



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/236832/>

Version: Published Version

Article:

Lambert, P., Drummond, R., Ross, J.P. et al. (2026) Detecting faulty lithium-ion cells in large-scale parallel battery packs using current distributions. *Communications Engineering*, 5, 17.

<https://doi.org/10.1038/s44172-025-00543-x>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:
<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.





Detecting faulty lithium-ion cells in large-scale parallel battery packs using current distributions



Pierre Lambert ^{1,2}, Ross Drummond ³✉, Joseph P. Ross ^{1,4}, Eloise C. Tredenick ^{1,5,6}, David A. Howey ^{1,6} & Stephen R. Duncan ^{1,6}

One of the main concerns affecting the uptake of battery packs is safety, particularly with respect to fires caused by cell faults. Mitigating possible risks from faults requires advances in battery management systems and an understanding of the dynamics of large packs. To address this, a machine learning classifier based upon a support vector machine was developed that detects cell faults within large packs using a limited number of current sensors. To train the classifier, a modelling framework for parallel-connected packs is introduced and shown to generalise to Doyle-Fuller-Newman electrochemical models. The fault classification performance was found to be satisfactory, with an accuracy of 83% using current information from only 27% of the cells. Validation on experimental pack data is also shown. These results highlight the potential to combine mathematical modelling and machine learning to improve battery management systems and deal with the complexities of large packs.

Applications including transport electrification and grid energy storage are leading a rapid growth in the roll-out of large lithium-ion (Li-ion) battery packs¹. As large packs become more widely adopted, concerns have been raised about their safety and the ability of the battery management systems (BMS) to detect faults that could eventually lead to fires^{2,3}. Various faults can occur during operation, mainly due to ageing and improper use⁴, and several instances of electric vehicle (EV) fires caused by battery pack faults have now been reported⁵. For example, the National Transportation Safety Board reported 17 lithium-ion battery fires on Tesla vehicles in the USA in 2018, out of a total of 350,000 vehicles⁶. Detecting these faults and isolating them before they become dangerous is therefore a growing concern for EV and grid storage users and manufacturers. Each Li-ion battery pack typically has its own individual architecture (with little standardisation across the field) because each application generally has unique energy and power requirements. Cells in battery packs are typically connected in parallel within individual modules, with the modules then connected in series⁷. This arrangement is often referred to as $nPmS$, where m modules of n parallel cells are connected in series. Within parallel-connected modules, both external and internal faults can occur^{8,9}. External faults are primarily caused by sensor and connection issues, while the causes of internal faults include overcharge, over-discharge, short circuits, accelerated degradation, and thermal runaway¹⁰; these form a varied collection of issues that can be

challenging to detect since each cell behaves as a 'black-box'. While the initiation of external faults can be sudden, internal faults often appear gradually, and thus their formation can be monitored and detected more effectively. The ability to detect and manage the development of internal faults (even though doing so may be challenging using existing pack sensor technology) motivates the fault-detection algorithms of this work.

Problems associated with battery pack faults, including fires and deterioration in pack performance, have motivated BMS fault-detection research. Cells in large-scale battery packs have intrinsic variability caused by slight changes in manufacturing and usage¹¹. Within parallel-connected packs, this cell-to-cell variability causes an uneven distribution of currents across the branches in the pack, resulting in the appearance of fluctuations¹²⁻¹⁴. As the behaviour of these current fluctuations depends upon the cell-to-cell variability, current measurements can be used to understand pack health. Since current sensors are already widely deployed for state-of-charge estimation (although usually not all parallel branch currents are measured) and are relatively inexpensive, they have the potential to be scalable, cost-effective, and accessible tools for diagnosing pack health and detecting faults. However, improvements in BMS algorithms are needed to make more effective use of this data.

In response, this paper develops a fault-detection method that uses the information encoded within the uneven current distributions of parallel-

¹Department of Engineering Science, University of Oxford, Oxford, UK. ²École Centrale de Nantes, Nantes, France. ³School of Electrical and Electronic Engineering, University of Sheffield, Sheffield, UK. ⁴Brill Power Limited, Oxford, UK. ⁵Faculty of Science and Technology, University of Canberra, Bruce, ACT, Australia. ⁶The Faraday Institution, Quad One, Harwell Campus, Didcot, UK. ✉e-mail: ross.drummond@sheffield.ac.uk

connected packs. First, a model of a large battery pack is established, and the branch currents simulated. An internal cell fault-detection algorithm based upon the support vector machine (SVM) is then proposed using the simulated data. It should be noted that the use of branch current distributions as a means of detecting cell faults has recently been demonstrated by Ding et al.¹⁵, whose work estimated the current distributions based on terminal voltage, total pack current and state-of-charge (SoC) using a long short-term memory (LSTM) network. They compared the estimated branch currents to the actual measured signals and used a generated residual signal to detect connection faults. The main differences with our work include the fact that Ding et al.¹⁵ proposed a detection method for connection faults (external faults) whereas we study accelerated degradation faults (internal faults); also, they focus on small packs (4p1s) whereas our results are based on large packs (74p1s).

The modelling framework and analysis proposed here also differs. In particular, our framework expresses the branch currents of the parallel pack explicitly in terms of the model states and the applied current. This formulation allows the differential algebraic equations (DAEs) of the pack model to be converted into a set of ordinary differential equations (ODEs), significantly simplifying the pack model equations. The obtained expressions for the branch currents are also shown to generalise to more complex battery models than the circuit models developed for the fault-detection algorithm, in particular to the Doyle-Fuller-Newman (DFN) electrochemical model based on porous-electrode theory. As far as the authors are aware, these are amongst the first simulations of parallel connected DFN models, a result that will enable detailed electrochemical simulations of Li-ion battery packs in the future.

Three main approaches have been used to detect battery faults in previous works: model-based methods, signal processing, and knowledge-based methods¹⁶. Model-based methods rely on battery models (mostly electrochemical or equivalent circuit) whose parameters are estimated using system identification techniques^{17–19}. The measured and estimated signals are compared and a residual signal is extracted to diagnose a potential fault²⁰. Signal processing methods rely on large datasets from which fault features are extracted (e.g., using entropy and wavelet transforms)^{21,22}; faults are detected once a given Z-score threshold has been exceeded for a certain feature. Finally, the knowledge-based approach involves applying methods such as expert systems, fuzzy logic, and neural networks to diagnose battery faults^{23–25}. Criteria describing the faulty state of the battery are established and thresholds defining the detection of a fault are determined.

The need for effective fault detection algorithms for BMS designs has motivated several studies on this topic. Tran et al.²⁶ proposed a model-based sensor fault diagnosis method and demonstrated how the equivalent circuit model parameters were affected by cell degradation and by sensor faults, with the proposed method able to detect and isolate voltage and current sensor faults from the estimated cell degradation. Nuhic et al.²⁷ developed a data-driven approach for health diagnosis using a SVM, which was shown to effectively learn the degradation behaviour of Li-ion cells. Yao et al.²⁸ also demonstrated the ability of a SVM classifier to identify a cell fault and give an indication of its severity. Finally, Sidhu et al.²⁹ proposed a model-based method to diagnose multiple faults using extended Kalman filters to represent signature-fault models.

Building on these results, the primary contributions of this article are as follows. An equivalent circuit model of a parallel-connected battery pack is developed and used to simulate single parallel modules (74p1s) within a Tesla Model S battery pack (74p96s) containing Panasonic NCR 18650B cells. An expression for the parallel-pack branch currents is derived which converts the model DAEs into ODEs. This expression for the currents is shown to hold for a broad class of cell-level models, in particular for parallel-connected DFN electrochemical battery models. The current distributions from the 74p1s pack model simulations are then analysed and their response to cell faults caused by accelerated degradation is investigated. A cell fault-detection algorithm based on a support vector machine is then trained and applied. The trained classifier was evaluated on experimental pack data, and, even though the algorithm was trained on simulation data, its ability to

predict faults transferred over to experimental data. These results highlight the potential to use relatively inexpensive sensing hardware—current measurements at pack level—for fault diagnosis, improving pack safety and performance.

Results and discussion

Simulations

We now describe our model for parallel-connected Li-ion cells that captures the impact of variations in the series resistance r and capacity Q on the overall response of the pack. The key feature is the resolving of Kirchhoff's laws such that the DAEs of the pack model are converted into ODEs, making it significantly simpler to solve. The class of cell-level models to which the following pack-level modelling results can be applied are those where voltage can be expressed as

$$V_k(t) = h(x_k(t)) + r_k i_k(t), \quad k = 1, 2, \dots, n \quad (1)$$

where the index $k \in \{1, 2, \dots, n\}$ relates to the parallel-connected cell number in the pack, n is the total number of cells in parallel, $x_k(t)$ is the dynamic state of cell k associated with a time derivative, r_k is the respective cell series resistance and $i_k(t)$ is the branch current (as in the current flowing into branch k of the parallel connections). Connecting these cell-level models in parallel means that Kirchhoff's laws have to be satisfied to compute the branch currents $i_k(t)$,

$$h(x_j(t)) + r_j i_j(t) = h(x_k(t)) + r_k i_k(t), \quad j, k \in \{1, 2, \dots, n\}, \quad (2a)$$

$$\sum_{k=1}^n i_k(t) = I(t). \quad (2b)$$

Combining these with the cell dynamics means that parallel-connected pack models are described by a set of DAEs³⁰. However, using the approach outlined in the Methods section, the algebraic equations of Kirchhoff's laws from Equation 2 can be resolved to give

$$i_1(t) = \left(r_1 \sum_{k=1}^n \frac{1}{r_k} \right)^{-1} \left(\sum_{k=2}^n \frac{\Delta h_{k1}(t)}{r_k} + I(t) \right), \quad (3a)$$

$$i_k(t) = \frac{1}{r_k} (r_1 i_1(t) - \Delta h_{k1}(t)), \quad k = 2, 3, n, \quad (3b)$$

where $\Delta h_{jk}(t) = h(x_j(t)) - h(x_k(t))$. With this expression, the branch currents can be written as a function of the states $x_k(t)$ (which, for equivalent circuit models such as Fig. 1, could be the state-of-charge and voltages, and, for electrochemical models, could be the potentials and concentrations) and the applied pack current $I(t)$. Resolving Kirchhoff's laws using Equation 3 means that the parallel-pack model does not have to numerically solve Equation 2 at each time step of the simulation to compute the branch currents $i_k(t)$. Instead, these currents are defined by Equation 3 and are expressed in terms of the states of the cell model $x_k(t)$ and the applied pack current $I(t)$, allowing the pack model dynamics to be solved as an ODE rather than a DAE³⁰. Turning the DAE parallel-pack model into an ODE also gives insight into the distribution of currents between branches in the pack, in contrast to the numerical approach where the computational solution only gives a limited insight into the mechanisms by which the pack's currents re-distribute themselves. The numerical solution for computing the parallel-pack branch currents is, however, widely used in modelling studies^{31–34}, estimator designs^{35–38} and simulations based upon the PyBAMM³⁹ open-source software⁴⁰.

The extent to which this computational solution has been used indicates the broad applicability of the analytic solution of Equation 3 to understand these nonlinear parallel-pack dynamics. Moreover, the proposed approach overcomes some of the restrictive assumptions of existing

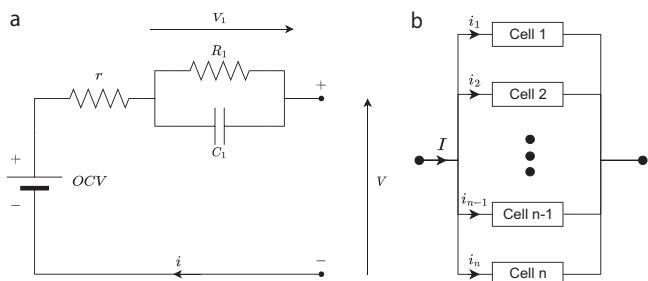


Fig. 1 | Schematic of the cell-level battery model used to build the packs. a Equivalent circuit model and **b** module of n cells connected in parallel.

Table 1 | Mean values of the cell model parameters

Parameter	Mean Value	Unit	References
r	19	$\text{m}\Omega$	61
R_1	1.7	$\text{m}\Omega$	61
C_1	5598	F	61
Q	3350	mAh	62

Table 2 | Standard deviations of the cell model parameters

Parameter	σ/μ ratio	Standard deviation σ	Unit
r	2.09 %	0.40	$\text{m}\Omega$
R_1	1.64 %	0.028	$\text{m}\Omega$
C_1	7.12 %	399	F
Q	0.28 %	9.4	mAh

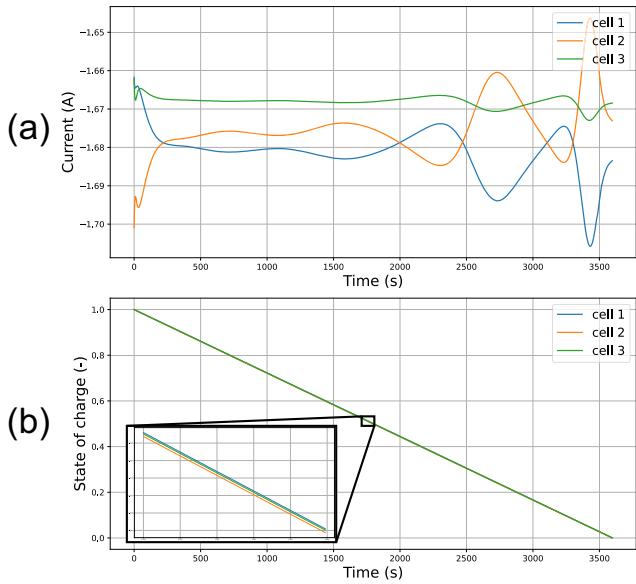


Fig. 2 | Discharging behaviour of three cells connected in parallel. Simulated currents (a) and states of charge (b) for three parallel-connected cells during a 1C discharge.

analytical solutions, such as the open circuit voltage (OCV) being affine^{[41,42](#)} and the cell-level model simply being a voltage source and a series resistor^{[13,43](#)}. Instead, with Equation 3, the voltage is allowed to be of the form of Eq. (1), as in the sum of a linear series resistance term and a nonlinear function of the states. Compared to the existing derivation for the branch

currents from Drummond et al.^{[44](#)}, the method presented here is simpler since it does not involve computing a matrix inverse. Instead, in this work, the structure of the equations defining Kirchhoff's laws are exploited to solve Equation 2.

The cell-level model used to design the fault-detection algorithm for the parallel-pack branch currents is the equivalent circuit model described in the Methods section. The different cell parameters were assumed to be normally distributed from cell-to-cell by extrapolating the results of Schneider et al.^{[45](#)}, with the means of the parameter distributions given in Table 1 and the standard deviations given in Table 2. Since the values observed in Schneider et al.^{[45](#)} were based on different cells (Samsung INR18650-25R) from those considered here (Panasonic NCR 18650B), the σ/μ ratios from Schneider et al.^{[45](#)} were first computed and then related to the mean values from Table 1 to derive the standard deviation values for the NCR 18650B cell considered here. The standard deviations computed using this method are stated in Table 2.

To analyse the behaviour of our model, a 1C discharge of three cells connected in parallel was considered. The simulated currents and state-of-charge values are plotted in Fig. 2, and it is noted that the current distributions for the cells are similar in shape to those measured experimentally by Chang et al.^{[46](#)}. In both cases, the characteristic current fluctuations of parallel-connected packs are observed. It is the properties of these fluctuations that are examined in this work for the purpose of fault detection.

The architecture of a Tesla Model S battery was used as a reference for a large parallel-connected pack. This pack has a 74p96s configuration, but a single module (74p1s configuration) was simulated here. Because all modules are subjected to the same pack current, they all behave similarly, hence our focus on the simulation of a single module. The 74 cells of the module were assigned parameters according to the normal distribution defined previously (Tables 1 and 2), and the ODEs of the pack model were solved to obtain the terminal voltage, state-of-charge, and current flowing through each cell in parallel during a full discharge. The results of this pack-level simulation are shown in Fig. 3. The observed current fluctuations typically have an amplitude of a few hundredths of an ampere and are therefore detectable with shunt current sensors that typically have tolerances ranging from 0.1 to 1% and can be connected in series with the cells^{[14](#)}—although, in practice, cost reasons make such mass sensing impractical. The highest current deviations from the mean are located at the beginning and end of the discharge.

At the start of the simulation, the immediate distribution of currents across the pack is due to differences in series resistance from cell-to-cell causing different currents to flow through the cells such that they have the same terminal voltage. Because of this initial current distribution, the cells discharge at different rates. The values of the cell OCVs will then also vary from cell-to-cell, since the OCV is a function of the state-of-charge, causing the currents within the pack to rebalance themselves to enforce Kirchhoff's voltage laws. This rebalancing mechanism is the reason why the fluctuations are more visible at the end of the discharge where the gradient of the OCV with respect to SoC changes is steep. It should also be noted from Fig. 3 that the SoCs vary slightly between the cells and that the terminal voltage is the same throughout the parallel pack, which is consistent with the Kirchhoff law of Equation 2.

It should be recalled that the cells modelled during this study were considered to be fresh and from the same batch, implying little cell-to-cell variability (except for the faulty cells used for the fault detection algorithm that were deliberately selected to have a high variation so as to represent a fault). For comparison, a simulation of the same pack but with the fresh cells replaced with aged ones was run, as shown in Fig. 3. To characterise the aged cells, the value of the standard deviations of the parameters was multiplied by five with respect to the values chosen previously (see Table 2). The current variations follow a similar shape to the fresh pack simulation of Fig. 3, with one significant difference; the amplitude of the variations is one order of magnitude higher. This increased variation in the currents is responsible for the significantly higher SoC deviations of the aged pack when compared to the healthy pack results in Fig. 3, which are amplified further by

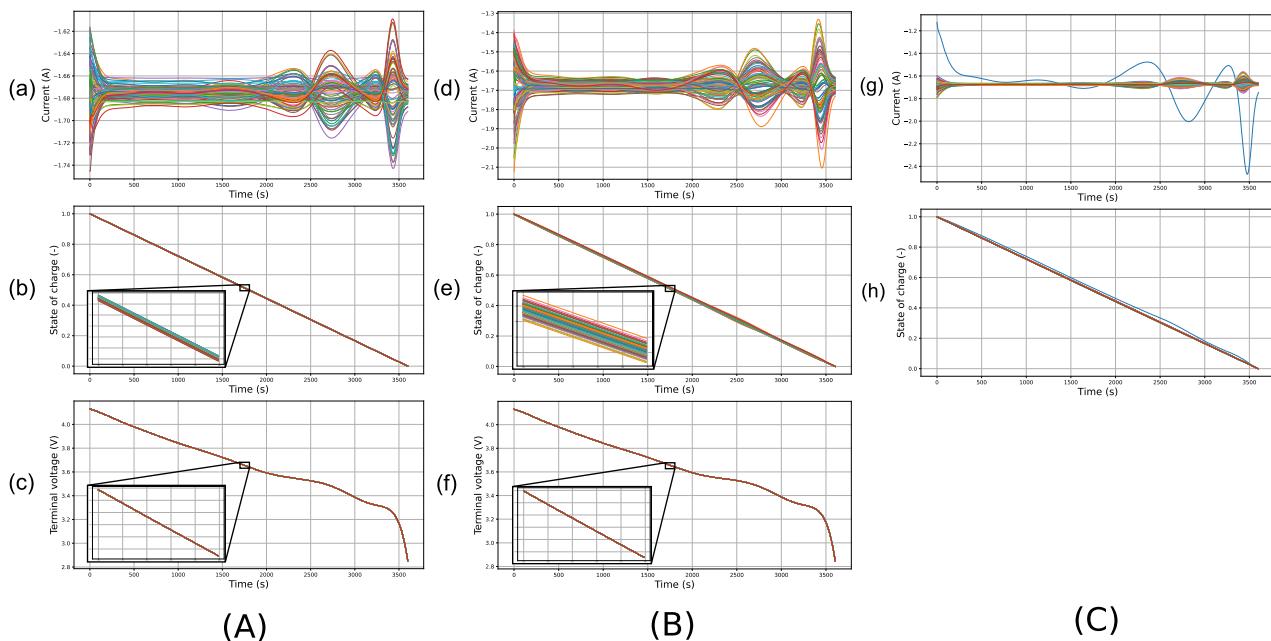


Fig. 3 | Simulated currents, state-of-charges, and voltages for 74 parallel-connected cells during a 1C discharge. A Fresh Tesla Model S module: currents (a), states of charge (b), and voltages (c). Each line corresponds to a cell. B Aged Tesla

Model S module: currents (d), states of charge (e), and voltages (f). Each line corresponds to a cell. C Pack with one faulty cell: currents (g) and states of charge (h), with the solid cyan line indicating the defective cell among 74 cells.

the increased variation in the capacitance of each cell. These capacitance differences also cause a larger shift in the current fluctuations over time; the current deviations of the cells with the highest capacitances have a slight delay compared to others, which is particularly noticeable at around 2800 s in the simulation. The simulations suggest that a degradation fault would be more easily identifiable in aged packs compared to fresh ones, as the current fluctuations would be greater. In the remainder of this paper, the focus will be on the design of fault-detection algorithms for the pack described above.

The general form of the cell-level voltage equations of Eq. (1) allows the equation for the branch currents given in Equation 3 to be applied to a broader class of models than just the circuit model discussed above. In particular, it can be applied to DFN-style electrochemical Li-ion battery models⁴⁷ with double layer effects included^{48,49}—a benchmark model for Li-ion batteries. Electrochemical battery models are, generally, more complex than circuit models and this added complexity introduces challenges when using them within pack models, as observed in the recent work of Reniers and Howey⁵⁰. In that work, a pack-level model was developed for a 1 MWh grid battery system containing 18,900 cells; simulations then cycled the system for 10 years. The cell-level model of that study was the single particle model (SPM), and although this is one of the simplest forms of battery electrochemical models, the analysis highlighted the difficulty of resolving Kirchhoff's laws for parallel connections of SPMs. Specifically, due to the SPM nonlinear series resistance, the parallel Kirchhoff laws had to be resolved using an approach based upon a PID controller, which introduced some errors.

By contrast, it is shown in the Supplementary Information (SI1 giving the mathematical analysis and SI2 defining the DFN model variables and parameters) that resolving parallel connections of DFN models is intuitive and possibly easier to implement than for the SPM. Even though the DFN model is more complex than the SPM, this result shows how computing the branch currents is easier. As detailed in SI1, this is obtained by expressing the DFN model voltage within the anode, cathode, and separator in the form of Eq. (1) with

$$r = R_{\text{ctc}} + \int_{\Omega_n} \frac{1}{\sigma_s + \kappa_e} dx + \int_{\Omega_p} \frac{1}{\sigma_s + \kappa_e} dx + \int_{\Omega_s} \frac{1}{\kappa_e} dx,$$

and

$$h(x_k(t)) = \phi_{\text{dl}}(L, t) - \phi_{\text{dl}}(0, t) + \int_{\Omega_n} \frac{\omega \kappa_e}{\sigma_s + \kappa_e} \frac{\partial \ln(c_e(x, t))}{\partial x} - \frac{\sigma_s}{\sigma_s + \kappa_e} \frac{\partial \phi_{\text{dl}}(x, t)}{\partial x} dx \\ + \int_{\Omega_p} \frac{\omega \kappa_e}{\sigma_s + \kappa_e} \frac{\partial \ln(c_e(x, t))}{\partial x} - \frac{\sigma_s}{\sigma_s + \kappa_e} \frac{\partial \phi_{\text{dl}}(x, t)}{\partial x} dx + \int_{\Omega_s} \omega \frac{\partial \ln(c_e(x, t))}{\partial x} dx.$$

Crucially, by including double-layer dynamics in the DFN model, the potential $\phi_{\text{dl}}(x, t)$ becomes a model state as the double-layer dynamics give it a time derivative. The parallel branch currents can then be computed directly using Equation 3, eliminating the need to solve the algebraic equations of Kirchhoff's laws when using DFN models. Thus, even though the DFN is a more complex electrochemical model than the SPM, computing the branch currents for parallel connections is intuitive and can be readily implemented, with the branch currents expressed directly as a function of the model states and the applied current.

To verify that Kirchhoff's laws had been satisfied with this DFN pack model, simulations of two parallel-connected DFN electrochemical models were conducted. The parameters of these simulations are given in SI2. The model was formulated numerically by discretising the PDEs using second-order central differences to approximate spatial derivatives, along with averaging of the diffusivity and conductivity functions at the control volume faces, and solved using “ode15s” within MATLAB® 2021b⁵¹. The parameters for each cell in this pack model are given in Table 1 and Table 2. A constant-current discharge was modelled with a current density of $I(t) = 129.55 \text{ A/m}^2$. It was assumed that the contact resistance between cells one and two in the simulated parallel pack differed with $R_{\text{ctc}} = 5.7 \times 10^{-4} \Omega$ for cell one and $R_{\text{ctc}} = 7.7 \times 10^{-4} \Omega$ for cell two—this setup is referred to as case one. As well as differences between cell contact resistances, two other cases were simulated. Case two additionally has the anode reaction rate coefficient in cell two being three times that of cell one, while case three has the anode particle radius of cell two being three times greater than in cell one.

The simulation results for the three cases discussed above are shown in Fig. 4. In this figure, the first row describes the pack voltages, the second row is the branch currents, the third row is the evolution of the cathode reaction rate kinetics at both the current collector and the separator boundaries, and the fourth row corresponds to the evolution of the spatial distribution of the

reaction kinetics through the thickness of the anode. Each line in the fourth row figures corresponds to a snapshot of $j(x, t)$ in the anode taken every 25 s during simulation, with darker lines corresponding to the start of the simulation and lighter lines corresponding to the end. The three columns in the figure corresponds to the three different cases of the cell parameters being simulated.

From the current and voltage plots of Fig. 4, it can be observed that Kirchhoff's laws are satisfied; cell voltages are equal and the current splits between the two cells and constantly rebalances itself during the discharge. The three different cases were found to lead to three different responses, highlighting the degree to which manufacturing variability of cells (and hence, variation in their electrochemical parameters) can affect pack performance. The final two rows in the figure illustrate how variations in cell parameters can lead to variations in the electrochemical response. These variations may impact the pack response in the long-term, especially cell degradation rates⁵².

The generalisation to parallel-connected DFN models discussed above shows how the modelling framework of Equation 2 can be applied to more complex models, beyond simple electrical circuits. The framework can also be generalised to time-varying current profiles, such as the drive cycles considered in ref. 44, and is not restricted to the constant-current profiles used here for fault diagnosis. With this approach, detailed simulations of pack electrochemical response can be undertaken. However, due to the significant computational complexity and lack of effective parameter estimation methods for DFN models, it was decided to base the fault detection algorithm here on parameterised circuit models for 74p96s Tesla Model S modules. Once the computational and parameterisation issues of DFN models are resolved, the expression for the branch currents stated in Equation 3 can then be used for pack-level diagnostics with more complex models, such as designing advanced fault-tolerant algorithms.

Fault detection

To implement the cell fault-detection algorithm, a cell fault must first be introduced. Here, the focus is on understanding faults caused by accelerated degradation. Amongst other phenomena, as a cell ages its resistance increases—especially at end-of-life⁵³. Within a pack, some cells will reach this stage sooner than others, leading to potential accelerated degradation faults. For this reason, the resistances of the cells within a pack can be regarded as useful indicators of cell faults, as they can significantly influence the pack dynamics⁵². We therefore consider the detection of faults caused by an abnormal increase in the ohmic resistance of a cell due to accelerated degradation⁵⁴. According to the cell model described in Fig. 1, a cell whose series resistance r follows a normal distribution based upon the values given in Tables 1 and 2 will be defined as healthy. A faulty cell will be defined by an abnormally high resistance r (at least 5 standard deviations above the mean). Experimental studies⁵⁵ have shown that series resistances of cells within packs can vary by several dozen percent during ageing. For this reason, the following analysis will consider the case when the faulty cell has an increased resistance ranging from 10 to 100%.

To evaluate the impact of cell faults on the current distributions across parallel-connected packs, simulations of both a healthy pack (see Fig. 3) and one with a fault were carried out. The pack containing the fault was similar in all aspects to the healthy pack but one of the cells had an increased series resistance, $r = 1.5 \times \mu_r = 28.5 \text{ m}\Omega$. The results are shown in Fig. 3. There is a clear difference compared to the healthy pack in Fig. 3: the faulty cell exhibits a discharge current at $t = 0$ with a low absolute value due to its resistance fault. Consequently, the current variations experienced by this cell are extreme in amplitude. However, in general, the currents flowing through the non-faulty cells in this aged pack all behave similarly to those obtained for a healthy pack. This similarity between the currents in fresh and aged packs introduces a challenge for fault detection when not all the currents are being monitored.

A fault detection algorithm based upon a SVM classifier was trained on the dataset to perform binary classification between healthy and faulty

parallel packs based upon the simulated current distributions, with details of this algorithm given in the Methods section. Features for the SVM fault-detection algorithm were extracted from the dataset samples in order to discriminate the healthy packs from the faulty packs. The performance of the classifier was first verified for detecting faulty packs from all the simulated currents as inputs. Using simple features based on the deviation of currents from the mean, it was expected that faulty packs could be detected because they present an abnormal current signal (see fault detection section). Once trained, the classifier was able to classify the 200 samples of the training set without making any classification errors, thus achieving a 100% accuracy score. The algorithm thus performed successfully when all the currents in the pack were being monitored.

However, in practice, only a limited number of branch currents are typically measured with large parallel-connected packs due to the increased costs of the sensors and added system complexity. To make the proposed algorithm more applicable the fault detection algorithm for the sparse sensing problem is now considered—i.e., when only a fraction of the number of branch currents are measured. With sparse sensing, fault detection becomes significantly harder, as it is challenging to infer the impact of a faulty cell on the rest of the pack when only measuring the healthy-cell currents, as seen in Fig. 5. For this analysis, it was assumed that only N_s current sensors were deployed in the pack and the current from the faulty cell was not measured by these sensors.

Features for the developed SVM fault-detection algorithm were extracted from the dataset samples in order to discriminate the healthy packs from the faulty packs. The extracted features described in the Methods section were then used and an ablation study was performed to characterise their impact on the classification, where the impact of each feature on the classifier was quantified. After empirically testing and selecting features, it was noticed that the most effective descriptors were formed from the local minima and maxima of the current signals. The classification was performed for a range of numbers of sensors N_s in the parallel pack. The classifier was first trained and tested for $N_s = 73$, i.e. all cell currents available except the faulty cell current. The confusion matrix of the corresponding predictions is given in Table 3a. The same procedure was then applied for $N_s = 20$, i.e., sensors measuring only a subset of the 74 cells, with results given in Table 3b.

In practice, these measurements may not be available because current sensors are usually not added to every branch of a parallel-connected pack. In fact, often only a few sensors are used due to added costs and additional compute power required from the BMS to process the data. For these reasons, a fault-detection algorithm must be robust in the sense that it should still function even when the current of the parallel branch is not measured. The results in the following section address this issue.

The features were then trained on the dataset generated by this model, whose structure is shown in Fig. 6. The profile of the current distributions simulated within the dataset can be seen in this figure, showing that faulty packs can be easily distinguished from healthy packs using measurements of the faulty cell current (with the exception of the case $r_{\text{faulty cell}} = 1.1 \times \mu_r$). This again highlights the increased difficulty in designing the fault-detection algorithm for the sparse sensing problem. After calculating the features for all samples in the training and testing set, their histograms were analysed. These were calculated for $N_s = 73$ (i.e. the current signal from the faulty cell was removed in faulty packs, while a random current was removed for healthy packs). The feature values for the healthy and faulty packs were separated, with the resulting histograms compared in Fig. 7. The observed distributions for these features differ between healthy and faulty packs, except for f_1 .

The histograms reveal that the extrema reached by the currents, which form the basis of the six identified features f_1 to f_6 , allow one to partially discriminate the two classes. This is the case even though it appears that the current distributions of the cells in a healthy pack and the healthy cells in a faulty pack are similar (from Fig. 3 and the fault-detection algorithm results). It is this discrimination that allows the SVM

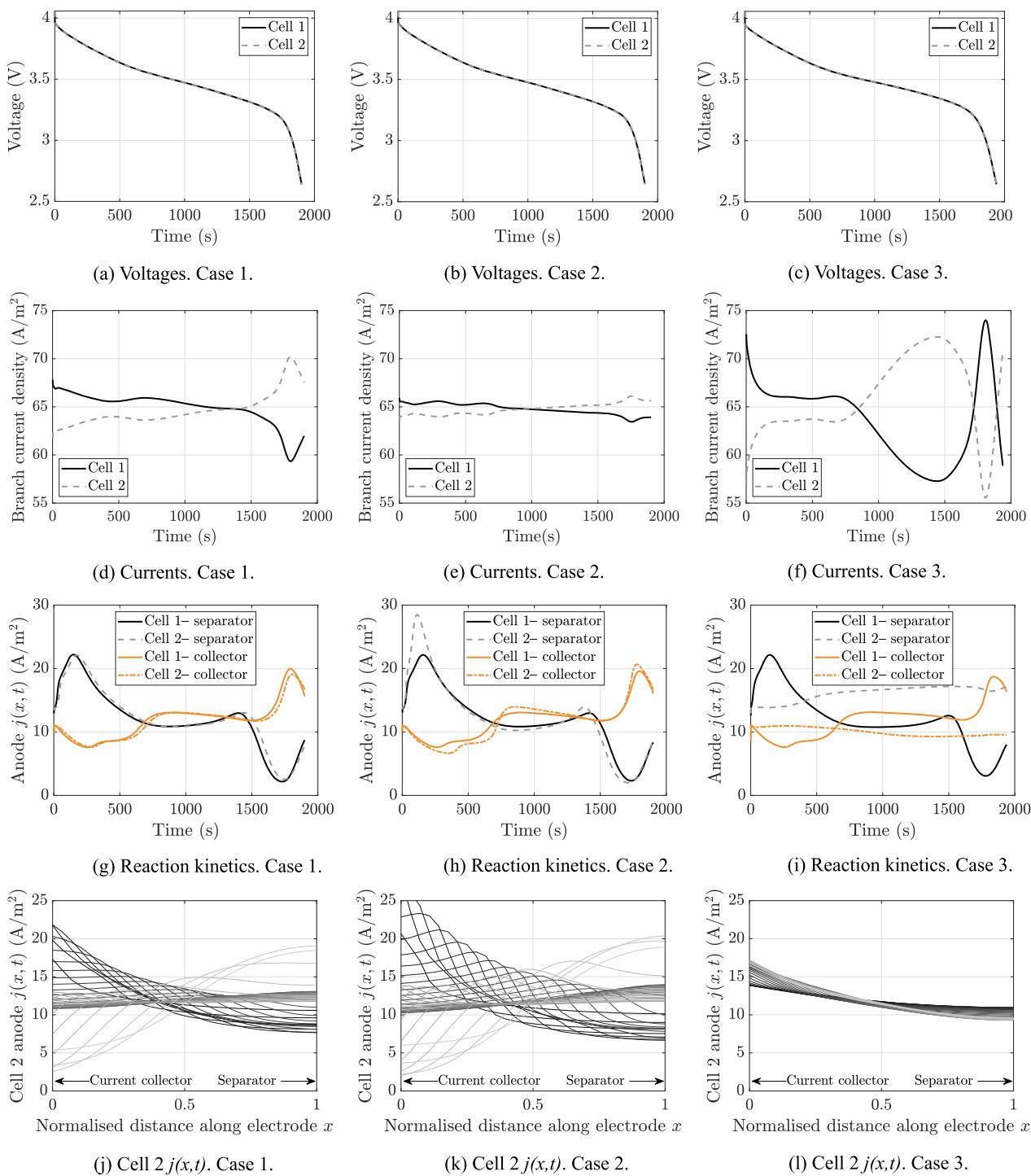


Fig. 4 | Discharge simulation results of two parallel-connected DFN models using the parameter values in Table 1 and Table 2. Case 1 corresponds to the nominal setup, with $R_{ctc} = 5.7 \times 10^{-4} \Omega$ for cell 1 and $R_{ctc} = 7.7 \times 10^{-4} \Omega$ for cell 2. Case 2 additionally has an anode reaction rate coefficient k three times greater in cell 2 than cell 1. Case 3 has an anode active particle radius three times greater in cell two than

cell one and the fourth row is plotted every 25 seconds, with dark lines corresponding to the start of the simulation and light lines to the end. **a–c** are the voltage responses for the three cases, **d–f** are the branch current densities, **g–i** are the anode reaction rates in time, and **j–l** are the distribution of the reaction rates in space across the anode.

algorithm to detect faults even when only limited sensor information is used. After empirically testing and selecting different features, it was also noticed that the most effective descriptors were formed from the local minima and maxima of the current signals. Finally, it is noted that as the number of removed current sensors increases, the distributions of the histograms of the two pack classes (healthy and faulty) overlap—resulting in reduced classification performance.

Finally, the SVM classifier was trained and tested for a larger number of N_s values, between 2 and 72, and the model accuracy evaluated for each trial. The results are shown in Fig. 8 and are a key contribution of this paper. This figure compares the accuracy of the proposed faulty pack detection algorithm as a function of the number of measured current branches N_s . A plateau can be seen in the prediction accuracy of the figure at around 20 sensors out of 74, continuing to around 50 sensors. This suggests that the

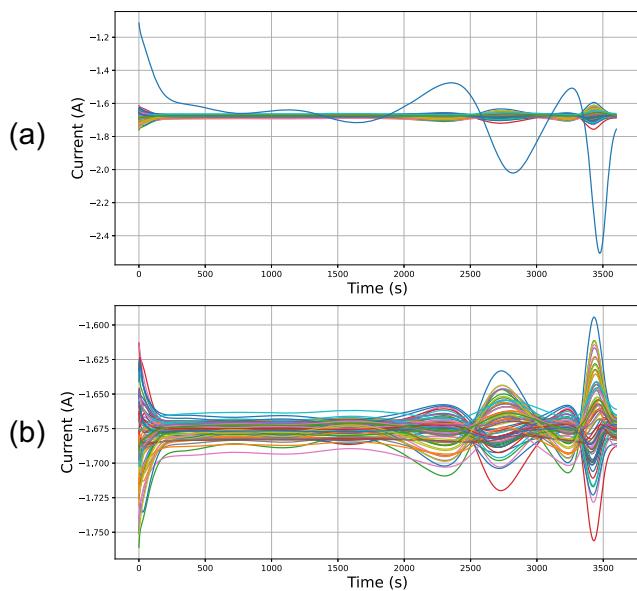


Fig. 5 | The impact of cell faults on pack-level current distributions. Comparison between the current distributions of a faulty pack (a) and the same pack without the faulty cell current (b) during a 1C discharge. Each line corresponds to a cell, with the solid cyan line in (a) corresponding to the defective cell.

Table 3 | Confusion matrices for two sensor configurations

		Prediction		Total
Truth	Faulty pack	Faulty pack	Healthy pack	
Faulty pack	81	17	98	
Healthy pack	5	97	102	
Total	86	114	200	

(a) Confusion matrix of SVM classifier with $N_s = 73$ (i.e. all but the faulty parallel branch currents are measured).

		Prediction		Total
Truth	Faulty pack	Faulty pack	Healthy pack	
Faulty pack	76	27	103	
Healthy pack	7	90	97	
Total	83	117	200	

(b) Confusion matrix of SVM classifier for $N_s = 20$.

optimal detection configuration for the Tesla Model S module, in terms of precision and sensing cost, is around 20 sensors. As the number of deployed sensors decreases, the accuracy score decreases: for 7 sensors (10% of the cells in the pack), an accuracy of 76% was obtained, and for 2 sensors only, it dropped to 67%.

Table 3b details the predictions made by the algorithm with only 20 sensors—the results contain few false positives, and a larger quantity of false negatives. Since an accelerated cell degradation fault in a pack is a relatively infrequent phenomenon, it is important that there are few false positives. However, false negatives can be problematic and it can be seen that this is the main difference compared to the $N_s = 73$ case discussed in Table 3a, where there are fewer false negatives. Considering the plot shown in Fig. 8, and the statistical nature of the extracted features used by the detection method, it appears that this method is applicable for a parallel pack made of a large number of cells (N_c , total number of cells in pack), such as the Tesla Model S module. However, for pack configurations where only a few cells are connected in parallel, a method that uses only a small fraction of the total number of cells in the module may prove difficult to implement.

The accuracy of the proposed SVM fault classifier was also evaluated against a recurrent neural network (RNN), since RNNs are widely used in many battery system algorithms, e.g., state-of-charge estimation⁵⁶ because they can implement online diagnosis in an end-to-end manner. Details of the RNN can be found in the Methods section. Training on the raw current data, the RNN had a fault-detection accuracy of 47.0% when $N_s = 73$ and 45.50% when $N_s = 20$, lower than the SVM. Whilst other neural network architectures may show higher accuracies, the results indicate that simple SVM classifiers can perform well for this task when compared against more complex deep learning methods. The simplicity of the SVM also brings advantages for practical deployment, both in terms of explainability of features but also compact size. However, deep learning approaches, such as the RNN considered here, will likely outperform SVMs for more diverse and complex usage profiles than the constant discharge currents of this paper. With rapidly changing use profiles, such as drive cycles, the SVM features may not be appropriate as it would be challenging to disentangle the fluctuations caused by the use profile and those caused by the imbalances across the parallel pack. Future work and experimental data will be required to validate this.

Several model parameters strongly influence the dynamic response of the current distributions across a parallel-connected pack and, consequently, the performance of the detection algorithm. For example, it was observed that the points where the fluctuations in the current distributions were largest, during both discharging and charging, correlated with the points where the slope of the OCV curves were steepest. The shape of the OCV curve (which is different for each cell chemistry) therefore strongly impacts the ability of the algorithm to detect cell faults. Additionally, it is noted that the pack ageing will influence the detection algorithm—the variation between cell parameters may grow, which could lead to greater current deviations and potentially improve the accuracy of the detection algorithm. It should also be noted that the obtained accuracy of 82.8% for $N_s = 20$ (i.e. sensors placed on about 27% of the cells in the pack) assumes that no sensor is ever placed on the faulty cell. Since this should happen in this configuration in 27% of cases and the fault is systematically detected in this case, we can assume a higher accuracy in the general case.

Experimental validation

To evaluate the practical applicability of the proposed approach, the SVD fault prediction classifier trained on 74p1s pack simulations with NCR 18650B cells was then applied to experimental testing data. Details of the experimental setup can be found in the Methods section. Figure 9a–d show the recorded current data of the four cells in the packs for the four cases. In order for the experimental current data to be in the same format as that of the SVD's training data (e.g., that shown in Fig. 6), the data was scaled according to the approach detailed in the Methods Section, with the normalised currents shown in Fig. 9e–h. Feeding this scaled experimental data into the SVD classifier trained on the 74p1s pack NCR 18650B cell simulation data gave the predictions of: (i) Fig. 9i–l with $N_s = 1$, and (ii) Fig. 9m–p with $N_s = 3$. In general, the SVD predictor performed relatively well on this experimental dataset. With $N_s = 1$, all the faulty cases were detected, but the healthy pack was also predicted to be faulty; with $N_s = 3$ (as in with only the faulty cell not being sensed), the SVD predictor detected the faults in Cases 1 and 3 but misclassified Case 2 during most of the discharge and Case 4 at some times. In general, these results highlight the transferability of machine learning algorithms for battery systems trained on simulation data to experimental data, with the SVD classifier in Fig. 9 performing satisfactorily even though it was trained on 74p1s pack simulation data with NCR 18650B cells and tested on experimental pack data of four RJ-LFP72174204-280 cells.

Conclusions

A method to detect cell faults in large parallel connected battery packs that combines model-based prediction with machine learning classification was

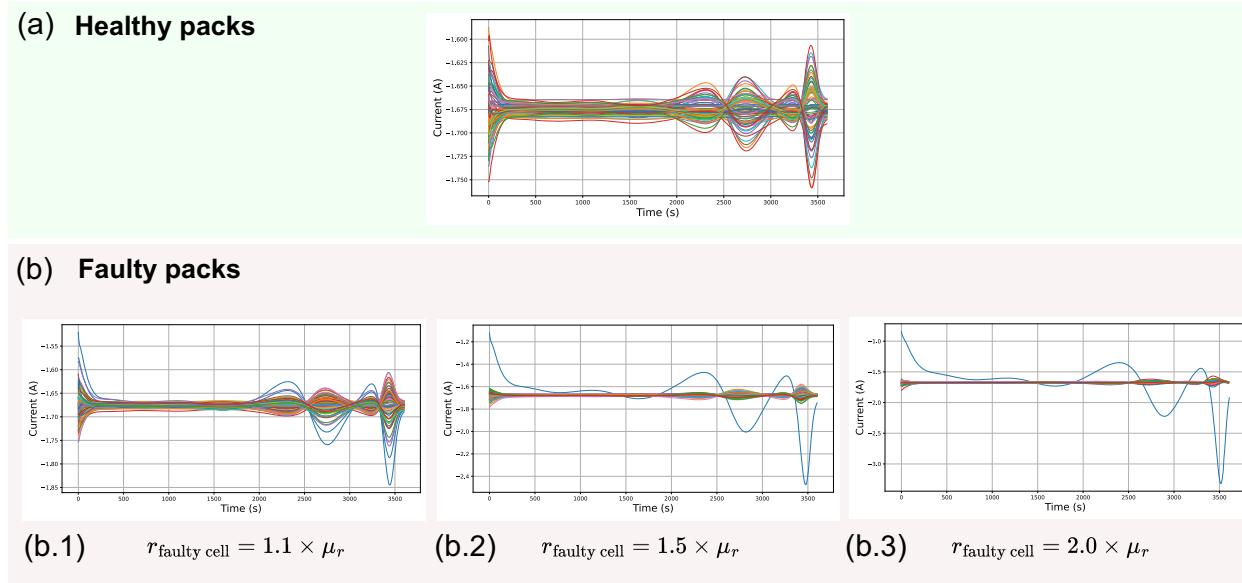


Fig. 6 | Change in pack-level current distributions as the faulty cell's resistance increases. Simulations of Tesla Model S battery pack module when all cells are healthy (a) and when one of the cells is faulty (b). The value of the faulty cell

resistance ($r_{\text{faulty cell}}$) varies between cases b.1, b.2 and b.3. It is defined as a multiple of pack mean resistance μ_r , and significantly impacts pack current distribution and hence the performance of the fault classification algorithm.

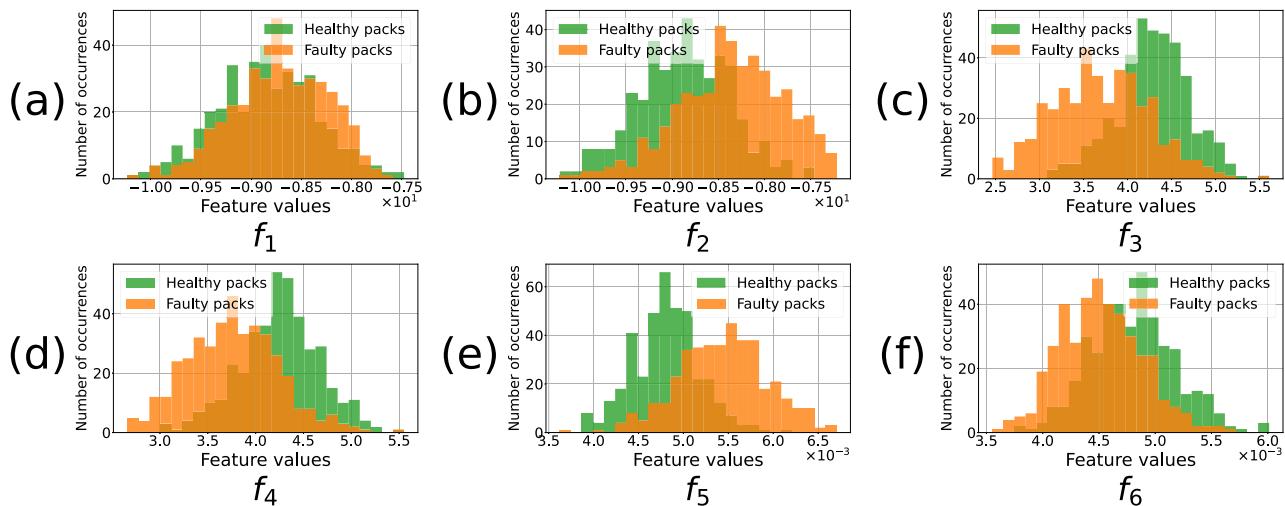


Fig. 7 | Histograms of feature for healthy and faulty packs of the simulated dataset. a gives the distribution of feature f_1 defined in (11), and similarly b in (12), c in (13), d in (14), e in (15), and f in (16).

presented. An equivalent circuit model of a parallel battery pack was developed to simulate the behaviour of an EV battery pack, specifically a Tesla Model S module. Current distributions within the parallel connected pack were studied in detail and an expression for the branch currents as a function of the model's states and the pack's applied current was derived. This expression for the parallel-pack branch currents was shown to generalise to more complex battery models, including the DFN electrochemical model. The branch currents of the parallel connected were used within an SVM-based algorithm to detect faulty cells in the pack to improve its safety. The proposed fault-detection algorithm was shown to perform well, achieving an accuracy of 83% when using only 20 current sensors per module of 74 cells in parallel. To assess its applicability in practice, the algorithm trained on simulation data was then testing on experimental pack data and performed well, with the results shown in Fig. 9. These results demonstrate the potential of detecting cell degradation faults with sparse sensing on large-scale parallel battery packs. By combining algorithms with

current sensing data in this way, these results can be used to improve the safety of a battery pack.

One limitation of this paper is the focus on faults from resistance increases in one of the pack's cells. A wider study including faults from both capacitance losses and impedance growth would result in a more comprehensive detection algorithm. The motivation to focus on resistance faults here was: (i) they could be replicated in experiments (see e.g. Fig. 9), (ii) these faults could occur quite suddenly, such as a tab becoming loose or lithium plating when fast charging in sub-zero temperatures, whereas slower degradation effects, such as capacitance losses from SEI layer growth, may be corrected by the averaging effect of the parallel connections, (iii) the higher resistances from the faults could lead to higher heat generation rates and so impact the pack's safety through thermal runaway. Generalising the results of this paper to the more diverse class of faults and current profiles found in practice is an important direction for future research in this area. For that generalisation, a rich-dataset containing a wide class of faults in

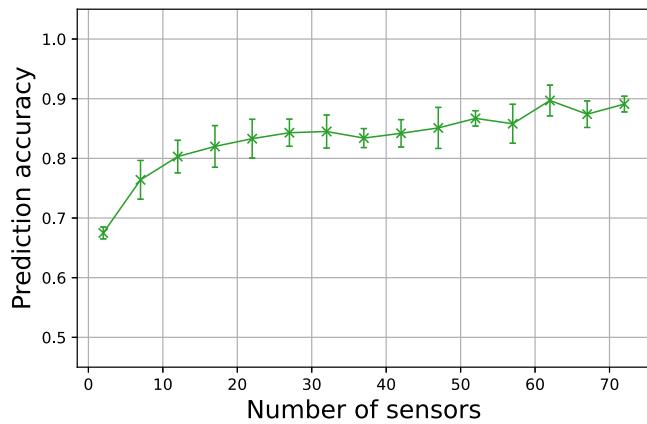


Fig. 8 | Accuracy score of fault-detection classifier (SVM) as a function of N_s (the number of current sensors within the 74p1s parallel pack). For each number of sensor plotted, 50 classifiers were trained, with the crosses representing the median accuracy and the bars representing the standard deviation of the accuracy across the tests.

practical battery systems would be required which would be challenging to achieve at scale.

As well as generalising the results to a more diverse class of faults, the authors are also comparing the efficacy of the proposed SVM algorithm with other approaches based upon deep learning, such as convolutional neural networks. As these algorithms have delivered strong results in other classification tasks, it is expected that they may also perform well for the problem considered here of detecting faults in large parallel connected packs. However, neural networks have limitations which may affect their performance in practice, notably, their lack of robustness may impact the extent to which engineers trust them for monitoring large and expensive packs, where safety issues need to be understood and failures explained.

Methods

Cell model

The pack model is based on the equivalent circuit model of a single cell and the connection of multiple cells in parallel. The cell model used is the Thevenin model shown in Fig. 1 which has been widely used due to its simplicity combined with its high accuracy⁵⁷. In this model, the dynamics for each cell $k \in \{1, 2, \dots, n\}$ are characterised by its series resistance r_k RC-pair

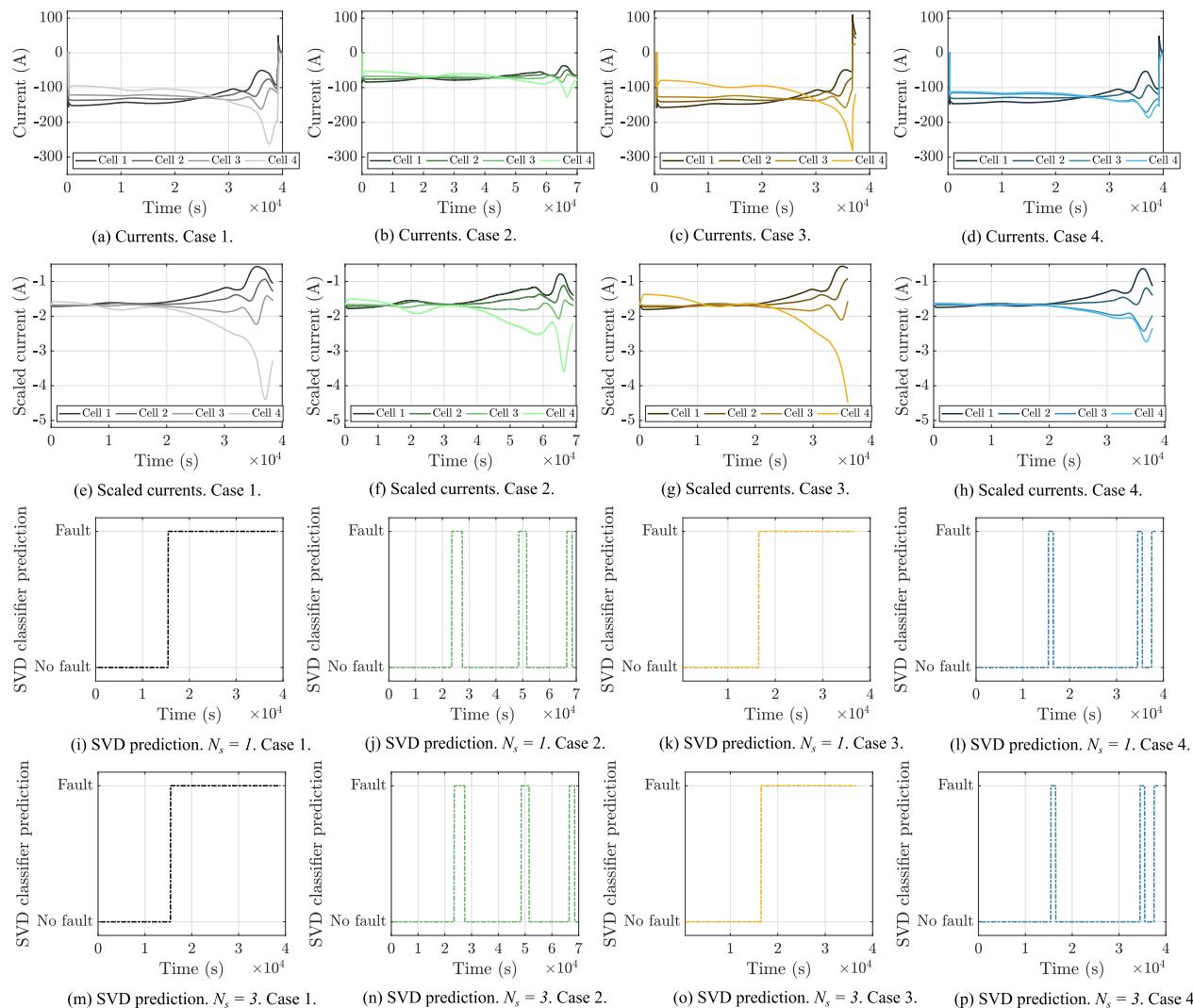


Fig. 9 | Application of SVD classifier to detect faults in experimental pack data with four RJ-LFP72174204-280 cells connected in parallel. As detailed in Methods Section, four fault cases were considered, with Cases 1–3 being faulty and Case 4 fault

free. **a–d** show experimental current data, **e–h** show scaled currents, **i–l** show fault predictions with $N_s = 1$ and **m–p** show fault prediction with $N_s = 3$.

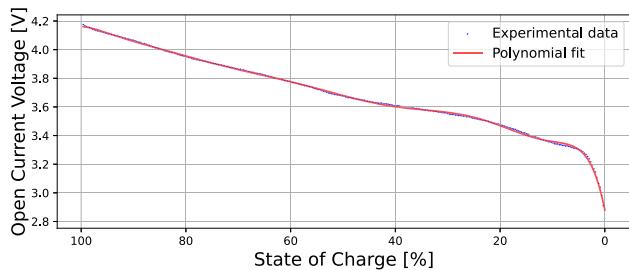


Fig. 10 | The open circuit voltage curve used for the Panasonic NCR 18650B cells. Comparison between the digitised OCV curve from ref. 58 (blue) and its polynomial interpolation (red).

resistance $R_{1,k}$ RC-pair capacitance $C_{1,k}$ and capacity Q_k . The subscript k is dropped from these parameter labels when referring to them in a general context, for example in Table 1 where their mean values are stated. The battery pack of a Tesla Model S was modelled and so the modelled pack's cells were chosen to be Panasonic NCR 18650B cells. The corresponding model parameters are given in Table 1 whilst the OCV curve was obtained from experimental data⁵⁸ and fitted using a polynomial approximation (see Fig. 10). The branch current $i_k(t)$ flowing through cell k was considered to be positive when charging and negative when discharging.

Within the pack model, the dynamics for cell $k \in \{1, 2, \dots, n\}$ were described by the simple equivalent circuit model

$$\begin{bmatrix} \dot{z}_k(t) \\ \dot{V}_{1,k}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_k C_k} \end{bmatrix} \begin{bmatrix} z_k(t) \\ V_{1,k}(t) \end{bmatrix} + \begin{bmatrix} 1/Q_k \\ 1/C_{1,k} \end{bmatrix} i_k(t), \quad (4a)$$

$$V(t) = OCV(z_k(t)) + V_{1,k}(t) + r_k i(t), \quad (4b)$$

where, for cell $k \in \{1, 2, \dots, n\}$, $V_{1,k}(t)$ is the voltage drop across the RC-pair of Fig. 1, $z_k(t)$ is the state-of-charge, $V(t)$ is the voltage and $OCV(z(t))$ is the open circuit voltage.

Parallel pack model: conversion of DAEs into ODEs

The parallel connection of n cells, as described in Fig. 1, was then modelled in accordance with Kirchhoff's laws. For parallel connections, these laws impose that the voltages $V(t)$ across each of the cells are the same and that the sum of the cell branch currents must equal the total current applied to the pack.

Defining $x_k(t)$ as the state vector of cell k , which for the circuit model is $x_k(t) = [z_k(t), V_{1,k}(t)]^\top$, and $i_k(t)$ as the current flowing through cell k in response to the pack current $I(t)$, then, for $j, k \in \{1, 2, \dots, n\}$, Kirchhoff's laws for parallel connected packs can be written as

$$OCV(z_j(t)) + V_{1,j}(t) + r_j i_j(t) = OCV(z_k(t)) + V_{1,k}(t) + r_k i_k(t), \quad (5a)$$

$$\sum_{k=1}^n i_k(t) = I(t). \quad (5b)$$

Combining Equation 4 and Equation 2 gives the system of DAEs of the parallel pack model. The use of DAE models is, however, often undesired because they can be significantly more challenging to simulate and analyse than those described by ODEs. In the following analysis, a method to translate this DAE model of the parallel pack into an ODE one is described. The first step of this conversion is to express Kirchhoff's laws as

$$V(t) = r_k i_k(t) + V_{1,k}(t) + OCV(z_k(t)) \quad (6a)$$

$$= r_k i_k(t) + h(x_k(t)), \forall k \in \{1, 2, \dots, n\}, \quad (6b)$$

where $h(x_k(t))$ is a function of cell k 's state-space. Kirchoff's laws can then alternatively be written as

$$r_1 i_1(t) - r_2 i_2(t) = \Delta h_{21}(t), \quad (7a)$$

$$\vdots \\ r_1 i_1(t) - r_n i_n(t) = \Delta h_{n1}(t), \quad (7b)$$

$$i_1(t) + i_2(t) + \dots + i_n(t) = I(t), \quad (7c)$$

where $\Delta h_{jk}(t) = h(x_j(t)) - h(x_k(t))$. With this formulation, each of the branch currents $i_k(t)$ can be expressed in terms of the first one, as in

$$i_2(t) = \frac{1}{r_2} (r_1 i_1(t) - \Delta h_{21}(t)), \quad (8a)$$

$$\vdots \\ i_n(t) = \frac{1}{r_n} (r_1 i_1(t) - \Delta h_{n1}(t)). \quad (8b)$$

Substituting these expressions back into Eq. (7c) gives

$$I(t) = i_1(t) + i_2(t) + \dots + i_n(t), \quad (9a)$$

$$= i_1(t) + \sum_{k=2}^n \frac{1}{r_k} (r_1 i_1(t) - \Delta h_{k1}(t)), \quad (9b)$$

$$= \left(1 + r_1 \sum_{k=2}^n \frac{1}{r_k} \right) i_1(t) - \sum_{k=2}^n \frac{\Delta h_{k1}(t)}{r_k}. \quad (9c)$$

The first branch current $i_1(t)$ can then be written exclusively in terms of the model's state and the applied current via

$$i_1(t) = \left(r_1 \sum_{k=1}^n \frac{1}{r_k} \right)^{-1} \left(\sum_{k=2}^n \frac{\Delta h_{k1}(t)}{r_k} + I(t) \right). \quad (10)$$

Using Equation 3, the other cell currents i_2, \dots, i_n can then also be written in terms of the model states and the applied current. This process allows the parallel pack model to be described by an ODE instead of a DAE, which greatly simplifies the analysis³⁰. The ODE model equations were then solved numerically using the `integrate.odeint` solver from the `scipy` package in Python.

Data processing pipeline

The faulty cell detection method is implemented through the following data processing pipeline:

1. Data Simulation. A large quantity of battery pack discharges, including both healthy and faulty packs, is simulated using the pack model developed around Equation 2 to generate a substantial number of current distributions.
2. Data Pre-processing. The simulated data is pre-processed to emulate the acquisition process of the current sensors used in practice. The data is then divided into a training set and a testing set.
3. Feature Extraction. Heuristic features are constructed from the pre-processed data to facilitate the classification of healthy and faulty battery packs.
4. Binary classification. A SVM is trained on the extracted features to perform binary classification, distinguishing between healthy and faulty battery packs.

Each pack was characterised by a set of 74 time signals (corresponding to the 74 cells in the Tesla Model S module). The generated dataset contained

1000 samples consisting of the simulated currents of the parallel pack. Each sample is represented as a matrix of size $74 \times t_{\max}$ with $t_{\max} \approx 3600$ s, being the duration of the pack discharge. A total of 500 healthy and 500 faulty packs were simulated. The healthy packs are assigned parameters following the normal distribution of the model for all 74 cells, while the faulty packs contain a faulty cell whose resistance $r_{\text{faulty cell}}$ has been set to be abnormally high (as defined at the beginning of the fault detection section). Different degrees of fault are considered during the simulations. Among the 500 simulated faulty packs, 50 are assigned the faulty cell resistance value $r_{\text{faulty cell}} = 1.1 \times \mu_r$, 50 are assigned the value $r_{\text{faulty cell}} = 1.2 \times \mu_r$, and so forth until $r_{\text{faulty cell}} = 2.0 \times \mu_r$.

Data pre-processing

The first step of the data pre-processing phase involves removing a number of current sensors from the samples of the data-set. By removing the current data of the faulty cell, the performance of the fault-detection algorithm was made more robust and more representative of packs deployed in practice, as costs may limit the number of branch currents being monitored. In fact, it was found to be straightforward to distinguish a healthy pack from a faulty one when all the current distributions were measured (i.e. when all the currents flowing through the cells were known, including that of the faulty cell). By deliberately removing the data generated by the faulty cell from the algorithm's input, the fault detection problem becomes non-trivial. If the current from the faulty cell is not measured, the behaviour of the current distributions appears to be similar to that of a healthy pack, can be seen from the results of Fig. 6. The method presented here thus aimed to perform faulty pack detection using only a fraction of the total potential current sensors. To do this, first, a given number of current signals, $N = N_c - N_s \geq 1$ where N_c is the total number of cells in the parallel pack and N_s is the number of sensors that are ultimately used, were removed from each sample of the dataset. This is equivalent to placing sensors on only a few of the 74 branches of the Tesla Model S's parallel pack module. For the faulty packs, the current sensor of the faulty branch is removed and the other removed current sensors are randomly selected.

The second step of the data pre-processing stage consisted in adding noise to the data and then filtering it. This noise was added to the model simulation data to make it more realistic with respect to the current sensor data measured from experiments. Given that the shunt resistors used for current sensing can generate a measurement error of approximately 0.1%¹⁴, a Gaussian noise with an amplitude of $\pm 0.1\%$ of the current signals was used. To be conservative, the standard deviation of the Gaussian noise distribution for each current signal was set to 0.05% of the mean value of the current during discharge. All the signals were then filtered by a low-pass Butterworth filter of order 5, and critical frequency $f_c = 0.005$ Hz (the sampling frequency being $f_s = 1$ Hz). These operations resulted in slightly distorted signals compared to the original data.

The third and last step of the data pre-processing stage is the train-test splitting of the data. The dataset was split into 80:20 proportions for the training and the testing set respectively; a 5-fold selection is applied for this purpose. These two sets contained samples consisting of matrices of size $N_s \times t_{\max}$. Features are then extracted from these samples and, after the feature transformation has been applied, each sample consists of a vector of length N_{features} . The standard technique of training an SVM classifier⁵⁹ was then applied to the datasets.

Feature extraction

Six features, labelled as f_1 to f_6 , were considered. Each sample consists of a matrix containing in each of its rows a branch current time-series. To be specific about individual time-series within those for the whole pack, the row index p is used to denote a given cell current within the parallel pack, and t will refer to the time index of the signal. The total number of cells is N_c and the duration of the time-series is t_{\max} , with a time step $\Delta t = 1$ s between each sample t . A given sample of the dataset is denoted by S and the value of the current in cell p at time t within the signal is $S[p, t]$.

1. Feature 1 is the average sum of local maxima over the cell currents across the whole discharge. Writing the local current maximum on a given time interval within a cell as $\max_{t \in S} S[p, t]$, and the k^{th} local maximum current encountered during its discharge (out of N local maxima) as $(\max_t S[p, t])_k$, the feature was then defined as:

$$f_1 = \frac{1}{N_c} \sum_{p=1}^{N_c} \sum_{k=1}^N (\max_t S[p, t])_k. \quad (11)$$

2. Feature 2 characterised the local current minima instead of the local maxima:

$$f_2 = \frac{1}{N_c} \sum_{p=1}^{N_c} \sum_{k=1}^N (\min_t S[p, t])_k. \quad (12)$$

3. Feature 3 uses the standard deviation instead of a mean:

$$f_3 = \sqrt{\frac{1}{N_c} \sum_{p=1}^{N_c} \left[\left[\sum_{k=1}^N \left(\max_t S[p, t] \right)_k - f_1 \right]^2 \right]}. \quad (13)$$

4. Feature 4 was defined similarly to f_3 but used the local minima:

$$f_4 = \sqrt{\frac{1}{N_c} \sum_{p=1}^{N_c} \left[\left[\sum_{k=1}^N \left(\min_t S[p, t] \right)_k - f_3 \right]^2 \right]}. \quad (14)$$

5. Feature 5 was based on the standard deviation of all local maxima within the signals (rather than taking their sum as in f_3) for a given p :

$$\{\max_t S[p, t]\} = \{(\max_t S[p, t])_0, \dots, (\max_t S[p, t])_N\}.$$

We define

$$\max S = \{\max_t S[1, t], \dots, \max_t S[N_c, t]\}$$

and denote its elements as $(\max S)_i$ which are numbered from 1 to N_m . Then, the feature, corresponding to the standard deviation of all local maxima within a sample was defined

$$f_5 = \sqrt{\frac{1}{N_m} \sum_{i=1}^{N_m} \left[[(\max S)_i - \overline{(\max S)}_i]^2 \right]}. \quad (15)$$

6. The final feature was defined as the standard deviation of all local minima within the signal S :

$$f_6 = \sqrt{\frac{1}{N_m} \sum_{i=1}^{N_m} \left[[(\min S)_i - \overline{(\min S)}_i]^2 \right]}. \quad (16)$$

SVM binary classification

The classification was performed by an SVM classifier⁶⁰ with hyperparameter optimisation using grid search. A 5-fold cross-validation was used to create a validation set from the test set obtained during the train-test split of the pre-processing phase. The hyperparameter grid was:

- $C: [0.01, 0.05, 0.1, 0.5, 1, 5, 10]$,
- $\gamma: [10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1, 1]$,
- Kernel: ['linear', 'rbf', 'poly', 'sigmoid'].

Table 4 | Four cases of packs in the experimental data

	Fault resistance	Discharge C-rate (Current)	Faulty pack
Case 1	125 $\mu\Omega$	0.45C (504 A)	✓
Case 2	125 $\mu\Omega$	0.25C (280 A)	✓
Case 3	250 $\mu\Omega$	0.45C (504 A)	✓
Case 4	0 $\mu\Omega$	0.45C (504 A)	✗

Here, C denotes the regularisation parameter, while the kernel function transforms the input data into another, often non-linear and high-dimensional feature space, and γ determines the influence of the training samples on the definition of the SVM separation hyperplane. The parameters retained after optimisation were $C = 1$, kernel = 'linear', and γ being used only for the 'rbf', 'poly' with 'sigmoid' kernels⁶⁰. The SVM predictions, y , were then obtained by

$$y = \text{sign}(w^T f + b) \quad (17)$$

with w being the weights and b the bias learned from data.

Recurrent neural network

The RNN was composed of an input layer, a Long Short Term Memory (LSTM) layer with 100 neurons, a fully connected layer, a softmax layer, and a classification layer. A learning rate of 0.001 was used with a gradient clipping threshold of 1 implemented with the ADAM optimisation algorithm. The inputs to the RNN were the current signals from each measured branch of the parallel pack and its output was a fault diagnosis. The number of epochs was 50 and the mini batch size was 32.

Experimental setup

Four 280 Ah prismatic LFP cells from RJ Energy (RJ-LFP72174204-280) were connected in parallel and current sensors recorded the branch currents. The sensors were IVT-S shunt sensors manufactured by Isabellenhütte with a ± 1000 A range and shunt resistor values of 20 $\mu\Omega$. Faults were introduced into Cell 4 of the packs through additional shunt resistors. As detailed in Table 4, four cases were considered with Cases 1–3 being faulty packs and Case 4 being healthy (i.e., without the additional shunt resistor).

The scaled currents, $\hat{i}_{k,\ell}(t)$, for cell k of Case ℓ were calculated from the experimental current data, $i_{k,\ell}(t)$, using

$$\hat{i}_{k,\ell}(t) = c_i(i_{k,\ell}(t_{0,\ell} : t_{f,\ell}))/i_{k,\ell}(t_{0,\ell}) \quad (18)$$

with $c_i = -1.675$ A being a scaling constant to normalise against the training data (i.e., for the currents shown in Fig. 3), $t_{0,\ell} = 500$ s being the starting time of the scaling current of Case ℓ (to reduce the effects of the harmonics at the beginning of the data), and $t_{f,\ell}$ being the final time.

Data availability

The simulation and experimental datasets are available from the corresponding authors upon request.

Code availability

The DFN-type simulation and post-processing codes that have been used to produce the results of this study are available by the corresponding authors upon request.

Received: 20 May 2023; Accepted: 29 October 2025;

Published online: 21 January 2026

References

1. Bauen, A. Future energy sources and systems—acting on climate change and energy security. *J. Power Sour.* **157**, 893–901 (2006).
2. Lu, L., Han, X., Li, J., Hua, J. & Ouyang, M. A review on the key issues for lithium-ion battery management in electric vehicles. *J. Power Sour.* **226**, 272–288 (2013).
3. Qi, C. et al. Safety analysis of lithium-ion battery by rheology-mutation theory coupling with fault tree method. *J. Loss Prev. Process Ind.* **49**, 603–611 (2017).
4. Wu, C., Zhu, C., Ge, Y. & Zhao, Y. A review on fault mechanism and diagnosis approach for Li-ion batteries. *J. Nanomater.* **2015**, 1–9 (2015).
5. Sun, P., Bisschop, R., Niu, H. & Huang, X. A review of battery fires in electric vehicles. *Fire Technol.* **56**, 1361–1410 (2020).
6. Diaz, L. B. et al. Review—Meta-review of fire safety of lithium-ion batteries: industry challenges and research contributions. *J. Electrochem. Soc.* **167**, 090559 (2020).
7. Cui, Z., Cui, N., Rao, J., Li, C. & Zhang, C. Current distribution estimation of parallel-connected batteries for inconsistency diagnosis using long short-term memory networks. *IEEE Trans. Transp. Electrif.* **8**, 1013–1025 (2022).
8. Ma, G., Xu, S. & Cheng, C. Fault detection of lithium-ion battery packs with a graph-based method. *J. Energy Storage* **43**, 103209 (2021).
9. Cheng, Y., D'Arpino, M. & Rizzoni, G. Optimal sensor placement for multifault detection and isolation in lithium-ion battery pack. *IEEE Trans. Transp. Electrif.* **8**, 4687–4707 (2021).
10. Tran, M.-K. & Fowler, M. A review of lithium-ion battery fault diagnostic algorithms: current progress and future challenges. *Algorithms* **13**<https://www.mdpi.com/1999-4893/13/3/62> (2020).
11. Devie, A., Baure, G. & Dubarry, M. Intrinsic variability in the degradation of a batch of commercial 18650 lithium-ion cells. *Energies* **11**, 1031 (2018).
12. Brand, M. J., Hofmann, M. H., Steinhardt, M., Schuster, S. F. & Jossen, A. Current distribution within parallel-connected battery cells. *J. Power Sour.* **334**, 202–212 (2016).
13. Fill, A., Koch, S. & Birke, K. P. Analytical model of the current distribution of parallel-connected battery cells and strings. *J. Energy Storage* **23**, 37–43 (2019).
14. Bruen, T. & Marco, J. Modelling and experimental evaluation of parallel connected lithium ion cells for an electric vehicle battery system. *J. Power Sour.* **310**, 91–101 (2016).
15. Ding, X., Cui, Z., Yuan, H. & Cui, N. Diagnosis of connection fault for parallel-connected lithium-ion batteries based on long short-term memory networks. *J. Energy Storage* **55**, 105552 (2022).
16. Kang, Y. et al. A comparative study of fault diagnostic methods for lithium-ion batteries based on a standardized fault feature comparison method. *J. Cleaner Prod.* **278**, 123424 (2021).
17. Feng, X., Pan, Y., He, X., Wang, L. & Ouyang, M. Detecting the internal short circuit in large-format lithium-ion battery using model-based fault-diagnosis algorithm. *J. Energy Storage* **18**, 26–39 (2018).
18. Gao, W. et al. Micro-short-circuit diagnosis for series-connected lithium-ion battery packs using mean-difference model. *IEEE Trans. Ind. Electron.* **66**, 2132–2142 (2019).
19. Couto, L. D., Reniers, J., Zhang, D., Howey, D. A. & Kinnaert, M. Degradation monitoring and characterization in lithium-ion batteries via the asymptotic local approach. *IEEE Trans. Control Syst. Technol.* **33**, 189–206 (2025).
20. Chen, W., Chen, W.-T., Saif, M., Li, M.-F. & Wu, H. Simultaneous fault isolation and estimation of lithium-ion batteries via synthesized design of Luenberger and learning observers. *IEEE Trans. Control Syst. Technol.* **22**, 290–298 (2014).
21. Jiang, J., Li, T., Chang, C., Yang, C. & Liao, L. Fault diagnosis method for lithium-ion batteries in electric vehicles based on isolated forest algorithm. *J. Energy Storage* **50**, 104177 (2022).
22. Sun, Z., Liu, P. & Wang, Z. Real-time fault diagnosis method of battery system based on Shannon entropy. *Energy Procedia* **105**, 2354–2359 (2017).

23. Xiong, J. et al. Failure detection for over-discharged Li-ion batteries. In *IEEE International Electric Vehicle Conference*, 1–5 (IEEE, 2012).

24. Xia, B., Chen, Z., Mi, C. & Robert, B. External short circuit fault diagnosis for lithium-ion batteries. In *IEEE Transportation Electrification Conference and Expo (ITEC)*, 1–7 (IEEE, 2014).

25. Zhao, Y., Liu, P., Wang, Z., Zhang, L. & Hong, J. Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods. *Appl. Energy* **207**, 354–362 (2017).

26. Tran, M.-K. & Fowler, M. Sensor fault detection and isolation for degrading lithium-ion batteries in electric vehicles using parameter estimation with recursive least squares. *Batteries* **6**, 1 (2019).

27. Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M. & Dietmayer, K. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *J. Power Sour.* **239**, 680–688 (2013).

28. Yao, L., Fang, Z., Xiao, Y., Hou, J. & Fu, Z. An intelligent fault diagnosis method for lithium battery systems based on grid search support vector machine. *Energy* **214**, 118866 (2021).

29. Sidhu, A., Izadian, A. & Anwar, S. Adaptive nonlinear model-based fault diagnosis of Li-ion batteries. *IEEE Trans. Ind. Electron.* **62**, 1002–1011 (2015).

30. Petzold, L. Differential/algebraic equations are not ODE's. *SIAM J. Sci. Stat. Comput.* **3**, 367–384 (1982).

31. Dubarry, M., Devie, A. & Liaw, B. Y. Cell-balancing currents in parallel strings of a battery system. *J. Power Sour.* **321**, 36–46 (2016).

32. Chang, L. et al. Correlations of cell-to-cell parameter variations on current and state-of-charge distributions within parallel-connected lithium-ion cells. *J. Power Sour.* **437**, 226869 (2019).

33. Hosseinzadeh, E., Marco, J. & Jennings, P. Combined electrical and electrochemical-thermal model of parallel connected large format pouch cells. *J. Energy Storage* **22**, 194–207 (2019).

34. Hosseinzadeh, E. et al. Quantifying cell-to-cell variations of a parallel battery module for different pack configurations. *Appl. Energy* **282**, 115859 (2021).

35. Lone, J. A., Tomar, N. K. & Bhaumik, S. Functional observer design for parallel connected Li-ion battery: a descriptor systems theory approach. *IEEE Control Syst. Lett.* **7**, 961–966 (2022).

36. Zhang, D., Couto, L. D., Drummond, R., Sripad, S. & Viswanathan, V. Cell-level state of charge estimation for battery packs under minimal sensing. Preprint at <https://doi.org/10.48550/arXiv.2109.08332> (2021).

37. Zhang, D. et al. State of charge estimation of parallel connected battery cells via descriptor system theory. In *American Control Conference (ACC)*, 2207–2212 (IEEE, 2020).

38. Han, W., Altaf, F., Zou, C. & Wik, T. State of power prediction for battery systems with parallel-connected units. *IEEE Trans. Transp. Electrif.* **8**, 925–935 (2021).

39. Sulzer, V., Marquis, S. G., Timms, R., Robinson, M. & Chapman, S. J. Python battery mathematical modelling (PyBaMM). *J. Open Res. Softw.* **9**, 14 (2021).

40. Tranter, T. et al. liionpack: a Python package for simulating packs of batteries with PyBaMM. *J. Open Sour. Softw.* **7**, 4051 (2022).

41. Fill, A., Koch, S., Pott, A. & Birke, K.-P. Current distribution of parallel-connected cells in dependence of cell resistance, capacity and number of parallel cells. *J. Power Sour.* **407**, 147–152 (2018).

42. Weng, A., Pannala, S., Siegel, J. B. & Stefanopoulou, A. G. Parallel-connected battery current imbalance dynamics. *IFAC-PapersOnLine* **55**, 37–43 (2022).

43. Fill, A. et al. New semi-analytical model approach of the current distribution within parallel-connected lithium-ion cells. *J. Energy Storage* **40**, 102653 (2021).

44. Drummond, R., Couto, L. D. & Zhang, D. Resolving Kirchhoff's laws for parallel Li-ion battery pack state-estimators. *IEEE Trans. Control Syst. Technol.* **30**, 2220–2227 (2021).

45. Schneider, D., Vögele, U. & Endisch, C. Model-based sensor data fusion of quasi-redundant voltage and current measurements in a lithium-ion battery module. *J. Power Sour.* **440**, 227156 (2019).

46. Chang, L. et al. A novel fast capacity estimation method based on current curves of parallel-connected cells for retired lithium-ion batteries in second-use applications. *J. Power Sour.* **459**, 227901 (2020).

47. Doyle, M., Fuller, T. F. & Newman, J. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *J. Electrochem. Soc.* **140**, 1526 (1993).

48. Ong, I. J. & Newman, J. Double-layer capacitance in a dual lithium ion insertion cell. *J. Electrochem. Soc.* **146**, 4360 (1999).

49. Drummond, R., Bizeray, A. M., Howey, D. A. & Duncan, S. R. A feedback interpretation of the Doyle–Fuller–Newman lithium-ion battery model. *IEEE Trans. Control Syst. Technol.* **28**, 1284–1295 (2019).

50. Reniers, J. M. & Howey, D. A. Digital twin of a MWh-scale grid battery system for efficiency and degradation analysis. *Appl. Energy* **336**, 120774 (2023).

51. MATLAB, version 9.11.0.1837725 (R2021b) (The MathWorks Inc., Natick, Massachusetts, 2021).

52. Frank, A. et al. Modeling inhomogeneities during parallel-connected fast charging of lithium-ion battery systems. *J. Electrochem. Soc.* **172**, 040505 (2025).

53. Kortschak, B., Kurtulus, C., Dohr, M., Wiedemann, U. & Hennige, V. Detection method of battery cell degradation. In *IEEE Vehicle Power and Propulsion Conference*, 1–6 (IEEE, 2011).

54. Song, Z., Delgado, F. P., Hou, J., Hofmann, H. & Sun, J. Individual cell fault detection for parallel-connected battery cells based on the statistical model and analysis. In *American Control Conference (ACC)*, 1155–1160 (IEEE, 2020).

55. Stroe, D.-I., Swierczynski, M., Kær, S. K. & Teodorescu, R. A comprehensive study on the degradation of lithium-ion batteries during calendar ageing: the internal resistance increase. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 1–7 (IEEE, 2016).

56. 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 (2017).

57. Dubarry, M., Vuillaume, N. & Liaw, B. Y. From single cell model to battery pack simulation for Li-ion batteries. *J. Power Sour.* **186**, 500–507 (2009).

58. Dubarry, M. & Devie, A. Initial conditioning characterization test and other preliminary testing. Tech. Rep. EVTC Report HNEI-06-15 (2015).

59. Cristianini, N. & Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (Cambridge University Press, 2000).

60. Cervantes, J., Garcia-Lamont, F., Rodriguez-Mazahua, L. & Lopez, A. A comprehensive survey on support vector machine classification: applications, challenges and trends. *Neurocomputing* **408**, 189–215 (2020).

61. Arabsalmanabadi, B., Tashakor, N., Goetz, S. & Al-Haddad, K. Li-ion battery models and a simplified online technique to identify parameters of electric equivalent circuit model for EV applications. In *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 4164–4169 (IEEE, 2020).

62. Duh, Y.-S. et al. Characterization on thermal runaway of commercial 18650 lithium-ion batteries used in electric vehicles: a review. *J. Energy Storage* **41**, 102888 (2021).

Acknowledgements

This work was supported by EDF Energy, who funded the internship of Pierre Lambert that resulted in this article. Ross Drummond was supported by a UK Intelligence Community Research Fellowship from the Royal Academy of Engineering. Stephen Duncan and Eloise Tredenick were supported in part by the Nextrode project, funded by the Faraday Institution [grant number FIRG015]. Joseph Ross was supported by Brill Power and the Royal Commission for the Exhibition of 1851. David Howey was supported by the Faraday Institution [grant number FIRG084].

Author contributions

P.L. is responsible for development of the pack model, detection methods and original article draft. R.D. is responsible for development of the pack model, DFN parallel model, detection methods and original article draft. E.C.T. is responsible for development of the DFN parallel model, design of the project approach and article review. JPR is responsible for the experiments and article review. DAH is responsible for article editing and supervision of JPR. SRD is responsible for design of the project approach and article review.

Competing interests

J.P.R. is employed by Brill Power Limited. The other authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s44172-025-00543-x>.

Correspondence and requests for materials should be addressed to Ross Drummond.

Peer review information *Communications Engineering* thanks Ala A Hussein and Jinpeng Tian for their contribution to the peer review of this

work. Primary Handling Editors: [Jiangong Zhu] and [Mengying Su and Rosamund Daw].

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 4.0 International License, which permits use, sharing, adaptation, 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 changes were made. 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/4.0/>.

© The Author(s) 2025