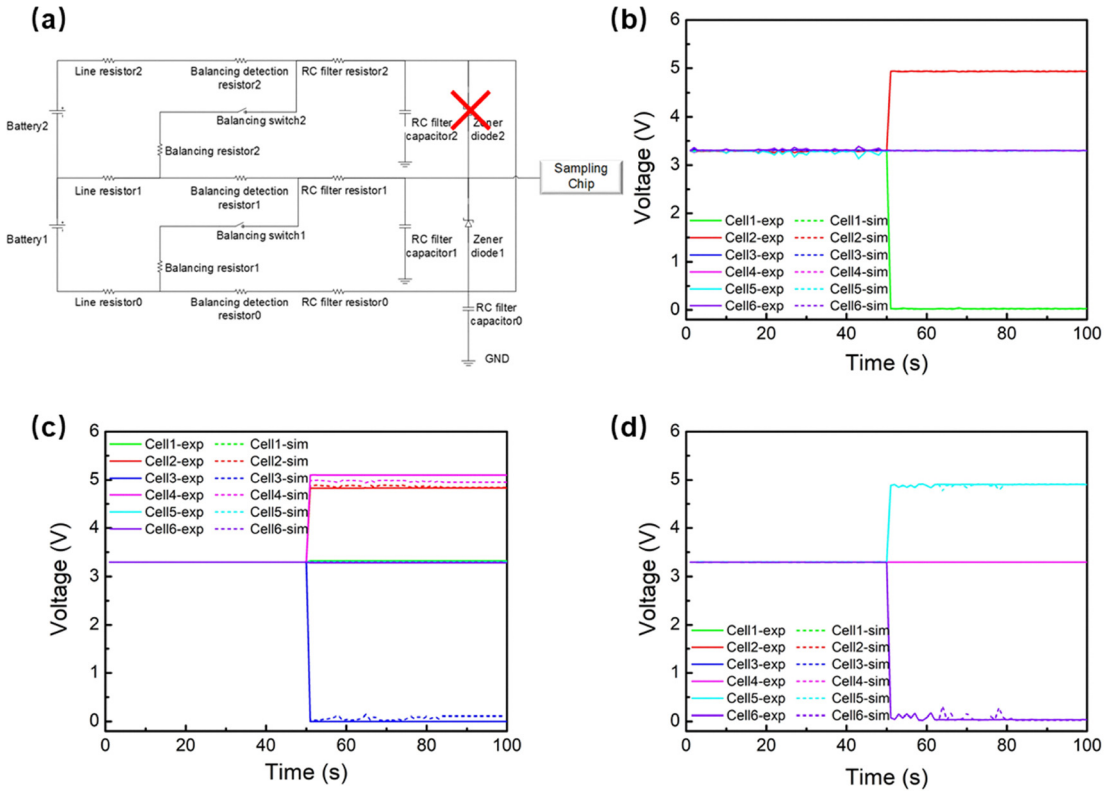

Fig. 7 Voltage values obtained from the fault injection experiment and physical model simulations for the voltage regulator diode breakdown: (a) location of the voltage regulator diode breakdown fault injection. (b) Experimental and simulated voltage values of the voltage regulator diode breakdown fault corresponding to the first cell. (c) Experimental and simulated voltage values of the voltage regulator diode breakdown fault corresponding to the middle (*i.e.*, third) cell. (d) Experimental and simulated voltage values of the voltage regulator diode breakdown fault corresponding to the last cell.

depicted in eqn (17).

$$\begin{cases} U_{\text{sample},n-k} = \sum_{i=1}^{n-k} U_{\text{bat},i} \left(k \cdot U_{\text{pos\_lead}} > \sum_{i=1}^{n-k} U_{\text{bat},i} \right) \\ U_{\text{sample},n-k} = k \cdot U_{\text{pos\_lead}} \left(k \cdot U_{\text{pos\_lead}} < \sum_{i=1}^{n-k} U_{\text{bat},i} \right) \\ U_n = 0 \\ U_{\text{sample},n+1} = U_{\text{neg\_lead}} \left(U_{\text{neg\_lead}} < \sum_{i=1}^{n+k} U_{\text{bat},i} \right) \end{cases} \quad (17)$$

In this case, the cell sampling voltages can be further derived as follows:

$$\begin{cases} U_1 = U_{\text{bat},1} \left(k \cdot U_{\text{pos\_lead}} > U_{\text{bat},1} \right) \\ U_1 = k \cdot U_{\text{pos\_lead}} \left(k \cdot U_{\text{pos\_lead}} < U_{\text{bat},1} \right) \\ U_{n-k} = U_{\text{bat},n-k} \left(k \cdot U_{\text{pos\_lead}} > \sum_{i=1}^{n-k} U_{\text{bat},i} \right) \\ U_{n-k} = 0 \left(k \cdot U_{\text{pos\_lead}} < \sum_{i=1}^{n-k} U_{\text{bat},i} \right) \\ U_n = 0 \\ U_{n+1} = 5.5 \end{cases} \quad (18)$$

When the first cell is injected fault, except the first two cells, the other cells have normal voltages. The faulty voltages are as follows (as shown in Fig. 6(b)):

$$\begin{cases} U_1 = 0 \\ U_2 = 5.5 \end{cases} \quad (19)$$

Similarly, after injecting the fault into the last cell, it can be found that except that the first cell sampling voltage is normal, the rest of the cell voltages are 0 V (as shown in Fig. 6(d)).

2.4 Voltage regulator diode breakdown experiments

In the case of overcurrent breakdown, the voltage regulator diode generally appears as a circuit break. The sampling circuit will be degraded to a differential filter circuit, and the over-voltage breakdown will appear as a short circuit state. A comprehensive exploration of these phenomena necessitates both the experimental simulation and the model simulation. In the fault injection experiment, the voltage regulator diode of the third branch is replaced by a 0 Ω short-circuit resistor (as shown in Fig. 7(a)). At this time, the potential of the two adjacent voltage measurement points of the diode is the same.

The current through the circuit is as follows:

$$I_3 = \frac{U_{\text{bat},3}}{R_{\text{RC},2} + R_{\text{D},2} + R_{\text{L},2} + R_{\text{RC},3} + R_{\text{D},3} + R_{\text{L},3}} \tag{20}$$

where $R_{\text{RC},i}$ is the resistance value of the ith RC filter, $R_{\text{D},i}$ is the value of the ith equalization detection resistor, and $R_{\text{L},i}$ is the value of the ith sampling line resistor. The measured voltages of the first four branches affected by the fault are as follows:

$$\begin{cases} U_{\text{sample},1} = U_{\text{bat},1} \\ U_{\text{sample},2} = U_{\text{bat},1} + U_{\text{bat},2} + I_3 \cdot \left(R_{\text{RC},2} + R_{\text{D},2} + R_{\text{L},2}\right) \\ U_{\text{sample},3} = U_{\text{bat},1} + U_{\text{bat},2} + I_3 \cdot \left(R_{\text{RC},2} + R_{\text{D},2} + R_{\text{L},2}\right) \\ U_{\text{sample},4} = U_{\text{bat},1} + U_{\text{bat},2} + U_{\text{bat},3} + U_{\text{bat},4} \end{cases} \tag{21}$$

The sampling voltage of the different cells can be obtained by making a difference between two adjacent cells as shown in the following equation (as shown in Fig. 7(c)).

$$\begin{cases} U_1 = U_{\text{bat},1} \\ U_2 = U_{\text{bat},2} + I_3 \cdot \left(R_{\text{RC},2} + R_{\text{D},2} + R_{\text{L},2}\right) \\ U_3 = 0 \\ U_4 = U_{\text{bat},3} + U_{\text{bat},4} - I_3 \cdot \left(R_{\text{RC},2} + R_{\text{D},2} + R_{\text{L},2}\right) \end{cases} \tag{22}$$

The resulting mathematical empirical model is as follows:

$$\begin{cases} U_n = 0 \\ U_{n-1} = U_{\text{bat},n-1} + \dfrac{U_{\text{bat},n} \cdot \left(R_{\text{D},n-1} + R_{\text{RC},n-1} + R_{\text{L},n-1}\right)}{\left(R_{\text{D},n} + R_{\text{D},n-1} + R_{\text{RC},n} + R_{\text{RC},n-1} + R_{\text{L},n} + R_{\text{L},n-1}\right)} \\ U_{n+1} = U_{\text{bat},n} + U_{\text{bat},n+1} - \dfrac{U_{\text{bat},n} \cdot \left(R_{\text{D},n-1} + R_{\text{RC},n-1} + R_{\text{L},n-1}\right)}{\left(R_{\text{D},n} + R_{\text{D},n-1} + R_{\text{RC},n} + R_{\text{RC},n-1} + R_{\text{L},n} + R_{\text{L},n-1}\right)} \end{cases} \tag{23}$$

Injecting a voltage regulator diode breakdown fault into the first cell (as shown in Fig. 7(b)), the first two cells can be sampled with the voltage as shown:

$$\begin{cases} U_1 = 0 \\ U_2 = U_{\text{bat},1} + U_{\text{bat},2} - \dfrac{U_{\text{bat},1} \cdot \left(R_{\text{D},0} + R_{\text{RC},0} + R_{\text{L},0}\right)}{\left(R_{\text{D},1} + R_{\text{D},0} + R_{\text{RC},1} + R_{\text{RC},0} + R_{\text{L},1} + R_{\text{L},0}\right)} \end{cases} \tag{24}$$

Injecting a fault into the last cell yields, the last two cell sampling voltages are as follows (as shown in Fig. 7(d)):

$$\begin{cases} U_5 = U_{\text{bat},5} + \dfrac{U_{\text{bat},6} \cdot \left(R_{\text{D},5} + R_{\text{RC},5} + R_{\text{L},5}\right)}{\left(R_{\text{D},6} + R_{\text{D},5} + R_{\text{RC},6} + R_{\text{RC},5} + R_{\text{L},6} + R_{\text{L},5}\right)} \\ U_6 = 0 \end{cases} \tag{25}$$

In summary, by conducting fault injection experiments and replicating faults in the physical model, empirical mathematical models corresponding to the four failure modes (*i.e.*, sampling harness breakage, equalization loop closure, filter capacitor breakdown and voltage regulator diode breakdown) can be developed. This enables the generation of a substantial volume of simulated failure data for subsequent algorithmic development. Meanwhile, the method can also be used to explore other possible device failure patterns. This paper only discusses the above four types of typical failures that have occurred in the real vehicle and test process.

3. Sampling fault diagnosis algorithm

As shown in Fig. 8, the research in this paper follows the procedure of "test–model–algorithm–validation". During this process, various new methods, including multidisciplinary simulation tools, fault tree analysis and statistical modeling, are involved to enable better fault diagnosis and mechanism traceability.[20] After fault injection experiments and data analysis of the sampling circuit, the constant voltage source in the physical simulation model is replaced with a higher accuracy second-order equivalent circuit model to simulate a real battery. The mathematical laws obtained from Section 2 are used to generate a simulation dataset for algorithm development and validation. In this section, a threshold-based sampling fault detection algorithm is first developed based on the data distribution pattern. And then, a deep learning model based on the residual network (ResNet) and bi-directional long-short term memory (BiLSTM) neural network is built for realizing higher accuracy sampling fault identification.

3.1 Threshold-based detection algorithm

When the collected signal value of the sensor deviates from the actual value, it can be assumed that the sensor has a fault. Although design principles and circuit topologies are not the same for different types of sensors, the signal patterns exhibited by faulty sensors are similar. Sensor failures can be categorized in various ways based on the severity of the failure and its maintainability. This classification often involves distinguishing between soft and hard failures. In this paper, soft failures (*e.g.*, sampling harness breaks and equalization loop closures) are explored, while hard failures are also addressed, exemplified by filter capacitor breakdown and voltage regulator diode breakdown. Faults are further categorized based on different behavior modes and frequencies of occurrence, leading to a division into intermittent or permanent faults. Typically, soft faults manifest as intermittent issues, while hard faults present as permanent malfunctions. Noteworthily, if without an attended maintenance for a long period, soft faults may evolve into permanent faults. Faults can be classified based on their causes into various categories, including a bias

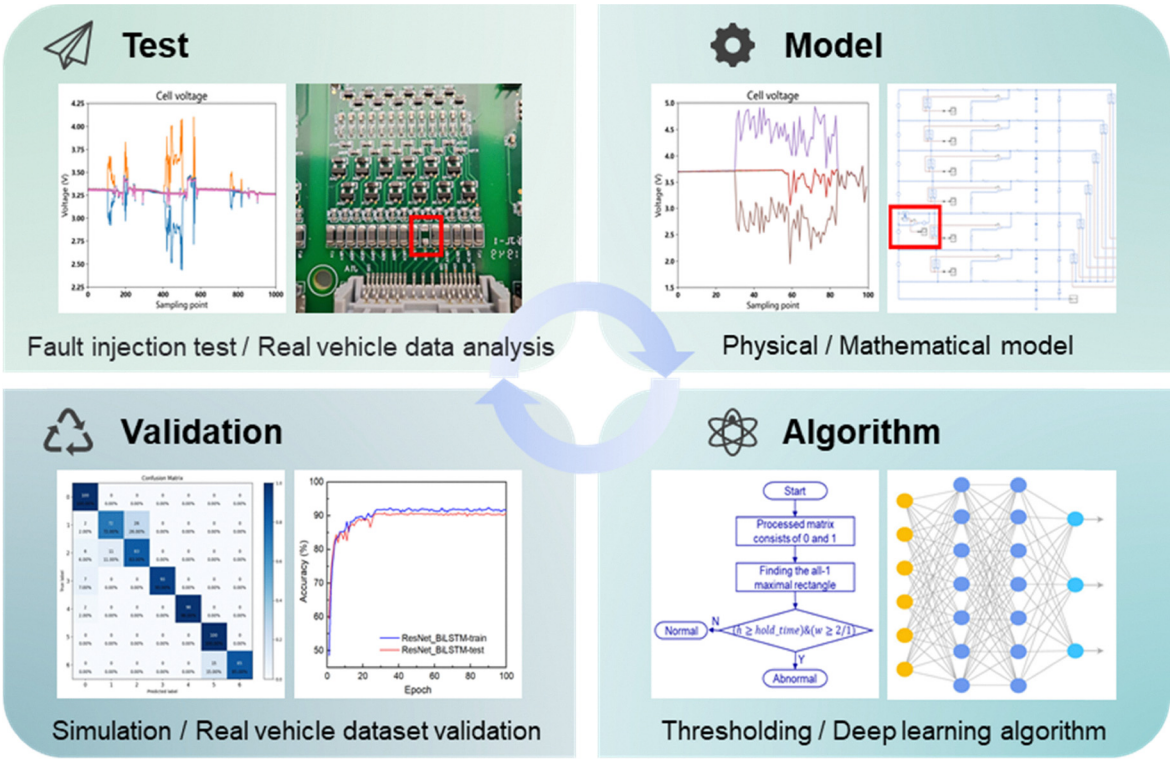

Fig. 8 The development of the sampling fault diagnosis algorithm in this paper, following the "test–model–algorithm–validation" procedure.

fault, an impulse fault, a drift fault, a periodic fault, an open circuit fault and a short circuit fault. The causative factors and behaviors of different failure types are detailed in Table 1.

According to Section 2, the data distribution pattern resulting from various failure types can be summarized. The sampling faults are mainly characterized by the following features:

(1) Failure of adjacent battery cells. Since the cell sampling voltage is obtained by differential calculations, the failure of a cell at any position other than the first and last two cells usually affects the sampling voltage calculation results of the adjacent cells.

(2) Similarity of voltage differential sequences. From the mathematical model of sampling faults, it can be seen that the difference between the voltage of each faulty battery and the reference voltage has similarity (usually taken the median or the mode of all cell voltages). Moreover, the sequence of voltage differences at adjacent moments of each faulty cell also follows a consistent pattern.

(3) Abnormal voltages could have fixed upper and lower limits. Battery internal failures (*e.g.*, capacity diving, internal short circuits and electrolyte leakage typically) lead to a gradual deviation of voltage data from that of normal cells. Unlike stable voltage values at the upper and lower limits of 5.5 V and 0 V, these failures disrupt the expected voltage stability.

A threshold-based sampling fault diagnosis algorithm has been developed on the basis of above laws, the flow of the algorithm is shown in Fig. 9, and the pseudo code is shown below:

Table 1 The causative factors and behaviors of different failure types

	Fault type	Fault causes
1	Bias fault	Stable deviations between the sensor output and the true value due to bias currents in the circuit, such as in this paper when the sampling fault occurs at the non-start and non-end cells.
2	Impulse fault	The sensor is disturbed by a certain pulse signal, such as a faulty connection between cells in the battery pack. An abnormal pulse signal will appear when the vehicle vibration reaches a certain amplitude.
3	Drift fault	Deviation of the output due to performance degradation, temperature drift, zero drift, *etc.*, such as capacity diving and electrolyte leakage of the battery.
4	Periodic fault	Sensors disturbed by a certain periodic signal cause the measured value to show a periodic trend, typically occurring in rotating components such as motors and bearings.
5	Open circuit fault	Sensor outputs are maximized due to disconnection of the power source system, component damage, *etc.*, such as the filter capacitor breakdown and voltage regulator diode breakdown faults involved in this paper.
6	Short circuit fault	Sensor outputs are close to zero due to component breakdown, device sticking, *etc.*, such as the filter capacitor breakdown and voltage regulator diode breakdown faults involved in this paper.

Algorithm 1. Main algorithm of sampling fault diagnosis based on the threshold

Result: Whether there is a fault in the sampling data (flag), and the start and end points of the test data segment (start_end_points).
Input: Voltage sampling data (volt).
Initialize: $\text{cell}_{\text{limit}}$ and $\text{time}_{\text{limit}}$ represent the thresholds of diffv_Z and dv_Z matrices, and volt_{up} and volt_{low} denote the upper and lower voltage limits of the volt matrix, respectively. These parameters are determined using the 3-sigma principle, according to which the original matrix is converted into a matrix containing only 0 and 1 elements. hold_time represents the shortest length that the faulty segment needs to satisfy.
(1) Take the median of all cell voltage data at each sampling moment as the reference to obtain the inter-cell voltage difference matrix diffv, and then calculate the diffv to obtain the corresponding Z-score matrix diffv_Z; use the voltage data at the previous sampling moment as the reference to obtain the time–difference matrix dv of cell voltages, respectively, and then calculate the dv to obtain the corresponding Z-score matrix dv_Z. The Z-score matrix is calculated using the following equation:

$$Z = \frac{X - \bar{X}}{S}$$

where X is the original data, $\bar{X}$ is the reference data (usually the mean, which can be replaced by the median or the mode depending on the actual scenario), and S is the standard deviation.
(2) Traverse each element of the matrix and convert it to a matrix containing only 0 and 1 elements according to the following rules:

$$\begin{cases} \text{diffv\_}Z_{i,j} = 1 & \left(\left|\text{diffv\_}Z_{i,j}\right| > \text{cell}_{\text{limit}}\right) \\ \text{diffv}_{Z_{i,j}} = 0 & \left(\left|\text{diffv}_{Z_{i,j}}\right| < \text{cell}_{\text{limit}}\right) \end{cases}$$

$$\begin{cases} \text{dv\_}Z_{i,j} = 1 & \left(\left|\text{dv}_{Z_{i,j}}\right| > \text{time}_{\text{limit}}\right) \\ \text{dv\_}Z_{i,j} = 0 & \left(\left|\text{dv}_{Z_{i,j}}\right| < \text{time}_{\text{limit}}\right) \end{cases}$$

$$\begin{cases} \text{volt}_{i,j} = 1 & \left[\left(\text{volt}_{i,j} > \text{volt}_{\text{up}}\right)\middle|\left(\text{volt}_{i,j} < \text{volt}_{\text{low}}\right)\right] \\ \text{volt}_{i,j} = 0 & \left(\text{volt}_{\text{up}} < \text{volt}_{i,j} < \text{volt}_{\text{low}}\right) \end{cases}$$

(3) Find the all-1 maximal rectangular region in the three matrices of diffv_$Z_{i,j}$, dv_$Z_{i,j}$ and $\text{volt}_{i,j}$, which only contain element 0 or 1, respectively. Save the position of the upper-left corner of the region max_pos and the size of the all-1 matrix max_size.
(4) The width of the all-1 matrix size is used to determine the existence of adjacent cells, and the length of the matrix size is used to determine whether the number of consecutive fault sampling points exceeds the hold_time length. If the fault retention time in matrices diffv_$Z_{i,j}$ and dv_$Z_{i,j}$ is greater than that in hold_time, and the abnormal sampling dimension is greater than 2, an alarm will be triggered. For matrix $\text{volt}_{i,j}$, as long as one dimension of sampling triggers under the condition that the fault retention time meets the requirements, an alarm will be issued.
(5) The results of the three matrices are evaluated for union, and a sampling fault is considered to exist if one of the matrices satisfies the condition.

The pseudo-code of finding the all-1 maximal rectangle in the matrix mentioned in step 3 of Algorithm 1 is as follows:

Algorithm 2. Algorithm of finding the all-1 maximal rectangle

Result: The area of the all-1 maximal rectangle (max_area), the position of the upper left corner (max_pos), and the length and width of the coverage (max_size).
Input: A matrix consists of 0 and 1 with rows and columns m and n, respectively.
Initialize m, n, heights, max_area, max_pos, and max_size.
for i from 0 to m:
  Update heights for each column j using matrix [i][j]
  Initialize stack with -1
  for j from 0 to n:
    while (stack not empty) & (heights [j] $<$ heights [stack [-1]]):
      – Pop the last element from stack as h
      – Calculate w and update max_area, max_pos, and max_size if necessary
    Append j to stack
Return max_area, max_pos, max_size

The algorithm first traverses the matrix by rows and columns, respectively, using a one-dimensional matrix of equal width with the input matrix. This auxiliary matrix is used to store a sequence of histograms with each row serving as the foundation. Subsequently, a monotonically decreasing stack is generated from this process. The height of the all-1 matrix to be solved is the current stack top value, and the width is the index of the current column minus the index of the top stack column. If the stack is empty, the width is the index j of the current column. The rectangle with the largest area is computed and selected as the required anomaly fragment.

While the logic of the threshold-based algorithm is relatively simple, it is imperative to determine the threshold size based on the data distribution pattern of the actual fault. It should be noted that various fault types may trigger the same threshold alarm, posing a challenge in identifying the specific cause of the fault. Therefore, it is necessary to develop a set of diagnosis algorithms which can realize fault recognition and cause inference at the same time.

3.2 ResNet–BiLSTM-based diagnosis algorithm

With the development of AI technology, various deep learning algorithms are continuously applied in the fields of image processing, natural language processing and knowledge mining. At the same time, some classical neural network structures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs) and graph neural network (GNNs) have emerged. Appropriate algorithmic models have been developed to suit the needs of different scenarios.[21]

CNNs have been widely used in the fields of pattern classification, object detection and object recognition, which utilize efficient convolutional computation to replace complex image

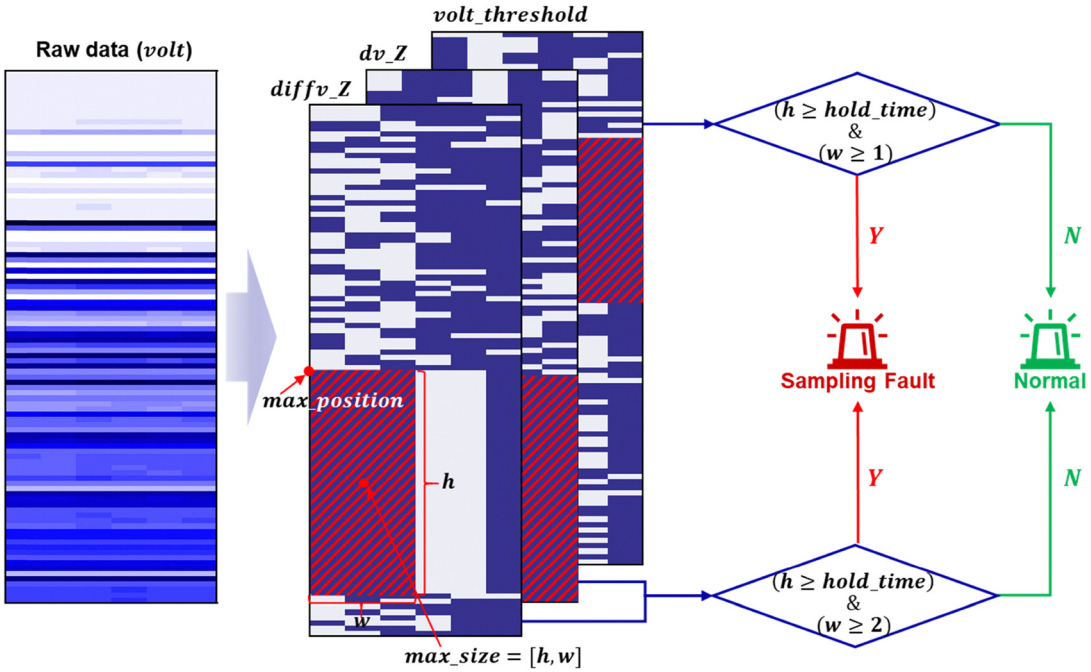

Fig. 9 Flow of the threshold-based detection algorithm.

feature extraction operations. The network is mainly composed of convolutional layers, pooling layers, activation functions, *etc.* By combining and arranging different layers, classical network structures such as ResNet and the visual geometry group (VGG) are created.[22,23] The close connection between the CNN layers and the ability to acquire spatial information make it particularly suitable for image processing and understanding with strong generalization ability and high computational efficiency. Fig. 10(a) shows the network structure of a ResNet block, which consists of a convolutional layer, a normalization layer (also known as the batch normalization layer), an activation function (also known as ReLU) and a residual feedback shortcut connection. The design of the residual block avoids the phenomena of gradient vanishing and gradient explosion when deepening the layers of the network, and speeds up the convergence of the network.

RNNs are mainly used to process time-series related data, which is recursive in the evolutionary direction of the sequence and is formed by all the recursive units being chained together. Variant unit structures such as long short-term memory (LSTM) and gated recurrent unit (GRU) have also been designed to address the problem of temporal memory and dependence of sequence data.[24] On this basis, researchers have found that the LSTM structure combining forward and backward processing can better capture bi-directional temporal dependencies. This variant is recognized as a Bi-LSTM network.[25] In this paper, the structure of the LSTM neural network is shown in Fig. 10(b). A pivotal feature is the gating mechanism to control the information transfer path, and the three gates need to be controlled as the input gate i_t, forgetting gate f_t and output gate o_t, respectively.[26] The gates in the LSTM unit are soft, which assume values between 0 and 1. These values signify the proportion of information that is allowed through. The functions of these three gates are as follows:

(1) The forgetting gate f_t controls the amount of internal state c_{t-1} information forgotten at the previous moment;

(2) The input gate i_t controls the amount of saved information about the candidate state $\tilde{c}_t$ at the current moment;

(3) The output gate o_t controls the amount of information passed from the internal state c_t to the external state h_t at the current moment.

The three gates and the candidate states are computed as follows:

$$\begin{bmatrix} \tilde{c}_t \\ o_t \\ i_t \\ f_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(W \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b \right) \tag{26}$$

Long-term memory determines whether historical information needs to be absorbed, retained and discarded from the training data through the gating strategy. Consequently, the acquisition of important historical information in memory unit c is stronger than short-term memory and weaker than long-term memory. This characteristic enables LSTM neural networks to enhance the performance compared to traditional RNNs.

In this paper, the measured voltage signal is affected by the working conditions and noise. Solely relying on CNNs to extract the correlation relationship of multiple cells would neglect the inherent temporal characteristics of the voltage data itself. While the RNN considers the temporal characteristics of voltage, it is difficult to focus on the problem of adjacent cell signal anomalies caused by sampling faults, and the network

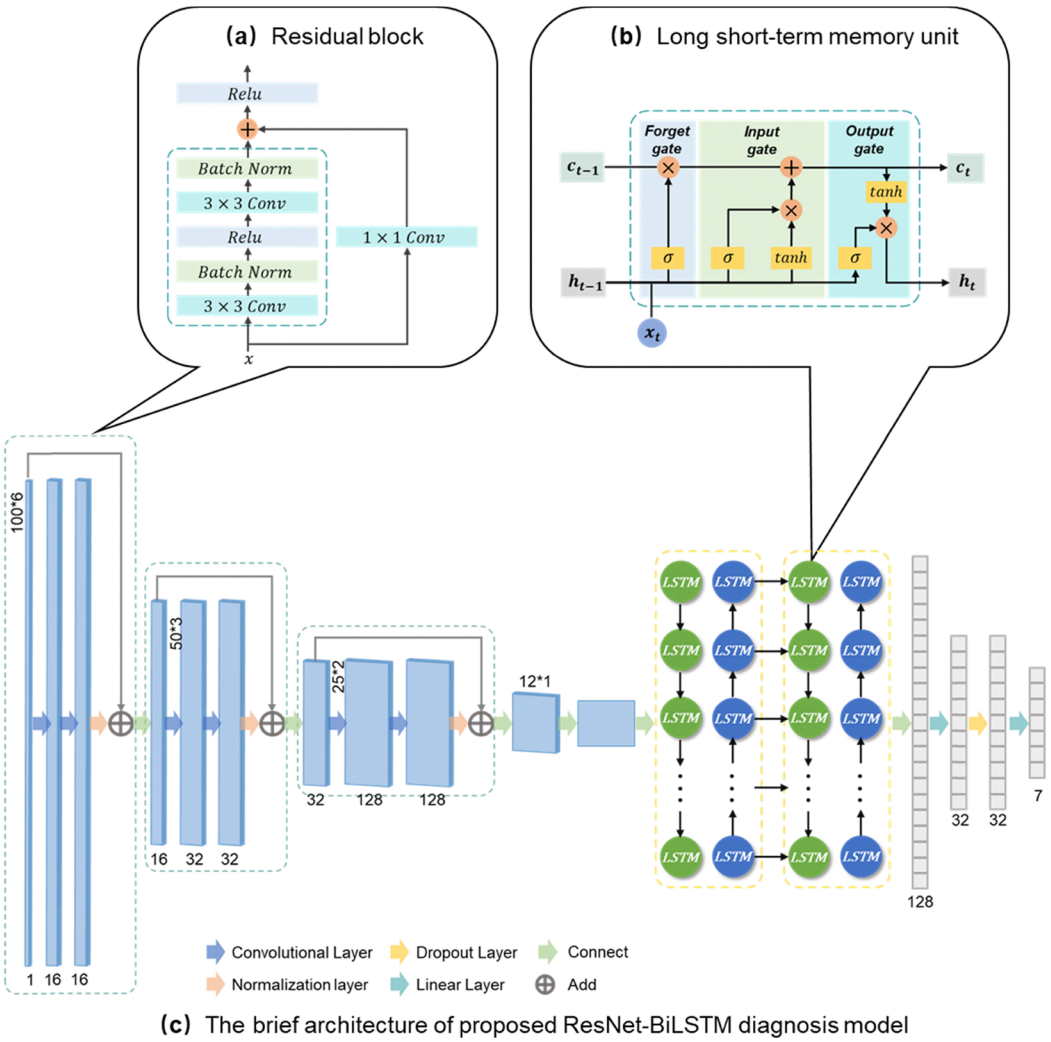

Fig. 10 The brief architecture of the proposed ResNet–BiLSTM diagnosis model and its details of the module structure. (a) Residual block. (b) Long short-term memory unit. (c) The brief architecture of the proposed ResNet–BiLSTM diagnosis model.

itself may have gradient disappearance and gradient explosion phenomenon in the process of model training. The voltage data in this paper contains local spatial features pertaining to the interaction between faulty cells and adjacent cells, as well as multiscale-dependent temporal features. These aspects cannot be extracted from the two kinds of features by a separate network structure; thus in this paper, we construct the fault diagnosis model by connecting the ResNet network and BiLSTM network in tandem.[27] The ResNet–BiLSTM fault diagnosis model constructed in this paper is shown in Fig. 10(c). The structure of the model is sequentially as follows:

(1) Firstly, the algorithm stacks three residual blocks and one max-pooling layer, which is designed to extract local features. The residual block contains two sets of the 3 × 3 convolutional kernel layer, batch normalization layer and ReLU activation function layer, and uses the shortcut mechanism to accelerate model training.

(2) In order to capture long-term dependencies from a sequence of local features, a Bi-LSTM layer is introduced to build ResNet–BiLSTM networks following the ResNet layers. The model is composed of a 2-layer bidirectional LSTM network, where a single-layer LSTM network encompasses 128 hidden states.

(3) In Section 4, model ablation experiments illustrate that the 2D ResNet–BiLSTM network effectively learns numerous spatio-temporal features from the data. Furthermore, a dropout mechanism is integrated into the final linear layer of the classification network to prevent the model from succumbing to overfitting.

The performance of a deep learning neural network model has an important relationship with the hyper-parameter settings. In this paper, the GridSearch technique is used to find the optimization of the network parameters, and the finalized parameter settings are as follows: batch size = 64, learning r ate = 0.001, and optimizer = Adam.

4. Results and discussion

In this section, the proposed algorithm will be tested based on simulation and real-vehicle datasets. The evaluation criteria for

the classification results will include accuracy, the F1-score and the kappa coefficient. The indicators are calculated as shown in the following equation, where TP is truely positive, TN is truely negative, FP is false positive, and FN is false negative.

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{27}$$

$$\begin{cases} P = \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \\ R = \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \end{cases} \xrightarrow{\text{yields}} \mathrm{F1}_{\mathrm{score}} = \frac{2 \times P \times R}{P + R} \tag{28}$$

The kappa coefficient, namely a kappa index of agreement (KIA), is a value used in statistics to assess the multi-classification effect. In practical applications, the range of value is generally [0,1], and higher values represent better classification accuracy achieved by the model.

$$\begin{cases} p_0 = \dfrac{\sum_{i=1}^{n} x_{ii}}{N} \\ p_c = \dfrac{\sum_{i=1}^{n} x_{i+} x_{+i}}{N^2} \end{cases} \xrightarrow{\text{yields}} \mathrm{KIA} = \frac{p_0 - p_c}{1 - p_c} \tag{29}$$

where x_{ii} is the element on the diagonal of the confusion matrix, x_{i+} is the sum of all elements in row i, x_{+i} is the sum of all elements in column i, and N is the sum of all elements.

4.1 Tests on simulated datasets

In order to verify whether the proposed model can accurately identify the sampling fault data segments without misreporting other fault types as sampling faults, it is necessary to construct a dataset containing different battery status and sampling faults for the development of the algorithm. The mathematical model of sampling faults is used to construct a simulation dataset containing: all normal cells, the existence of the internal short circuit cell (connected an internal short circuit resistor in parallel with the battery Rint model,[28] abbreviated as ISC in Fig. 11(b)), the existence of the aging cell (the capacity of the cell is reduced to 75%–90% of the initial, abbreviated as CF in Fig. 11(b)), sampling harness breakage, equalization loop closure, filter capacitor breakdown and voltage regulator diode breakdown, with a total of seven types of battery status. Each type of battery generates 500 segments, respectively, and each segment contains data of 6 cells of length 100. The simulation dataset is divided into training and test sets at a ratio of 4 : 1, where the training data comprises a comprehensive total of 2800 segments, and for the test data, 100 segments are randomly selected from each type, culminating in a total of 700 segments. All the training and testing of the algorithms in this paper are done on the same hardware devices, the CPU model is i7-10875H with 16G RAM, and the GPU model is GeForce RTX 2060 with 6G RAM.

Ablation experiments are designed to explore the accuracy of models composed of various algorithmic blocks. These experiments aimed to assess the effectiveness of the algorithms

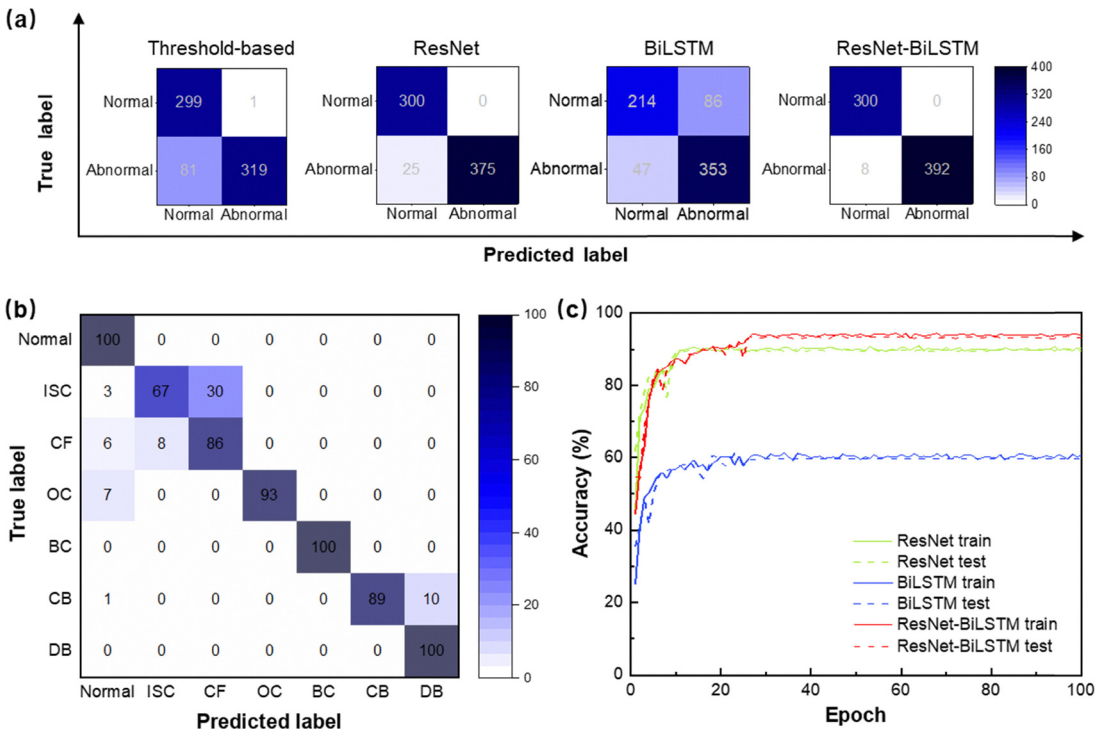

Fig. 11 The test results of the ResNet–BiLSTM neural network on simulated datasets: (a) the binary classification confusion matrices of threshold-based, ResNet, BiLSTM and ResNet–BiLSTM algorithms. (b) Multi-classification confusion matrix of the ResNet–BiLSTM algorithm. (c) Accuracy of training and test on simulated datasets per epoch.

proposed by comparing the results of ResNet, BiLSTM and ResNet–BiLSTM algorithms, respectively.[29] The results of the algorithms on the test set are shown in Fig. 11, where Fig. 11(a) shows the binary classification confusion matrices obtained from the experiments of the different algorithms on the test set for determining whether a sampling fault occurs or not. Each row of the matrix represents the number of samples for true classification, and each column represents the number of samples for predictive classification. The more samples falling on the diagonal of each confusion matrix indicates the better classification of the model. The results of the confusion matrix are further quantified using the accuracy and F1-score indicators as shown in Table 2, which shows that the results of the ResNet–BiLSTM algorithm are significantly better than those of the other three algorithms, with a classification accuracy of 98.86%. Table 2 also includes the most fundamental CNN-LSTM algorithm as a benchmark for comparison, utilized to analyze the ResNet–BiLSTM algorithm's capability in characterizing the time-space distribution characteristics of fault data. The results likewise demonstrate that the ResNet–BiLSTM algorithm exhibits marked advantages in both accuracy and classification performance. The last two columns of the table compare the computational time and model size required by different algorithms for processing the test data. It is found that the ResNet–BiLSTM algorithm achieves superior results without significantly increasing the computational demand. Additionally, the F1-score attains a value of 0.9868, which indicates that the model is able to achieve excellent classification results and sufficient stability. It is also found that the simplest threshold recognition algorithm can also achieve high accuracy, and its indicators are even better than the BiLSTM algorithm. Since the threshold-based algorithm is unable to multi-classify the samples, the confusion matrices of the three neural network algorithms are further subdivided into multi-classification confusion matrices for seven types of data. Among these, the confusion matrix of the ResNet–BiLSTM algorithm is shown in Fig. 11(b). It can be seen that the true labels and the classification labels are able to almost completely correspond to the main diagonal of the matrix, indicating that the algorithms have good classification ability for different fault types. The figure also shows that there are no normal samples misclassified, which can greatly reduce costs in real industrial application scenarios. Although 8 faulty samples are misclassified as normal, they may still be recognized by the fault diagnostic algorithm for the battery itself.[30] Using the KIA as an evaluation indicator of the algorithm's multi-classification ability, it is found that the ResNet–BiLSTM algorithm is still better than the ResNet and BiLSTM algorithms, which indicates that the algorithm has the ability to accurately classify the failure modes. The inclusion of simulated internal short circuit and capacity degradation fault data enables the testing of whether the algorithm has the ability to distinguish between sampling faults and battery faults. As can be seen from the confusion matrix, the number of false alarms between different fault types is relatively small. This is primarily attributable to the fact that battery faults typically affect only a single cell and induces sustained voltage outliers, whereas sampling faults intermittently corrupt measurements across multiple channels and frequently manifest distinctive anomalous values (such as 0 V). Fig. 11(c) illustrates the accuracy curves of the three neural networks in the training and test sets. The ResNet–BiLSTM algorithm obtains stable results at about the thirtieth epoch, and its accuracy in the training and test sets is significantly higher than that of the other three algorithms. In summary, it is considered to have the possibility of being applied in practical engineering applications.

Table 2 Comparison of the classification accuracy of different algorithms

	ACC	F1-score	KIA	Time (ms)	Model size (M)
Threshold-based	0.8829	0.8794	—	22.2311	—
ResNet	0.9643	0.9600	0.8850	118.3542	6.31
BiLSTM	0.8100	0.7629	0.7783	92.5617	4.09
CNN-LSTM	0.9443	0.9349	0.8533	102.1473	7.52
ResNet–BiLSTM	0.9886	0.9868	0.8917	121.2940	8.72

4.2 Tests on real-vehicle datasets

The ResNet–BiLSTM neural network trained by the simulation dataset was used for the real-vehicle dataset. Since filter capacitor breakdown and voltage regulator diode breakdown were not found to be real occurrences on the real-vehicle for the time being, 10 data segments of the normal battery, sampling harness breakage and equalization loop closure were selected for testing, respectively. Although real-vehicle battery packs usually consist of numerous battery cells which are connected in series and/or parallel configurations to meet the voltage and capacity requirements, and the BMS also utilizes a single chip to sample 6 adjacent cells with sequential numbers; thus, the real-vehicle data are still intercepted as a 100×6 matrix for testing. The real-vehicle datasets utilized in this study employ a sampling interval of 30 s, which also constitutes the minimum sampling frequency mandated by current standards. Employing more advanced sampling intervals of 1 s, 2 s, or 10 s enables the acquisition of more comprehensive fault data and facilitates more stable model performance. Furthermore, based on modeling and analysis of the fault circuit, it is revealed that anomalous data associated with different fault types exhibit fixed mathematical patterns. Consequently, the fault detection algorithms trained in this work can be directly transferred to scenarios employing other sampling intervals.

Fig. 12(a) shows the sampling harness breakage fault occurring in the real vehicle. It can be seen that except that the voltages of cell 1 and cell 2 deviate as outliers and exhibit symmetrical deviations, all the other cells demonstrate consistent behavior. Since the fault disappears at the end of the segment, it is hypothesized that the fault may be due to poor contact. As shown in Fig. 12(b) for the equalization loop closure fault occurring in the real vehicle, cell 3 is shifted downward about the reference voltage, cell 2 and cell 4 are shifted upward about the reference voltage, and the shift amplitude of cell 2 is obviously smaller than that of cell 4. Consequently, it can be

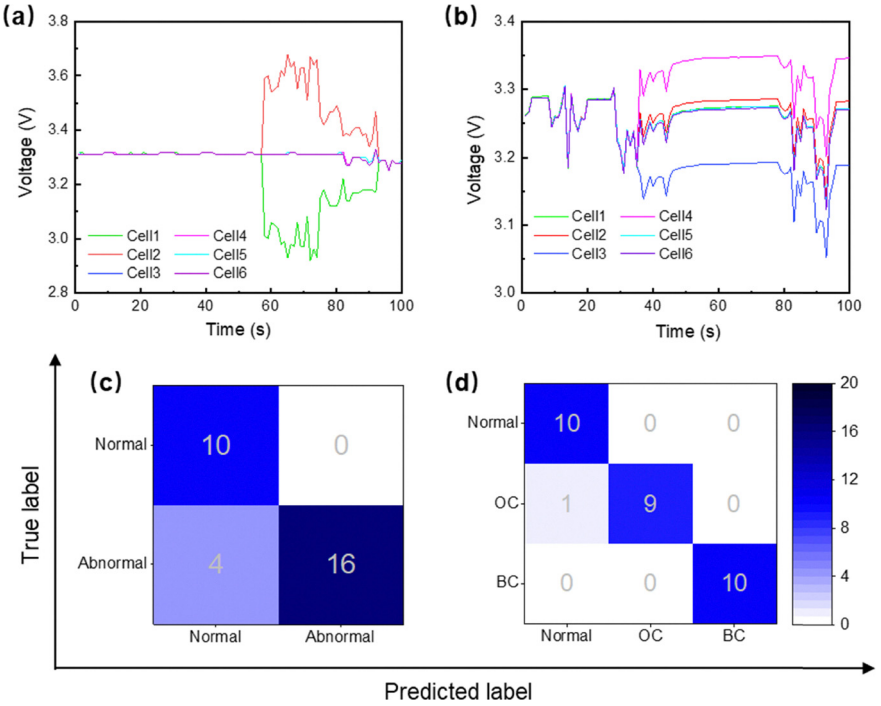

Fig. 12 The test results of the ResNet–BiLSTM neural network in real-vehicle datasets: (a) the sampling harness breakage fault occurring in the real vehicle. (b) The equalization loop closure fault occurring in the real vehicle. The confusion matrices of (c) threshold-based and (d) ResNet–BiLSTM algorithms.

considered that the fault is caused by the equalization loop closure corresponding to cell 3.

Fig. 12(c) shows the test results of the threshold-based fault detection algorithm, which achieves an accuracy of 86.67% on the real-vehicle dataset, and 4 fault samples are not accurately recognized in them. Fig. 12(d) illustrates that the accuracy of the ResNet–BiLSTM algorithm reaches 96.67%, and its KIA attains 0.9450. Importantly, no normal samples are erroneously classified as abnormal. Since the occurrence of sampling faults is a small sample event and the probability of a failure vehicle is less than 0.1%, the accuracy achieved by the ResNet–BiLSTM algorithm proposed in this paper on the real-vehicle test set can meet the application requirements in real-world scenarios.

5. Conclusions

In this paper, a cloud-based battery data sampling fault diagnosis method based on a ResNet–BiLSTM neural network is proposed, and the algorithm can achieve high-precision sampling fault type recognition relying only on voltage data in the cloud platform. Firstly, this paper describes the operational principles of commonly employed sampling boards. And, four typical failure types, namely, sampling harness breakage, equalization loop closure, filter capacitor breakdown and voltage regulator diode breakdown, are chosen as the research focus and analyzed in combination with real-vehicle fault data. Fault injection experiments were carried out for each type, and corresponding physical simulation models were established using MATLAB/Simscape software. These models were utilized to analyze the occurrence mechanisms of data anomalies, and the corresponding mathematical empirical models were derived for different fault types. After this, based on the understanding of the distribution pattern of fault data, a sampling fault detection algorithm with a threshold was developed. Since the threshold-based algorithm is unable to achieve classification of different fault types, a ResNet–BiLSTM algorithm was developed for sampling fault diagnosis, considering the spatial distribution and time-dependent characteristics of fault data. A dataset containing 3500 samples was generated for algorithm training and testing using the summarized fault mathematical model. Ablation experiments were devised to demonstrate the efficacy of the algorithm proposed in this paper. The test results reveal an accuracy of 98.86% and a KIA of 0.8917. These findings indicate the potential suitability of this algorithm in practical applications. To validate the performance of the algorithm in real-world scenarios, we established a real-vehicle dataset, which contains 10 fragments each for normal operation, sampling harness breakage and equalization loop closure, respectively. The ResNet–BiLSTM algorithm demonstrates an accuracy of 96.67% and a KIA of 0.9450, indicating its potential in practical applications.

The research on sampling faults in this paper follows the steps of ''test–model–algorithm–validation''. The fault diagnosis algorithm developed can realize the accurate detection and classification of sampling faults in the cloud platform without increasing hardware redundancy. Furthermore, it can also better support the BMS in accomplishing more complex tasks such as state estimation and thermal runaway warning.

Conflicts of interest

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

Data availability

The data that support the findings of this study are available from the corresponding author, Zhengjie Zhang, upon reasonable request: detailed explanations of the sampling board hardware circuit topology and sampling fault injection locations, physical diagrams and physical model structures. See DOI: https://doi.org/10.1039/d5ya00093a

Acknowledgements

The authors acknowledge financial support from the National Key R&D Program of China under Grant 2021YFF0601100.

References

1 W. Vermeer, G. R. Chandra Mouli and P. Bauer, A Comprehensive Review on the Characteristics and Modeling of Lithium-Ion Battery Aging, *IEEE Trans. Transp. Electrif.*, 2022, **8**, 2205–2232, DOI: 10.1109/TTE.2021.3138357.

2 S. Yang, Z. Zhang, L. Zhang, H. Yu, K. Yang and X. Liu, CHAIN: Cyber hierarchy and interactional network, *eTransportation*, 2023, **17**, 100256, DOI: 10.1016/j.etran.2023.100256.

3 R. Xiong, X. Sun, X. Meng, W. Shen and F. Sun, Advancing fault diagnosis in next-generation smart battery with multi-dimensional sensors, *Appl. Energy*, 2024, **364**, 123202, DOI: 10.1016/j.apenergy.2024.123202.

4 X. Feng, M. Ouyang, X. Liu, L. Lu, Y. Xia and X. He, Thermal runaway mechanism of lithium ion battery for electric vehicles: A review, *Energy Storage Mater.*, 2018, **10**, 246–267, DOI: 10.1016/j.ensm.2017.05.013.

5 D. Li, P. Liu, Z. Zhang, L. Zhang, J. Deng, Z. Wang, D. G. Dorrell, W. Li and D. U. Sauer, Battery Thermal Runaway Fault Prognosis in Electric Vehicles Based on Abnormal Heat Generation and Deep Learning Algorithms, *IEEE Trans. Power Electron.*, 2022, **37**, 8513–8525, DOI: 10.1109/TPEL.2022.3150026.

6 L. Jiang, Z. Deng, X. Tang, L. Hu, X. Lin and X. Hu, Data-driven fault diagnosis and thermal runaway warning for battery packs using real-world vehicle data, *Energy*, 2021, **234**, 121266, DOI: 10.1016/j.energy.2021.121266.

7 J. Zhang, Y. Wang, B. Jiang, H. He, S. Huang, C. Wang, Y. Zhang, X. Han, D. Guo, G. He and M. Ouyang, Realistic fault detection of li-ion battery *via* dynamical deep learning, *Nat. Commun.*, 2023, **14**, 5940, DOI: 10.1038/s41467-023-41226-5.

8 C. Li, K. Zeng, G. Li, P. Chen and B. Li, A novel fault diagnosis method for battery energy storage station based on differential current, *Appl. Energy*, 2023, **352**, 121970, DOI: 10.1016/j.apenergy.2023.121970.

9 Y. Liu, T. Dillon, W. Yu, W. Rahayu and F. Mostafa, Missing Value Imputation for Industrial IoT Sensor Data With Large Gaps, *IEEE Internet Things J.*, 2020, **7**, 6855–6867, DOI: 10.1109/JIOT.2020.2970467.

10 G. Li, Y. Hu, H. Chen, H. Li, M. Hu, Y. Guo, S. Shi and W. Hu, A sensor fault detection and diagnosis strategy for screw chiller system using support vector data description-based D-statistic and DV-contribution plots, *Energy Build.*, 2016, **133**, 230–245, DOI: 10.1016/j.enbuild.2016.09.037.

11 H. Wang, M. Liserre, F. Blaabjerg, P. De Place Rimmen, J. B. Jacobsen, T. Kvisgaard and J. Landkildehus, Transitioning to Physics-of-Failure as a Reliability Driver in Power Electronics, *IEEE J. Emerging Sel. Top. Power Electron.*, 2014, **2**, 97–114, DOI: 10.1109/JESTPE.2013.2290282.

12 C. Bülte, M. Kleinebrahm, H. Ü. Yilmaz and J. Gómez-Romero, Multivariate time series imputation for energy data using neural networks, *Energy AI*, 2023, **13**, 100239, DOI: 10.1016/j.egyai.2023.100239.

13 J. Kullaa, Detection, identification, and quantification of sensor fault in a sensor network, *Mech. Syst. Signal Process.*, 2013, **40**, 208–221, DOI: 10.1016/j.ymssp.2013.05.007.

14 G. J. Kacprzynski, A. Sarlashkar, M. J. Roemer, A. Hess and B. Hardman, Predicting remaining life by fusing the physics of failure modeling with diagnostics, *JOM*, 2004, **56**, 29–35, DOI: 10.1007/s11837-004-0029-2.

15 W. Li, J. Zhu, Y. Xia, M. B. Gorji and T. Wierzbicki, Data-Driven Safety Envelope of Lithium-Ion Batteries for Electric Vehicles, *Joule*, 2019, **3**, 2703–2715, DOI: 10.1016/j.joule.2019.07.026.

16 Y. Wang, G. Sun and Q. Jin, Imbalanced sample fault diagnosis of rotating machinery using conditional variational auto-encoder generative adversarial network, *Appl. Soft Comput.*, 2020, **92**, 106333, DOI: 10.1016/j.asoc.2020.106333.

17 A. R. Neuhaus, W. F. Rieder and M. Hammerschmidt, *Influence of arc duration and current on contact welding in low power switches*, in Proceedings of the Forty-Eighth IEEE Holm Conference on Electrical Contacts, IEEE, Orlando, FL, USA, 2002, pp. 17–20, DOI: 10.1109/HOLM.2002.1040817.

18 K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen and M. Srivastava, Sensor network data fault types, *ACM Trans. Sens. Networks*, 2009, **5**, 1–29, DOI: 10.1145/1525856.1525863.

19 D. Li, Y. Wang, J. Wang, C. Wang and Y. Duan, Recent advances in sensor fault diagnosis: A review, *Sens. Actuators, A*, 2020, **309**, 111990, DOI: 10.1016/j.sna.2020.111990.

20 K. Ma, H. Wang and F. Blaabjerg, New Approaches to Reliability Assessment: Using physics-of-failure for prediction and design in power electronics systems, *IEEE Power Electron. Mag.*, 2016, **3**, 28–41, DOI: 10.1109/MPEL.2016.2615277.

21 P. P. Shinde and S. Shah, *A Review of Machine Learning and Deep Learning Applications*, in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, Pune, India, 2018, pp. 1–6, DOI: 10.1109/ICCUBEA.2018.8697857.

22 K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, in 2016 IEEE Conference on Computer

Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016, pp. 770–778, DOI: 10.1109/CVPR.2016.90.

23 K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv*, 2015, preprint, arXiv:1409.1556, DOI: 10.48550/arXiv.1409.1556 .

24 J. Chung, C. Gulcehre, K. H. Cho and Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, *arXiv*, 2014, preprint, arXiv:1412.3555, DOI: 10.48550/arXiv.1412.3555.

25 S. Siami-Namini, N. Tavakoli and A. S. Namin, *The Performance of LSTM and BiLSTM in Forecasting Time Series*, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, Los Angeles, CA, USA, 2019, pp. 3285–3292, DOI: 10.1109/BigData47090.2019.9005997.

26 G. Van Houdt, C. Mosquera and G. Nápoles, A review on the long short-term memory model, *Artif. Intell. Rev.*, 2020, **53**, 5929–5955, DOI: 10.1007/s10462-020-09838-1.

27 J. Zhao, X. Mao and L. Chen, Speech emotion recognition using deep 1D & 2D CNN LSTM networks, *Biomed. Signal Process. Control*, 2019, **47**, 312–323, DOI: 10.1016/j.bspc.2018.08.035.

28 M. Ouyang, M. Zhang, X. Feng, L. Lu, J. Li, X. He and Y. Zheng, Internal short circuit detection for battery pack using equivalent parameter and consistency method, *J. Power Sources*, 2015, **294**, 272–283, DOI: 10.1016/j.jpowsour.2015.06.087.

29 J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers and A. W. M. Smeulders, Selective Search for Object Recognition, *Int. J. Comput. Vis.*, 2013, **104**, 154–171, DOI: 10.1007/s11263-013-0620-5.

30 Z. Zhang, R. Cao, Y. Jin, J. Lin, Y. Zheng, L. Zhang, X. Gao and S. Yang, Battery leakage fault diagnosis based on multi-modality multi-classifier fusion decision algorithm, *J. Energy Storage*, 2023, **72**, 108741, DOI: 10.1016/j.est.2023.108741.