

Systemic Software: an IT network at the service of the environment

Original

Systemic Software: an IT network at the service of the environment / Ceppa, C.. - In: DESIGN PRINCIPLES & PRACTICES. - ISSN 1833-1874. - STAMPA. - 4:(2010), pp. 345-354.

Availability:

This version is available at: 11583/2372659 since:

Publisher:

Common Ground

Published

DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)



Real-time spatiotemporal temperature gradient estimation based on a graph convolutional neural network for battery cells

Saulius Pakštys¹*, Fabio Boscarino¹, Marco Maritano¹, Angelo Bonfitto¹

Center for Automotive Research and Sustainable Mobility, Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy

HIGHLIGHTS

- Accurate estimation of axial surface temperature gradient of lithium-ion cell.
- Experimental dynamic thermal and electrical data exploited for model training.
- Cell sensor-to-position ratio of 1:3 shown for temperature gradient estimation.
- Real-time capability of the model shown through processor-in-the-loop analysis.

ARTICLE INFO

Keywords:

Graph convolutional neural network
Partial immersion cooling
21700 cylindrical cell
Spatiotemporal modelling
Temperature estimation

ABSTRACT

Monitoring the temperature distribution of cells in large-scale battery packs can be a costly endeavour. This paper proposes a spatiotemporal model based on a graph convolutional neural network for the estimation of the axial surface temperature distribution of 21 700 cylindrical cells in a small-scale battery pack. The model is developed on the basis of experimental results obtained from the studied pack subject to various immersion thermal management conditions. The pack is charged and discharged at current rates of 0.5C and 1C at four immersion ratios: 25%, 50%, 75% and 100% with respect to the total cell height. A cross-fold training methodology is employed to train the model making use of thermally dynamic input data. The results show that a root mean square error of less than 1.1 °C can be achieved for the fold displaying the poorest performance. Finally, this paper includes a processor-in-the-loop analysis, indicating the feasibility of the model to run on embedded hardware. It is shown that up to 50 cells may be monitored in real-time. The model offers the possibility to expand the safety and health monitoring in battery management systems for not only battery packs with novel thermal management solutions, but conventional solutions as well.

1. Introduction

Lithium-ion battery usage has significantly increased in many industrial sectors globally, including the automotive and renewable energy storage industries. Continued efforts are being made to improve costs, energy density, lifetime, safety and operating temperature of these systems [1], such that sustainability and net-zero goals set by policy-makers are reached [2]. One important element of a lithium-ion battery system is its thermal management system, which plays a crucial role in regulating the temperature of the cells during electrical loads. This component enables for the safe operation of the battery pack, through the removal or addition of heat power based on the temperature distribution across it. Temperature homogeneity across the pack, as well as within each individual cell arranged in the system, is of paramount importance for an extended lifetime and elevated performance. A study

has shown [3] that due to the change in internal resistance between cell layers caused by temperature inhomogeneity, an uneven current flow will occur and contribute to a reduction in the cell lifetime. In addition, the available power and useable capacity of the battery pack may be reduced resulting from thermal gradients, where certain cells reach their imposed voltage limit earlier [4,5].

An optimal temperature range between 15 °C and 35 °C is proposed by [6,7], outside of which unwanted side reactions will occur leading to the degradation of the cell. In addition, the maximum temperature gradient within a battery pack is generally accepted to be 5 °C [8,9]. Given the higher sensitivity of electrical performance at the level of the cell, the temperature gradient should be limited even further. Studies on cylindrical cells show that temperature gradients of up to 5 °C are possible between the core and surface of the jellyroll [10–12]. A study

* Corresponding author.

E-mail address: saulius.pakstys@polito.it (S. Pakštys).

by [10] shows that a variation of the thermal management medium flow rate affects the radial and axial temperature gradients of a cell. The work indicates that a radial and an axial temperature gradient of up to 6 °C and up to 2 °C, respectively, are possible.

For large power applications, liquid thermal management systems are considered most adequate, given the high specific heat capacity and thermal conductivity observed in comparison with air-based systems [13]. Indirect liquid thermal management solutions have been widely adopted, predominantly in the automotive industry, in the form of bottom or side-cooling plates. With adequate cooling-plate design and coolant velocity, a study [14] shows that the maximum cell temperature inside the battery pack may be limited below 30 °C during harsh electrical loads. Additionally, the same study indicates that the cell temperature gradient can be limited to 5 °C. However, indirect liquid cooling poses challenges in system complexity and thermal gradients at the cell and battery pack level. This may be improved through immersion thermal management.

Direct liquid thermal management of lithium-ion battery cells is characterized by the contact of a dielectric fluid with the cell surface [13]. This solution simplifies the thermal management system by reducing complexity, and allows for the best cell and pack temperature uniformity compared to air-based and indirect liquid thermal management strategies [13]. This is particularly true for full immersion of the cells. In the case of partial immersion, in which only a portion of the cells are in direct contact with the coolant, this may not hold true. It is explored as a solution due to the potential reductions in fluid cost that can be achieved, as well as reductions in system mass that leads to higher gravimetric energy densities of large-scale battery packs. With the reduction of fluid volume comes the challenge of mitigating thermal gradients within the cell and the pack. This requires temperature feedback to the battery management system which is conventionally performed through the use of temperature sensors, such as thermistors. Cost constraints limit the use of large quantities of sensors within the pack, and this challenge is augmented when considering the monitoring of axial surface temperature gradients of cells in real-time. This is identified as the primary motivation for the use of thermal modelling techniques to estimate cell-specific thermal behaviour.

1.1. Battery cell thermal modelling

Thermal modelling of lithium-ion cells is a prevalent topic in the literature, with multiple methods having been explored. These may be divided into: impedance and resistance-based, thermal physics-based, and data-driven solutions. Impedance and resistance-based solutions use a look-up table to map the battery temperature based on parameters such as electrochemical impedance spectroscopy and direct current resistance [15–19]. These parameters are obtained offline in laboratory environments. A limitation of such methods include measurement interference and the need for a highly accurate and reliable electrochemical workstation. Thermal physics-based solutions are the most popular for estimating temperatures in battery packs, commonly using a coupled method between thermal and electrochemical physics [20–26]. It must be noted that these methods can either be applied in finite element modelling, or low-order modelling which is less computationally demanding and can be implemented in online applications. Low-order thermal modelling includes the use of lumped-parameter thermal networks (LPTN) which have shown promising results in estimating cell temperatures [27–29].

Data-driven techniques, namely machine learning, have shown great interest in the literature. This is partly due to the significant amount of electrochemical and thermal parameters that do not become necessary to be known a-priori. Indeed, such methods rely on mathematical formulations to mimic the structure and function of biological neural networks, with learned parameters that associate input with output features. When considering the complexity of large-scale battery packs, it becomes increasingly difficult to represent spatiotemporal dynamics

by a single physics-based model. The number of parameters to identify increases drastically and the modelling process becomes unfeasibly laborious. Machine learning methods can be adequate alternatives, having shown success in the estimation of state-of-health, state-of-charge and state-of-temperature [30–33]. However, a shortcoming of such methods is their black-box nature, which leads to difficulty in their interpretation. To further enhance the thermal modelling of large-scale battery packs and the complexity of the spatial domain, recent studies have turned their attention to graph-based learning.

The graph convolutional neural network (GCNN) was first proposed by [34], and is a modelling method that is based on the concepts and attributes of graphs. It offers a greater possibility for interpretation, moving further away from the black-box nature of other machine learning techniques and more towards a grey-box representation. GCNNs have been successfully implemented in various applications. In the semiconductor field, where chips become more complex and traditional thermal analysis struggles with modelling and cost computation, Miao et al. [35] propose a real-time temperature prediction of multi-core chips based on GCNN with a mean squared error within 0.5 and time delay up to 2 ms. A study by [36] investigates the estimation of induction motor temperature by comparing different modelling approaches based on GCNNs.

This method is also extended for estimation of different lithium-ion battery parameters. A study by [37] predicts the battery temperature by combining GCNN to capture the spatial relationship between battery cycles with a transformer encoder to extract temporal and sequential thermal patterns. In the same context of battery temperature estimation, [38] proposes a novel neural network that integrates GCNN with LSTM to reconstruct the temperature field of a battery pack with sparse temperature measurements. The model achieves the lowest RMS error of 0.11 °C and shows that a sensor-to-cell ratio of 1:40 is sufficient to train the model and achieve the reported level of accuracy. Other studies employing GCNN focus on estimating the capacity of lithium-ion batteries [39,40], which provide valuable insights to determine battery ageing.

The work herein exploits three distinct surface temperature measurements of the central cell of a low-voltage battery pack. The use of the surface temperature here is aligned with automotive industry practices which consider costs and practical implementation associated with thermal monitoring of battery packs. The pack features a partial immersion thermal management design and is subjected to four different immersion ratios. The cylindrical cells in the pack are immersed at 25%, 50%, 75% and 100% of the total cell height. The utility of GCNNs has been shown in the literature for the modelling of temperature fields in large-scale lithium-ion battery packs. The application of this method can be extended towards the spatiotemporal temperature modelling of the single cell in the studied lithium-ion battery pack, defining the case of study. The method shows that high estimation accuracies are possible, as well as its deployment on embedded hardware being a feasible venture. Additionally, leveraging the method's use of graph structures, a similarity with lumped-parameter thermal networks (derived from physics-based foundations) may be drawn. Thus, a grey-box representation is possible due to the modelled physical thermal exchanges that occur between regions within the system.

1.2. Contributions

This paper proposes the use of a graph convolutional neural network to reliably estimate the temperature evolution of the central cell in the pack at its different surface axial positions. To maintain the optimal performance of the individual cells in the pack, their temperature distribution must be known, such that adequate thermal management actions can be implemented in a timely manner. Temperature variation at the level of the module or pack is required to be measured and controlled, with numerous papers addressing this topic. The axial surface temperature variation estimation and control of individual cells in a

Table 1
Parameters of the Samsung INR21700-50G used in this study.

Property	Value	Unit
Minimum Capacity	4.8	[Ah]
Typical capacity	4.85	[Ah]
Nominal voltage	3.6	[V]
Discharge cut-off voltage	2.5	[V]
Charging voltage	4.2	[V]
Maximum discharge rate	2C	[1/h]
Maximum charge rate	1C	[1/h]
Mass	69.5	[g]
Chemistry	Nickel-Cobalt-Manganese	[-]

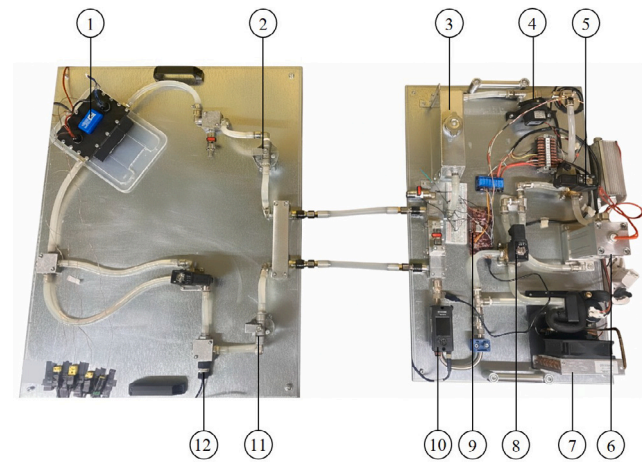
battery pack has a limited presence in the literature, and even less presence considering novel thermal management strategies, such as those using immersion technology. The present work improves on previous research using GCNNs to estimate the thermal distribution within a battery pack through the use of a low number of sensors, with focus on estimating the axial surface thermal gradients of cells when subjected to partial direct liquid cooling. The thermal management strategy enables for relatively rapid thermal transients that are non-negligible when considering the longevity of the cells. Thus, the developed GCNN is offered as a cost-effective method to monitor cell thermal gradients. To the best of the authors' knowledge, the use of a GCNN to estimate the axial surface temperature gradient of a cell within a battery pack has not been previously explored. The contributions of the work are as follows:

1. An accurate temperature estimation method of positions-of-interest of a battery cell through the implementation of a GCNN. This is experimentally validated on the central cell of the studied battery pack, with a validation RMSE of 0.07 °C achieved for the best performing condition. Literature [38] has shown a validation error of 0.19 °C, with which a majority of validation errors in this study align with.
2. The importance of using dynamic thermal and electrical data for the establishment of a robust model is shown. The work herein makes use of a data set that is comprised of different dynamic thermal conditions and electrical loads, including charging cycles of 1C. The additional use of dynamic thermal training data expands on the existing literature [38] which primarily uses dynamic electrical data.
3. A reduction of temperature sensors to estimate the axial thermal gradient of cells is achieved. The reduction allows for only one sensor to obtain three temperature evolutions along the axial length of the cell. This enables a sensor-to-position ratio of 1:3, allowing for sensor cost reductions while still able to evaluate the thermal field across the cell.
4. The work shows that up to 50 cells can be monitored online, exhibiting the capability of real-time implementation of the model onto a 32-bit microprocessor. This is performed through a processor-in-the-loop analysis of the developed GCNN, indicating a CPU utilization of 53.1%. Therefore, a comprehensive thermal field of both the cells and the pack may be realized.

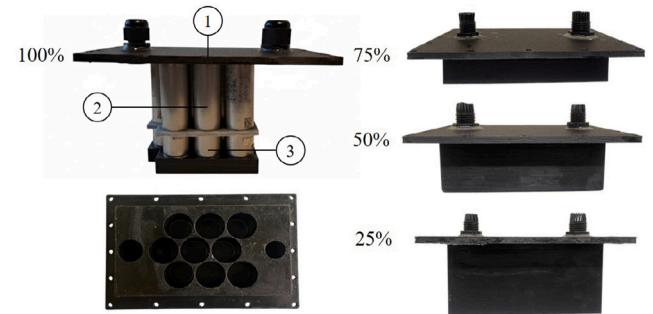
The remainder of the paper is structured as follows: The testbed used to obtain the experimental data for the training and validation of the model is described in Section 2, followed by Section 3 which presents the developed spatiotemporal temperature model. Section 4 contains the results and discussion, and finally, the conclusions and future steps are presented in Section 5.

2. Experimental setup and data

The small-scale battery pack employs Samsung INR21700-50G cylindrical cells, with their relevant dimensional and electrical parameters listed in Table 1. The nominal energy rating of the pack is 158.8



(a) Thermal management system - 1: 3s3p battery pack, 2: oil filter, 3: reservoir, 4: pump, 5: solenoid valve, 6: heater, 7: chiller, 8: solenoid valve, 9: Texas Instruments F28379D, 10: KEYENCE flowmeter, 11: oil filter, 12: conductivity sensor by JUMO.



(b) Four lids manufactured to enable partial immersion testing. The bottom left image displays the bottom of the lid to show the honeycomb layout of the battery pack. The 100% immersion assembly including the battery pack indicates approximate positions of the K-type thermocouples measuring the following surface temperatures: 1. negative terminal, 2. cell body surface, 3. positive terminal.

Fig. 1. Equipment for experimental testing and data generation.

Wh, with a 3S3P electrical configuration. It displays an intermediate spacer designed to prevent short-circuiting and a mount on which the cells are positioned in a honeycomb layout. The assembly is placed into a housing tray with one inlet and one outlet (both BSP 3/8 inch connectors) positioned at the level of the mount. This configuration is selected because the cells are flat-top one-sided, with the busbars located only on the positive terminal side (mount level). The busbars typically display higher current densities and subsequently higher temperatures [41], motivating the placement of the hydraulic ports at their level. The pack is sealed with four different lids, enabling the battery pack to be immersed at four immersion ratios: 25%, 50%, 75% and 100%. The immersion ratio is defined as:

$$r_i = h_i/h_c \quad (1)$$

in which h_i is the height of fluid relative to the cell height, and h_c is the total height of the cell.

The selected coolant is AmpCool AC-110, a dielectric mineral oil from Engineered Fluids. It circulates within a dedicated thermal management circuit, directly connected to the battery pack inlet and outlet. The fluid is stored in a 1-litre reservoir and circulated by a 24 V centrifugal pump. Two 24 V solenoid valves are implemented to direct the fluid either to a 200 W heater or a 400 W chiller, depending on the thermal state of the battery, which is monitored and controlled by a Texas Instruments microcontroller. It uses a deterministic rule-based

strategy based on temperature readings, where three K-type thermocouples are positioned on the surface along the axis of the central cell: one at the negative terminal, one at the positive terminal and one at mid-height (cell body). The activation and deactivation of the chiller is commanded by the maximum of these three measurements. The bounds set for the present study are between 18 °C and 28 °C. In addition, the system also includes two passive oil filters and a conductivity sensor to monitor fluid dielectric degradation. The described components are visible in Fig. 1(a). The assembly with the four lids is displayed in Fig. 1(b), as well as the indicative positions of the three thermocouples.

An ITECH IT6010C-80-300 bi-directional programmable DC power supply facilitates the pack's charging and discharging, delivering a maximum output of 80 V and up to 10 kW of power. Temperature data is logged using an ITECH ITS5601 multi-channel temperature logger, which is capable of monitoring up to eight channels at once. The logger's specifications include a measurement range spanning -200 °C to 2000 °C, an accuracy of 0.5 °C, and a resolution of 0.01 °C.

2.1. Data preparation

The cross-fold strategy is employed in the training and validation of the proposed model. Four electrical loads are applied to the battery pack: a constant current (CC) discharge at 0.5C, a constant-current-constant-voltage (CC/CV) charge at 0.5C, a CC discharge at 1C, and a CC/CV charge at 1C. The battery is discharged from a state-of-charge of 100% up to the cut-off voltage of 7.5 V (corresponding to 2.5 V at each cell). The pack is then charged up to the maximum voltage of 12.6 V. The charge and discharge loads are performed at the same C-rate, with a resting period of at least 12 h in between each test. The loads are repeated for each immersion ratio, producing a total of 16 temperature evolutions at each axial position of the central cell. The battery pack is discharged and charged once per C-rate per immersion ratio.

The datasets are divided into groups according to the immersion ratio. For each of them, there are four temperature evolutions corresponding to the four electrical loads. Thus, there are four groups of four temperature sets: G1, G2, G3 and G4. For Fold one (F1), the group corresponding to 100% immersion is used as the validation dataset, for which the model has not been exposed to during the training phase. The remaining groups become the training datasets. Fold two (F2) uses data from the 75% immersion for validation, while the remaining groups are used for training. Similarly, Folds three (F3) and four (F4) use 50% and 25% respectively, for model validation. Table 2 collects the groups for training and validation in each fold. Overall, the dataset has a total size of 4×4 temperature–time histories with an acquisition frequency of 10 Hz. The total dataset is approximately 27.9 h, of which Folds one, two, three and four utilize approximately 21.0, 20.8, 20.9 and 21.0 h of this total respectively.

The model is trained as described above to evaluate the method's effectiveness in estimating temperature at different locations in different dynamic thermal conditions. The separation of the dataset into training and validation sets based on the different immersion levels provides an opportunity to address distinct thermal scenarios that may occur in battery packs designed for direct liquid thermal management. Due to possible inhomogeneous temperature distributions along the longitudinal direction, the axial temperature profiles of cells at different locations will differ. By evaluating the model using a cross-fold strategy, the robustness of the modelling approach is demonstrated, providing valuable insights for future data generation and operating conditions.

3. Spatiotemporal modelling

The fundamental concept of GCNN modelling lies in the structure of a graph, which can be described by relations and their interactions with nodes. The relations are termed as edges, while the nodes are called vertices [42]. Information may be stored at the vertices, edges

Table 2

Cross-fold data groups.

Group	Electrical load	Immersion ratio [%]	F1	F2	F3	F4
G1	0.5C CC discharge	100	V	T	T	T
	0.5C CC/CV charge					
	1C CC discharge					
	1C CC/CV charge					
G2	0.5C CC discharge	75	T	V	T	T
	0.5C CC/CV charge					
	1C CC discharge					
	1C CC/CV charge					
G3	0.5C CC discharge	50	T	T	V	T
	0.5C CC/CV charge					
	1C CC discharge					
	1C CC/CV charge					
G4	0.5C CC discharge	25	T	T	T	V
	0.5C CC/CV charge					
	1C CC discharge					
	1C CC/CV charge					

F: fold, T: training, V: validation.

or master node (a global vector connected to all vertices and edges of the network). To describe the relations between vertices within the graph, the edges may be directed or undirected. In the former, information flows from a source node to a destination node, while in the latter information may flow between nodes, meaning the edge is bidirectional. Moreover, a GCNN is a transformation that is optimisable and preserves all the attributes of the graph [42].

The objective is to estimate the axial surface temperature distribution of a cell in a battery pack using one temperature sensor and an electrical current measurement. As temperature measurements in a large-scale battery pack are taken at certain locations, indeed located at specific cells, the model aims to leverage these types of measurements to recreate the temperature field of said cells. The problem to be solved lies in the mapping of an input matrix X and estimating the temperature in time t , through a function f and the topology of the cell H . The definition provided by [38] is modified to depict a single time instant of temperature estimation:

$$Y_{t+1} = f(H; X_t). \quad (2)$$

The development of the proposed GCNN begins with the definition of the grid. In this application, the vertices are the positions of the thermocouples on the surfaces of the central cell, as well as the inputs. The inputs are the temperature measured at the positive terminal of the cell and the applied current. The edges of the graph are chosen to have a directional nature, meaning a relative importance is established on how each vertex interacts with another. Here, given that thermal phenomena are being modelled, the edges that are connected between the vertices that represent the cell body (B) and the negative terminal (NT) surface temperatures are bidirectional, representing the nature of heat power that can be transferred from one location to the another and vice versa. The edge connecting the positive terminal (PT) and the nodes found on the cell is modelled as bidirectional due to this reason, although the PT node is an input. The edges that connect the vertices representing the current inputs to the vertices required to be estimated are unidirectional. The described grid with its edges and vertices is visible in Fig. 2(a). The grid is subsequently plotted onto Cartesian coordinates such that it may be encoded within the GCNN as shown in Fig. 2(b).

The connectivity of a given graph can be described by an adjacency matrix, in which the relation between one node and another can be identified through a binary notation. Depending on the directionality of the edge, the information flow from one node to another can be represented with a unitary value. If no information flows from one node to another, a null value is imposed. The spatial aspect of the model is considered appropriately through a adjacency matrix, A , that represents

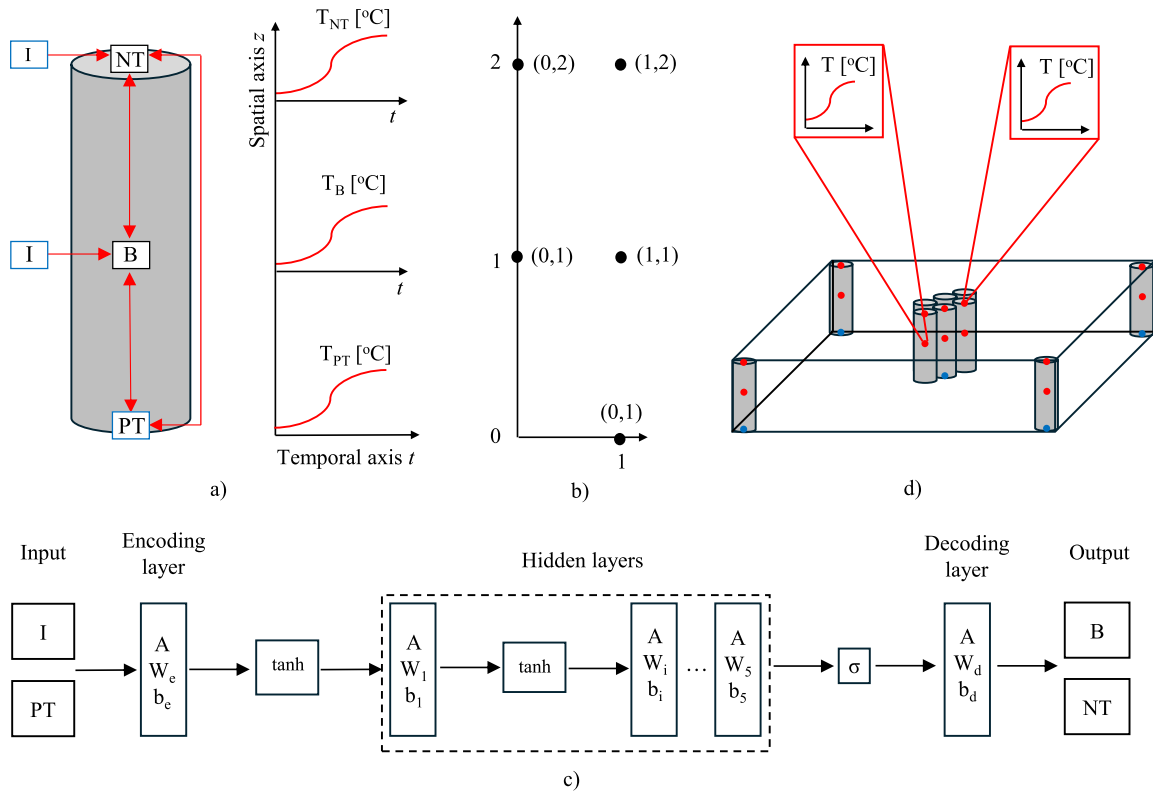


Fig. 2. (a) Identified graph for the system under study. NT: negative terminal temperature node (model output), B: cell body temperature node (model output), PT: positive terminal temperature node (model input), I: applied electrical current nodes (model inputs). The figure provides a perspective on the spatiotemporal nature of the developed model, with the temperature traces being representative of what temporal temperature measurements can be at certain locations of the cell. (b) The nodes of the identified graph plotted on a Cartesian plane for encoding. (c) The model architecture scheme. (d) The application of the spatiotemporal temperature estimation method on a large-scale battery pack in which the sensor-to-position ratio may be limited drastically. The method displays the potential for online axial temperature estimation of the cell where the sensor is placed, as well as of those nearby.

the graph in Fig. 2. It is defined as a square array of $M \times M$ (where M is the number of vertices) comprised of 0 and 1:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

The data is normalized prior to training. The applied current, I_a , is scaled by the maximum current I_{\max} , specified by the cell datasheet:

$$I_n = \frac{I_a}{I_{\max}}. \quad (4)$$

Concerning the temperature of the cell, the upper threshold of the ideal temperature range of a lithium-ion cell ($\theta_c \in [15 \leq \theta_c \leq 35]$) [6] is implemented through:

$$\theta_n = \frac{\theta_c}{\theta_{\max}} \quad (5)$$

with θ_c being the cell temperature and θ_{\max} being the upper temperature threshold. This is performed for the two axial cell surface temperatures to be estimated.

In a basic layer of a GCNN, an input graph is updated at the node and edge level by a differentiable model, to produce an output graph that is represented by the same adjacency matrix. This can be done in a cascaded arrangement of layers, such that a previous learned graph is the input to the following layer. The embeddings of each graph are updated, however the graph structure remains the same. The connectivity of the graph is the key feature that allows for the prediction task of the model. To do so, aggregation of information is required, which is performed by a pooling function [42]. Embeddings

of a graph are collected, concatenated into a matrix and aggregated, typically through a summation.

The output of the employed GCNN is computed in the following manner. The temporal features of the cell are extracted by the encoding layer, followed by five hidden layers and finally a decoding layer to generate the temperature reconstruction of the central cell in time. The input features at time $t - 1$, denoted as x_{t-1} , are transformed into the initial latent state, u_0 , through:

$$u_0 = \tanh(W_e \cdot x_{t-1} + b_e) \quad (6)$$

where

$$W_e \in \mathbb{R}^{k \times d}, b_e \in \mathbb{R}^{k \times 1}.$$

The encoding weights matrix, W_e , has a size of $k \times d$ and the encoding biases vector, b_e , has a size of $k \times 1$. The latent dimension (responsible for encoding the graph information into the latent space) is denoted by k , while d is the input dimension.

For each hidden layer, the weights matrix, W_i , and biases vector, b_i , map the preceding layer output, u_{i-1} , to an intermediate feature representation, f_i :

$$f_i = W_i \cdot u_{i-1} + b_i. \quad (7)$$

where

$$W_i \in \mathbb{R}^{k \times k}, b_i \in \mathbb{R}^{k \times 1}.$$

The weights matrix has a size of $k \times k$ and the biases vector has a size of $k \times 1$, with k being the latent dimension. Subsequently, the intermediate outputs are aggregated using the adjacency matrix, A , to account for the spatial connectivity:

$$g_i = f_i \cdot (\tilde{D}^{-\frac{1}{2}} A \tilde{D}^{-\frac{1}{2}}). \quad (8)$$

Table 3
Hyperparameters of the proposed GCNN model.

Parameter	Value
Hidden layers	5
Latent dimension	20
Learning rate	0.003

The adjacency matrix is symmetrically normalized through the degree matrix, D . It is a diagonal matrix where each diagonal element, D_{pp} , represents the sum of the weights of the edges connected to node p :

$$D_{pp} = \sum_q A_{pq} \quad (9)$$

The resulting output of a given hidden layer is treated through a hyperbolic tangent function:

$$v_i = \tanh(g_i). \quad (10)$$

Following, a residual connection is employed where the computed update, v_i , is added to the current layer input, u_{i-1} :

$$u_i = u_{i-1} + v_i. \quad (11)$$

Executing the described process involving a set number of layers, the output of the final layer, u_F , is treated through the decoding matrix, W_d , and the decoding biases vector, b_d , through:

$$h_t = W_d \cdot u_F + b_d \quad (12)$$

where

$$W_d \in \mathbb{R}^{1 \times k}, b_d \in \mathbb{R}^{1 \times 1}.$$

Finally, the output is scaled through a sigmoid function:

$$h_t = \sigma(u_F). \quad (13)$$

The model architecture is schematized in Fig. 2(c). The training of the model is performed on a machine with four cores, an Intel Core i7-8665U CPU and 32 GB of RAM, using the Deep Learning Toolbox available in MATLAB and Simulink R2024a. The algorithm developed by [43] is taken as a starting point, from which the current model is developed for the purpose of timeseries predictions. The learnable parameters (weights and biases) are tuned using the Adaptive Moment Estimation (Adam) algorithm through back-propagation. The loss function used for the reduction of the error between the estimated temperature values and the measured values is the mean squared error. The formulation is reported in Eq. (17). The maximum number of epochs is set to 1000, within which convergence in each fold is reached. The learning rate is selected through a trial-and-error process, with a value of 0.003 showing an acceptable trade-off between convergence and time during training, while the latent dimension is set to 20. In addition, it must be noted that the training of the model is performed through teacher-forcing, where the previous time step temperature estimates are assumed to be correctly estimated by the model.

Five hidden layers are initialized in the model, selected by a trial-and-error process. A single layer was selected initially, however, the fitting over the training datasets was extremely poor. Increasing the number of the layers significantly improved the estimation results, with five being an acceptable trade-off in training time and accuracy. Indeed, it was observed that above five layers, the accuracy gained is marginal compared to the increase in training time. The total number of parameters of the model are 2201, considering the five hidden, encoding, and decoding layers. Table 3 collects the hyperparameters of the GCNN model.

4. Results and discussion

The performance of the model is quantified by a set of metrics. The root mean squared error (RMSE), the maximum absolute error ($\|e\|_\infty$), the mean absolute error (MAE), the mean squared (MSE) error, the mean bias error (MBE), and the coefficient of determination (R^2) are used:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2} \quad (14)$$

$$\|e\|_\infty = \max_n |y_n - \hat{y}_n| \quad (15)$$

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |y_n - \hat{y}_n|. \quad (16)$$

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2 \quad (17)$$

$$\text{MBE} = \frac{1}{N} \sum_{n=1}^N (\hat{y}_n - y_n) \quad (18)$$

$$R^2 = 1 - \frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{\sum_{n=1}^N (y_n - \bar{y})^2} \quad (19)$$

in which for all metrics y_n is the measured value, \hat{y}_n is the estimated value, \bar{y} is the average of the measurements and n is the index of the observation in the signal. The number of observations or data points is denoted by N .

The performance evaluation of the model is first conducted on the training datasets. As for the training, teacher-forcing is employed to evaluate the model on the training and validation datasets. Tests with random noise injection on the input have shown that the model behaves neutrally, propagating the noise to the output. To remain within an acceptable length of the work, this is not reported. Table 4 collects the metrics for the different folds. It can be seen that the maximum absolute temperature error is within 6 °C for all the folds, while the negative terminal temperature fitting appears to be marginally poorer compared to that of the negative terminal. The latter can be stated by observing the remaining metrics.

Given that for each fold there are four validation scenarios, the graphical presentation of all validations is deemed excessive. Rather, this is shown only for Folds one and four, which display, respectively, one of the best and the worst validation errors. Figs. 3 and 4 display the temperature traces of the model and of the measurements for Fold one, consisting of the constant charging and discharging data at a 100% immersion ratio. Fig. 3 displays the results for the validation of a 0.5C charge and discharge, in which the error is between ± 0.5 °C. Fig. 4 displays the estimation performance for a 1C charge and discharge, showing an error bounded between ± 1 °C. The maximum absolute error for Fold one is identified to be for the 1C discharge and charge cases, with a value of 0.92 °C for both (see Figs. 7(c) and 7(d)). The maximum RMSE, MAE and MSE are seen for the 1C discharge case, being 0.51 °C, 0.46 °C and 0.26 °C² respectively for the negative terminal temperature estimation.

Additionally, the MBE provides an indication of the model performance in overestimating or underestimating the temperatures of interest. The model displays positive values for the negative terminal temperature estimation and negative values for the cell body temperature estimation (except for the 1C discharge), meaning it is overestimating the former and underestimating the latter. In this case, overestimating the temperature at the negative terminal may mean earlier thermal management control actions will be imposed, as the battery management system will be exposed to a value that is larger than the reality. This can be mitigated, however, with a moderate increase in maximum thermal limits that dictate the activation of thermal management apparatus. The largest value of MBE (by magnitude) is

Table 4
Collection of metrics for the performance of the proposed temperature estimation method over the training datasets. F: fold.

Location	RMSE [°C]				$\ e\ _{\infty}$ [°C]				MAE [°C]			
	F1	F2	F3	F4	F1	F2	F3	F4	F1	F2	F3	F4
Cell body	0.14	0.20	0.16	0.25	3.95	3.91	4.37	3.95	0.10	0.16	0.13	0.20
Negative terminal	0.33	0.34	0.25	0.28	5.27	5.27	5.69	4.57	0.23	0.25	0.18	0.19
Location	MSE [°C ²]				MBE [°C]				R ² [-]			
	F1	F2	F3	F4	F1	F2	F3	F4	F1	F2	F3	F4
Cell body	0.02	0.04	0.03	0.06	-0.06	-0.001	0.001	-0.009	0.98	0.98	0.98	0.95
Negative terminal	0.11	0.11	0.06	0.08	-0.05	-0.007	0.003	-0.001	0.93	0.93	0.96	0.94

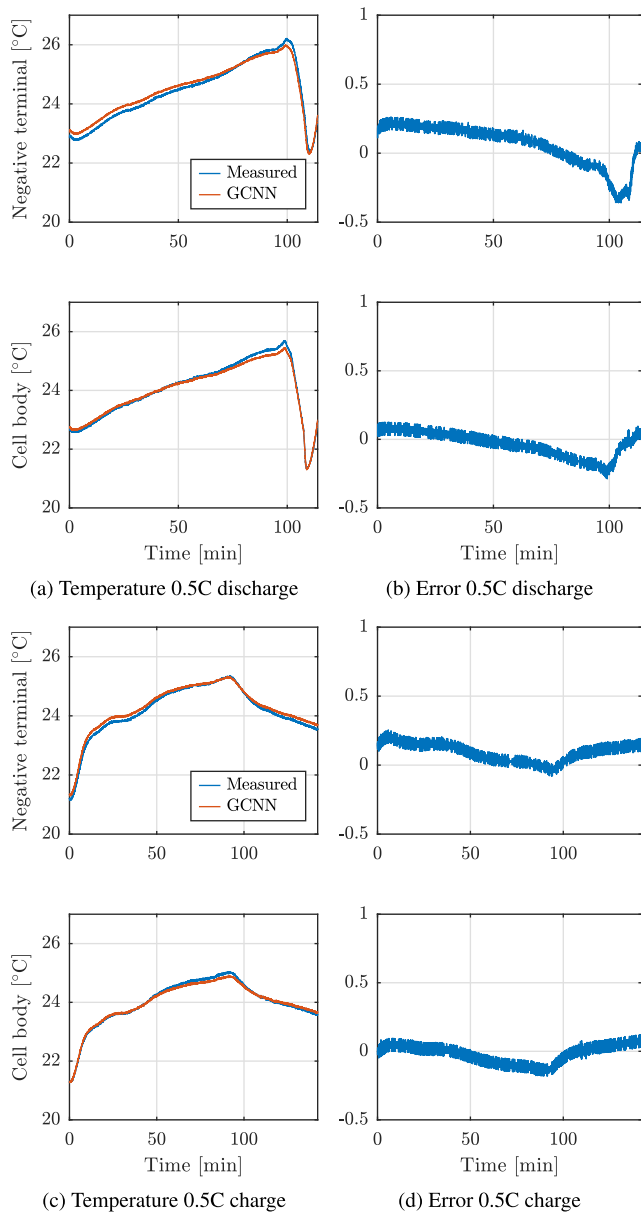


Fig. 3. Temperature estimation performance validation of the proposed model subject to 0.5C charge and discharge cycles of Fold one. Validation dataset: 100% immersion. Minimum RMSE: 0.07 °C.

seen for the 1C discharge case, with a value of 0.45 °C. Regarding the coefficient of determination, the lowest value of 0.79 is seen for the 1C discharge case, however, a value of 0.99 is seen for the 0.5C charge and discharge cases.

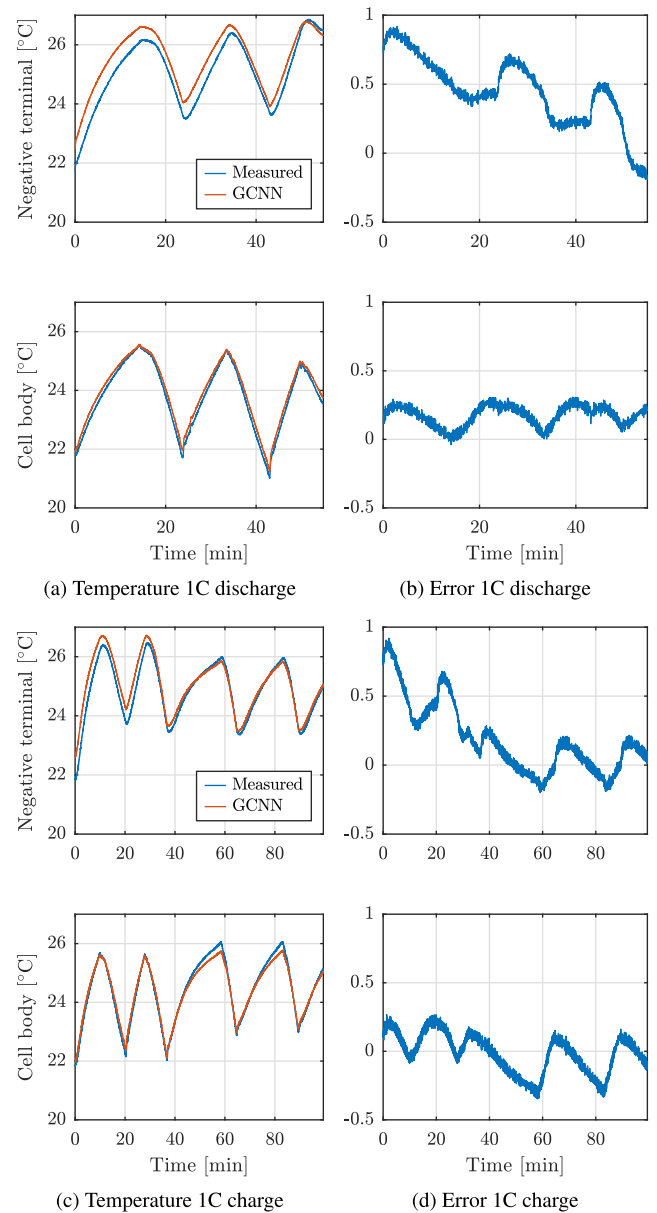


Fig. 4. Temperature estimation performance validation of the proposed model subject to 1C charge and discharge cycles of Fold one. Validation dataset: 100% immersion. Minimum RMSE: 0.14 °C.

The performance metrics indicate an adequate training of the model is performed, in which groups G2, G3, and G4 contain temperature measurements at different thermal immersion conditions which prove to be sufficient to estimate the temperature of the cell body and negative terminal for a full immersion scenario. This is coherent, as

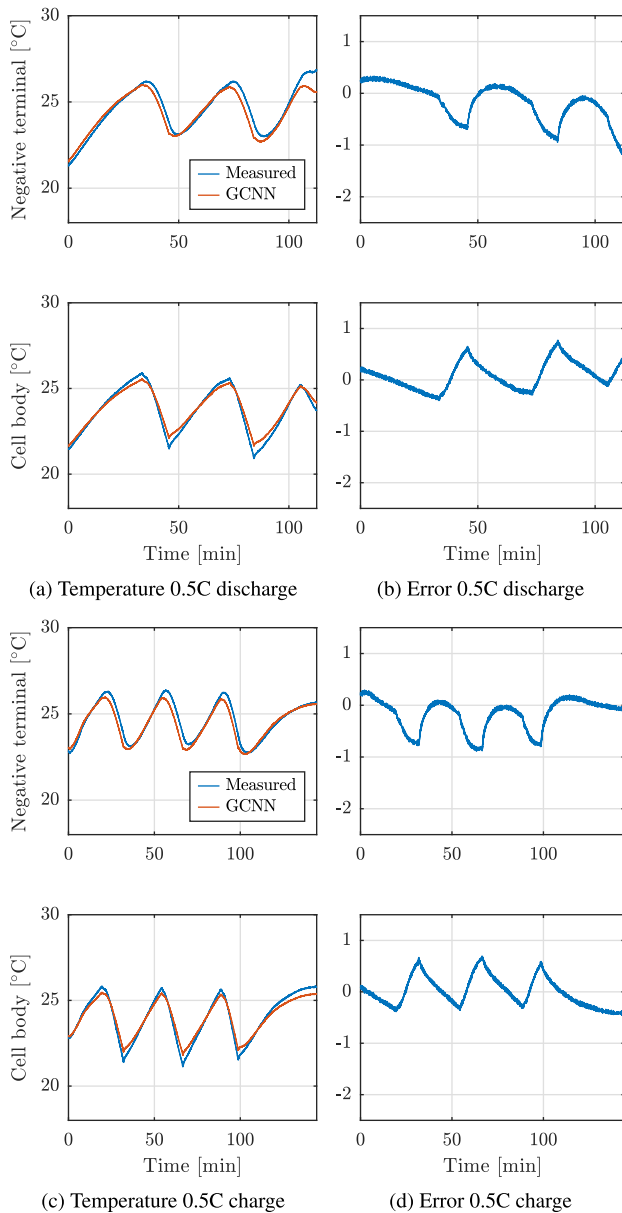


Fig. 5. Temperature estimation performance validation of the proposed model subject to 0.5C charge and discharge cycles of Fold four. Validation dataset: 25% immersion. Minimum RMSE: 0.28 °C.

the more critical cases of thermal immersion that display a greater temperature variation are used to train the GCNN. Subsequently, the validation temperature behaviour is within the temperature envelope the model is exposed to, and therefore a good estimation capability is achieved.

Concerning Fold four, the validation temperature metrics are significantly poorer. This fold includes a validation dataset for charging and discharging at 25% immersion, a more critical thermal condition. Figs. 5 and 6 display the estimation performance of the trained model and the errors for each electrical load. The former shows the model performance for 0.5C charge and discharge, in which the error is bounded between ± 1 °C. The latter presents the temperature estimation for a charge and discharge of 1C, with the error bounded between ± 2 °C. The maximum absolute error in Fold four is for the negative terminal temperature estimation in a 1C discharge condition, at 2.03 °C. The remaining metrics: the RMSE, MAE, and MSE, are the greatest for this validation set, with values of 1.05 °C, 0.97 °C

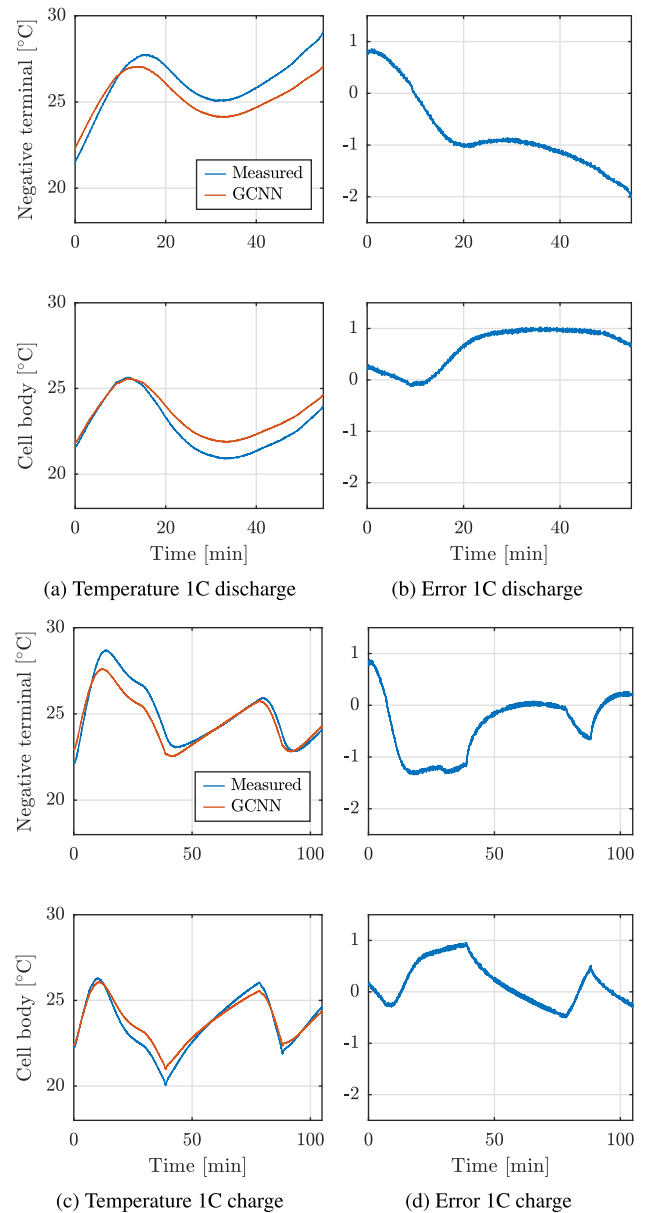


Fig. 6. Temperature estimation performance validation of the proposed model subject to 1C charge and discharge cycles of Fold four. Validation dataset: 25% immersion. Minimum RMSE: 0.44 °C.

and 1.11 °C² respectively. This may be viewed to be relatively high compared to the validation results obtained for Fold one for instance, as well as compared to the results obtained by [38]. It presents an interim conclusion, in which the use of carefully selected variable temperature data is crucial for the training of the model in this study. This expands on the results of [38], which did not consider the variation of cell temperatures through a thermal management circuit, using dynamic electrical loading only.

As for Fold one, the MBE provides indications on overestimation and underestimation for Fold four. The model is shown to underestimate the surface temperature estimations for the negative terminal and overestimate the temperature estimations for the cell body. The largest value of the MBE (by magnitude) is observed for the 1C discharge case, with a value of -0.77 °C. The lowest value of the coefficient of determination is seen to be 0.49, for the 1C discharge case.

The values of the metrics for the folds discussed thus far are presented in a collection of bar charts in Fig. 7. The same figure displays

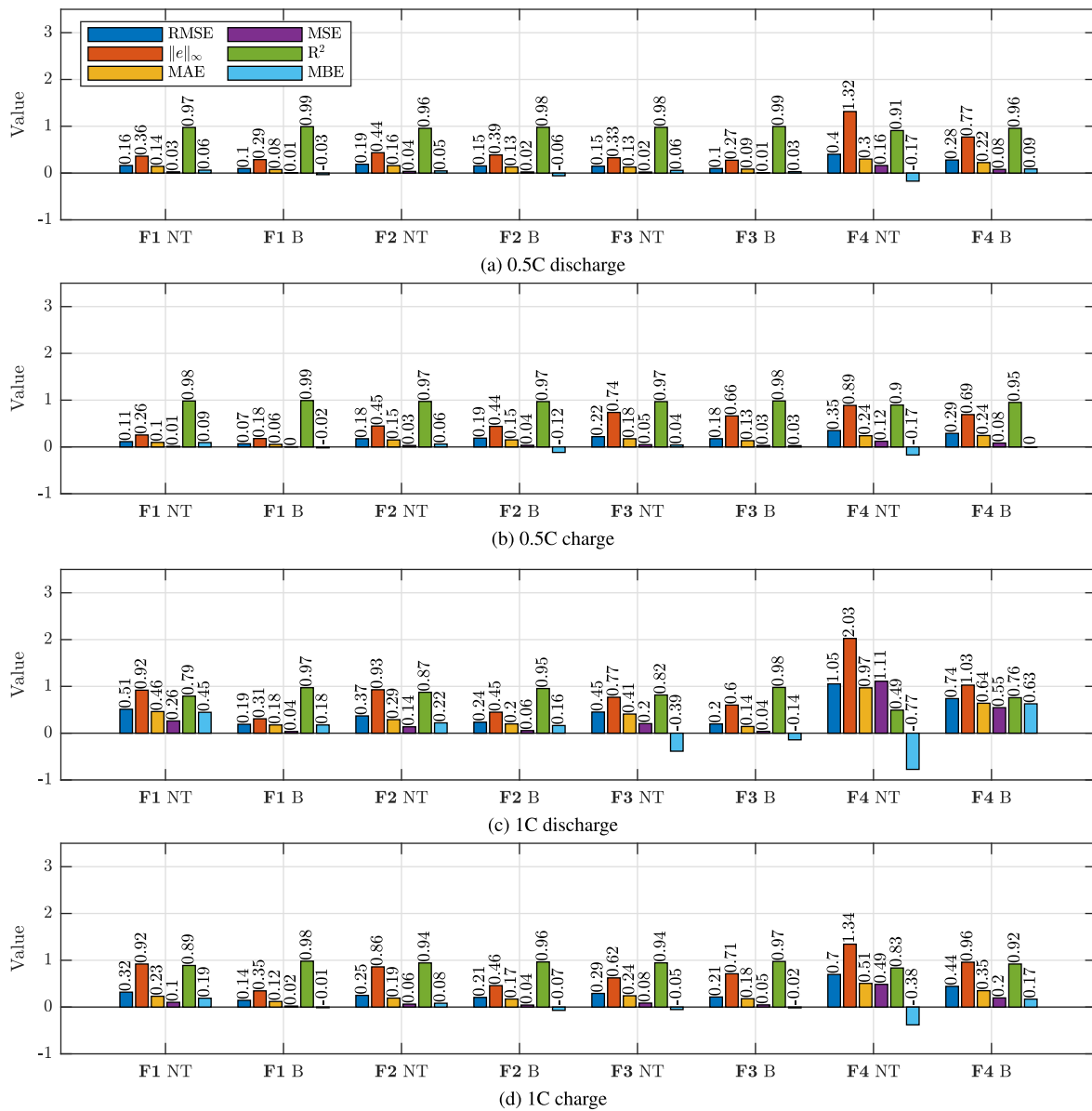


Fig. 7. Bar charts collecting the validation metrics for each fold.

the validation of the remaining folds. It may be observed that the general trend in the error increases from Fold one to Fold four, indicating that there is a necessity to collect and use data obtained during harsh thermal operating conditions. Nevertheless, it can be noted that the model is effective in estimating the temperature of the cell body and negative terminal, whereby the lowest RMSE of 0.07 °C (for the cell body) is seen for Fold one consisting of the charge and discharge cycles for immersion levels of 75%, 50%, and 25% for training. To better understand the variation of model performance between folds, the data used for training may be analysed. Fig. 8 displays three box plots showing the maximum, minimum, median and interquartile range (IQR) for the data used as training and validation in each fold. Fig. 8(a) shows the positive terminal temperature data distribution used for training in each fold. A similar distribution across the folds is visible, considering the maximum and minimum values. The IQR however is the greatest for Fold one, with a value of 2.7 °C, as compared to the lowest value of 1.9 °C exhibited by Fold four.

Fig. 8(b) displays the box plots of each fold for the cell body temperature used as the target during the training process, indicating a similar trend as for the positive terminal temperature. The IQR

decreases from Fold one at 2.1 °C to Fold four at 1.7 °C. The same decreasing trend may be observed for the range between maximum and minimum values. Additionally, within the figure, the red points display the maximum and minimum values of the validation data set for each fold. These values remain within the range of the target data for all the folds apart from Fold four, for which the values fall outside the ranges for the cell body and above the range for the negative terminal. Fig. 8(c) shows the IQR, maximum and minimum values for the negative terminal, with a similar trend visible. The IQR value decreases from 1.9 °C to 1.7 °C. As for the cell body temperature, the maximum and minimum values of the validation data for each fold are plotted. Again, these values are within the range of the target data for all folds apart from Fold four.

The lower IQR values for Fold four partially motivate the poorer performance of the model during validation, in which a lower exposure of diverse dynamic thermal conditions may be deduced. With 50% of the temperature values centred closer to the median, the spread of variables is up to 30% lower when considering the IQR decrease for the positive terminal temperature. Additionally, it is shown that some output temperature values for Fold four in the validation set

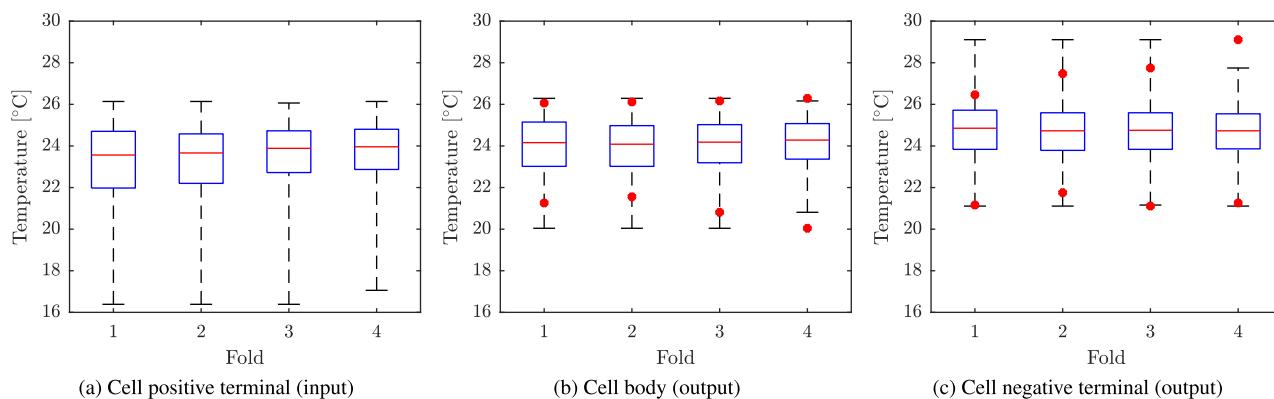


Fig. 8. Box plots displaying the interquartile range, maximum, minimum and median values for the dynamic temperature data sets of each fold used for training. The red points indicate the maximums and minimums of the validation sets for each fold.

Table 5
Benchmark comparison of low-order thermal and data-driven models.

Source	Application	Model	RMSE	MAE	MSE	R ²
Zheng et al. [16]	Li-ion battery	Impedance-based	0.20	0.17	–	–
Rodríguez-Iturriaga et al. [19]	Li-ion battery	ECM	0.82	0.65	–	–
Domalanta et al. [26]	Li-polymer battery	P2D electro-thermal	1.52	–	–	–
Allafi et al. [29]	Li-ion battery	Lumped thermal	–	0.19	–	–
Miao et al. [35]	Multi-cores chips	MLP	–	1.90	1.07	–
		CNN	–	2.06	1.01	–
		GCNN	–	0.38	0.50	–
Chen et al. [36]	Induction Motor	LSTM	2.31	1.97	–	0.68
		GCNN	1.94	1.15	–	0.78
		LSTM-GCNN	1.39	1.02	–	0.89
Li et al. [37]	Li-ion battery	GCNN	0.28	0.20	–	0.92
		TE	0.30	0.22	–	0.83
		GCNN-TE	0.25	0.05	–	0.94
Zhao et al. [38]	Li-ion battery	sL-GCNN	0.19	0.15	–	0.97
Our model (Fold one)	Li-ion battery	GCNN	0.07	0.06	0.01	0.99

fall outside the range of the training set, providing further reason for the poorer performance. Nevertheless, the use of data with a high variability would lead to better model performances, as evident for Fold three.

Moreover, to provide a benchmark of the proposed model, Table 5 reports a comparison between performance metrics obtained in this work and representative surrogate modelling approaches for temperature prediction available in the literature. A direct quantitative comparison is challenging due to the lack of studies addressing exactly the same system configuration and experimental conditions, therefore the comparison is not intended to establish which of the considered model performs best. However, several works in the literature investigate temperature prediction using different approaches in different contexts, such as lithium-ion battery or other electro-thermal components. Within this framework, the proposed approach achieves prediction errors that are similar to those reported for GCNN in the comparison table, considering the minimum validation error reported in each reviewed literature work.

Furthermore, a comparison with the existing literature highlights that GCNN-based models generally outperform other data-driven approaches, confirming the superior capability in prediction. In addition, combining GCNN with other modelling techniques further reduces the prediction error.

It is also worth noticing that traditional approaches, such as the ones adopted in [16,29], can achieve temperature estimation accuracies comparable to data-driven techniques. However, they require detailed a priori knowledge of physical and electrochemical parameters and tuning process, which limits their applicability in large-scale battery packs.

Finally, a PIL analysis is conducted to evaluate the real-time deployability of the model. The model is implemented into Simulink after being converted to a discrete time (1 Hz) 32-bit representation. The model block is converted to a PIL subsystem which is deployed onto a Texas Instruments F28069M Launchpad evaluation board, with the inputs to the model being simulated on the host computer. The inputs in this analysis are those from a 1C charge with 100% immersion. The results indicate a low CPU utilization of 1.1%, and maximum execution time of 10.7 ms. It can be concluded that the developed GCNN can be run in real time, with minimal impact on microprocessor resources.

Additionally, a further analysis is made to determine the hardware resource utilization when considering the axial surface temperature estimation of multiple cells in a battery pack. Fig. 2(d) provides a visual representation of this evaluation. To do so, the model is replicated from one to five times, representing the temperature estimation of five cells. This showed that each cell model increments the CPU utilization by approximately 1.1%, and the execution time by approximately 11 ms. Considering this, it can be shown that the CPU utilization of 10 and 50 cells is 10.6% and 53.1%, respectively. Table 6 collects the PIL results for the models with the discussed number of cells.

The developed model displays scalability for real-time estimation of the axial surface temperature distribution in multiple cells. Additionally, further development of the model can be executed such that a single temperature sensor can not only be used to estimate the axial surface temperature gradient of the cell where this sensor is mounted, but of those surrounding it as well (see Fig. 2(d)). However, considering the application of the method to a large-scale battery pack, a dedicated prototype is required to first obtain the training data for the

Table 6
Impact of GCNN battery cell number on a Texas Instruments F28069M Launchpad CPU utilization.

Cell number	CPU utilization [%]
1	1.1
2	2.2
3	3.2
4	4.3
5	5.4
10	10.6
50	53.1

cells in chosen locations. Alternatively, computational fluid dynamics models may be exploited to generate synthetic temperature data of large-scale battery pack systems that can subsequently be used to train the model. The hyperparameters used within this study are a possible area of improvement as they have been selected on a trial-and-error process. Further work would include the optimization of these variables through methods such as Bayesian optimization. Furthermore, it must be noted that if the physical design of the partial immersion thermal management were to change significantly, then a necessity for new training data would arise. This may also be stated for conventional thermal management systems using aluminium cooling-plate technology, whereby the proposed model may be employed as well. Further testing of the developed model is needed to provide insights into its adaptability to differing thermal management designs.

The developed model is aimed towards real-time applications, however it displays the possibility for battery design processes. The immersion ratio is not included as an explicit input, as it is treated as a fixed design parameter in the direct liquid-cooled battery pack, consistent with practical implementations. The generalizability of the model with respect to the immersion ratio is shown through the use of the folds as training sets, however, the lack of inclusion of the immersion ratio as an input to the model in this study does not pose limitations as each fold contains dynamic temperature evolutions. The considered immersion ratios are intrinsic to the datasets used for training. Including the immersion ratio as an input to the model would be beneficial for thermal management design, using it as a rapid tool to evaluate various concepts.

5. Conclusions

This paper proposes the use of a graph convolutional neural network, mimicking the structure of a lumped-parameter thermal network which models thermal exchange phenomena, to reliably estimate two axial positions of the cell. The work has used a thermally diverse dataset obtained experimentally to train the model using a cross-fold strategy. It is shown that temperature measurements in charging and discharging conditions ranging from 0.5C to 1C, for immersion levels of 25%, 50% and 75% are most valuable in generating a comprehensive GCNN. This is shown with a performance validation for the four identified folds, with Fold four displaying the poorest performance due to a lower spread of dynamic temperature operating points in the training dataset. An analysis of the input and output temperature data used to train the model across the four folds provides a possible reason for the different performances achieved. Nonetheless, the model exhibits an excellent capability to estimate temperature evolutions of the cell body and negative terminal, with the lowest RMSE of 0.07 °C achieved in validation. The analysis has focused primarily on the use of constant discharge and constant-current/constant-voltage charge load profiles, meaning further development would be required when considering automotive driving cycles. However, with the use of dynamic temperature data, the model's effectiveness in reproducing the targeted temperatures of a central cell in a 3s3p battery pack has been shown. Additionally, a processor-in-the-loop analysis shows that the proposed

model can be run in real-time on embedded hardware, enabling online temperature predictions of up to 50 cells in lithium-ion battery packs with a CPU utilization of 53.1%. Further developments include the use of dynamic electrical load data combined with dynamic thermal management conditions to train the model for more complex loading scenarios. Additionally, the use of the model for real-time immersion thermal management control is provisioned.

CRedit authorship contribution statement

Saulius Pakštys: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Conceptualization. **Fabio Boscarino:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis. **Marco Maritano:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis. **Angelo Bonfitto:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization.

Declaration of competing 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.

Acknowledgements

The work has been conducted within a research project funded by Stellantis/CRF. The PhD scholarships of Saulius Pakštys, Fabio Boscarino and Marco Maritano are co-funded by the Italian Ministry of Universities and Research (MUR) under DM 352/2022, DM 630/2024 and DM 117/2023, respectively.

Data availability

The data that has been used is confidential.

References

- [1] S. Mahmud, M. Rahman, M. Kamruzzaman, M.O. Ali, M.S.A. Emon, H. Khatun, M.R. Ali, Recent advances in lithium-ion battery materials for improved electrochemical performance: A review, *Results Eng.* 15 (2022) <http://dx.doi.org/10.1016/j.rineng.2022.100472>.
- [2] The European green deal - European commission, 2021, URL https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_en.
- [3] I.A. Hunt, Y. Zhao, Y. Patel, J. Offer, Surface cooling causes accelerated degradation compared to tab cooling for lithium-ion pouch cells, *J. Electrochem. Soc.* 163 (9) (2016) A1846–A1852, <http://dx.doi.org/10.1149/2.0361609jes>.
- [4] X. Liu, W. Ai, M. Naylor Marlow, Y. Patel, B. Wu, The effect of cell-to-cell variations and thermal gradients on the performance and degradation of lithium-ion battery packs, *Appl. Energy* 248 (2019) 489–499, <http://dx.doi.org/10.1016/j.apenergy.2019.04.108>.
- [5] T.M. Bandhauer, S. Garimella, T.F. Fuller, A critical review of thermal issues in lithium-ion batteries, *J. Electrochem. Soc.* 158 (3) (2011) R1, <http://dx.doi.org/10.1149/1.3515880>.
- [6] D. Chen, J. Jiang, G.-H. Kim, C. Yang, A. Pesaran, Comparison of different cooling methods for lithium ion battery cells, *Appl. Therm. Eng.* 94 (2016) 846–854, <http://dx.doi.org/10.1016/j.applthermaleng.2015.10.015>.
- [7] S. Ma, M. Jiang, P. Tao, C. Song, J. Wu, J. Wang, T. Deng, W. Shang, Temperature effect and thermal impact in lithium-ion batteries: A review, *Prog. Nat. Sci.: Mater. Int.* 28 (2018) 653–666, <http://dx.doi.org/10.1016/j.pnsc.2018.11.002>.
- [8] C. Vidal, O. Gross, R. Gu, P. Kollmeyer, A. Emadi, xEV Li-ion battery low-temperature effects—Review, *IEEE Trans. Veh. Technol.* 68 (5) (2019) 4560–4572, <http://dx.doi.org/10.1109/TVT.2019.2906487>.
- [9] L. Zhou, Y. Zheng, M. Ouyang, L. Lu, A study on parameter variation effects on battery packs for electric vehicles, *J. Power Sources* 364 (2017) 242–252, <http://dx.doi.org/10.1016/j.jpowsour.2017.08.033>.
- [10] C. Bolsinger, K.P. Birke, Effect of different cooling configurations on thermal gradients inside cylindrical battery cells, *J. Energy Storage* 21 (2019) 222–230, <http://dx.doi.org/10.1016/j.est.2018.11.030>.

- [11] D. Anthony, D. Wong, D. Wetz, A. Jain, Improved thermal performance of a li-ion cell through heat pipe insertion, *J. Electrochem. Soc.* 164 (6) (2017) A961–A967, <http://dx.doi.org/10.1149/2.0191706jes>.
- [12] D. Anthony, D. Wong, D. Wetz, A. Jain, Non-invasive measurement of internal temperature of a cylindrical li-ion cell during high-rate discharge, *Int. J. Heat Mass Transfer* 111 (2017) 223–231, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2017.03.095>.
- [13] C. Roe, X. Feng, G. White, R. Li, H. Wang, X. Rui, C. Li, F. Zhang, V. Null, M. Parkes, Y. Patel, Y. Wang, H. Wang, M. Ouyang, G. Offer, B. Wu, Immersion cooling for lithium-ion batteries – A review, *J. Power Sources* 525 (2022) 231094, <http://dx.doi.org/10.1016/j.jpowsour.2022.231094>.
- [14] Y. Li, Z. Zhou, W.-T. Wu, Three-dimensional thermal modeling of li-ion battery cell and 50 V li-ion battery pack cooled by mini-channel cold plate, *Appl. Therm. Eng.* 147 (2019) 829–840, <http://dx.doi.org/10.1016/j.applthermaleng.2018.11.009>.
- [15] Y. Zheng, N.A. Weinreich, A. Kulkarni, Y. Che, H. Sorouri, X. Sui, R. Teodorescu, Sensorless state of temperature estimation for smart battery based on electrochemical impedance, in: 2023 25th European Conference on Power Electronics and Applications, EPE'23 ECCE Europe, IEEE, Aalborg, Denmark, 2023, pp. 1–8, <http://dx.doi.org/10.23919/EPE23ECCEurope58414.2023.10264452>.
- [16] Y. Zheng, Y. Che, J. Guo, N.A. Weinreich, A. Kulkarni, A. Nadeem, X. Sui, R. Teodorescu, Real-time sensorless temperature estimation of lithium-ion batteries based on online operando impedance acquisition, *IEEE Trans. Power Electron.* 39 (2024) 13853–13868, <http://dx.doi.org/10.1109/TPEL.2024.3424267>.
- [17] X. Wang, X. Wei, J. Zhu, H. Dai, Y. Zheng, X. Xu, Q. Chen, A review of modeling, acquisition, and application of lithium-ion battery impedance for onboard battery management, *ETransportation* 7 (2021) 100093, <http://dx.doi.org/10.1016/j.etrans.2020.100093>.
- [18] S.S. Madani, E. Schaltz, S.K. Kær, An electrical equivalent circuit model of a lithium titanate oxide battery, *Batteries* 5 (2019) <http://dx.doi.org/10.3390/batteries5010031>.
- [19] P. Rodríguez-Iturriaga, D. Anseán, J.A. López-Villanueva, M. González, S. Rodríguez-Bolívar, A method for the lifetime sensorless estimation of surface and core temperature in lithium-ion batteries via online updating of electrical parameters, *J. Energy Storage* 58 (2023) <http://dx.doi.org/10.1016/j.est.2022.106260>.
- [20] K. Jithin, P. Rajesh, Numerical analysis of single-phase liquid immersion cooling for lithium-ion battery thermal management using different dielectric fluids, *Int. J. Heat Mass Transfer* 188 (2022) 122608, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.122608>.
- [21] J. Newman, W. Tiedemann, Potential and current distribution in electrochemical cells: Interpretation of the half-cell voltage measurements as a function of reference-electrode location, *J. Electrochem. Soc.* 140 (7) (1993) 1961–1968, <http://dx.doi.org/10.1149/1.2220746>.
- [22] U.S. Kim, C.B. Shin, C.-S. Kim, Effect of electrode configuration on the thermal behavior of a lithium-polymer battery, *J. Power Sources* 180 (2) (2008) 909–916, <http://dx.doi.org/10.1016/j.jpowsour.2007.09.054>.
- [23] M. Suresh Patil, J.-H. Seo, M.-Y. Lee, A novel dielectric fluid immersion cooling technology for Li-ion battery thermal management, *Energy Convers. Manage.* 229 (2021) 113715, <http://dx.doi.org/10.1016/j.enconman.2020.113715>.
- [24] V.-T. Ho, K. Chang, S.W. Lee, S.H. Kim, Transient thermal analysis of a li-ion battery module for electric cars based on various cooling fan arrangements, *Energies* 13 (9) (2020) 2387, <http://dx.doi.org/10.3390/en13092387>.
- [25] D. Chalise, K. Shah, R. Prasher, A. Jain, Conjugate heat transfer analysis of thermal management of a li-ion battery pack, *J. Electrochem. Energy Convers. Storage* 15 (1) (2018) 011008, <http://dx.doi.org/10.1115/1.4038258>.
- [26] M.R.B. Domalanta, M.T. Castro, J.D. Ocon, J.A.D. del Rosario, An electrochemical-thermal multiphysics model for lithium polymer battery, *Chem. Eng. Trans.* 94 (2022) 145–150, <http://dx.doi.org/10.3303/CET2294024>.
- [27] X. Lin, H.E. Perez, J.B. Siegel, A.G. Stefanopoulou, Robust estimation of battery system temperature distribution under sparse sensing and uncertainty, *IEEE Trans. Control Syst. Technol.* 28 (3) (2020) 753–765, <http://dx.doi.org/10.1109/TCST.2019.2892019>.
- [28] A. Samanta, S.S. Williamson, A comprehensive review of lithium-ion cell temperature estimation techniques applicable to health-conscious fast charging and smart battery management systems, *Energies* 14 (18) (2021) 5960, <http://dx.doi.org/10.3390/en14185960>.
- [29] W. Allafi, C. Zhang, K. Uddin, D. Worwood, T.Q. Dinh, P.A. Ormeno, K. Li, J. Marco, A lumped thermal model of lithium-ion battery cells considering radiative heat transfer, *Appl. Therm. Eng.* 143 (2018) 472–481, <http://dx.doi.org/10.1016/j.applthermaleng.2018.07.105>.
- [30] C. Sbarufatti, M. Corbetta, M. Giglio, F. Cadini, Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks, *J. Power Sources* 344 (2017) 128–140, <http://dx.doi.org/10.1016/j.jpowsour.2017.01.105>.
- [31] Z. Liu, H.-X. Li, A spatiotemporal estimation method for temperature distribution in lithium-ion batteries, *IEEE Trans. Ind. Informatics* 10 (4) (2014) 2300–2307, <http://dx.doi.org/10.1109/TII.2014.2341955>.
- [32] K. Liu, K. Li, Q. Peng, Y. Guo, L. Zhang, Data-driven hybrid internal temperature estimation approach for battery thermal management, in: W. Wang (Ed.), *Complexity* 2018 (1) (2018) 9642892, <http://dx.doi.org/10.1155/2018/9642892>.
- [33] F. Feng, S. Teng, K. Liu, J. Xie, Y. Xie, B. Liu, K. Li, Co-estimation of lithium-ion battery state of charge and state of temperature based on a hybrid electrochemical-thermal-neural-network model, *J. Power Sources* 455 (2020) 227935, <http://dx.doi.org/10.1016/j.jpowsour.2020.227935>.
- [34] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Trans. Neural Netw.* 20 (1) (2009) 61–80, <http://dx.doi.org/10.1109/TNN.2008.2005605>.
- [35] D. Miao, G. Duan, D. Chen, Y. Zhu, X. Zheng, Real-time temperature prediction for large-scale multi-core chips based on graph convolutional neural networks, *Electron. (Switzerland)* 14 (2025) <http://dx.doi.org/10.3390/electronics14061223>.
- [36] C. Chen, Y. Yuan, W. Sun, F. Zhao, Multivariate multi-step time series prediction of induction motor situation based on fused temporal-spatial features, *Int. J. Hydrog. Energy* 50 (2024) 1386–1394, <http://dx.doi.org/10.1016/j.ijhydene.2023.11.047>.
- [37] S.Y. Li, M.G. Fetene, Temperature prediction by graph convolutional network with transformer for lithium-ion battery to improve the thermal management system, *J. Energy Storage* 137 (2025) <http://dx.doi.org/10.1016/j.est.2025.118537>.
- [38] X. Zhao, B. Sun, W. Zhang, X. He, S. Ma, J. Zhang, Temperature field spatiotemporal modeling of lithium-ion battery pack configured sparse temperature sensors, *J. Energy Storage* 97 (2024) <http://dx.doi.org/10.1016/j.est.2024.113005>.
- [39] Z. Wang, F. Yang, Q. Xu, Y. Wang, H. Yan, M. Xie, Capacity estimation of lithium-ion batteries based on data aggregation and feature fusion via graph neural network, *Appl. Energy* 336 (2023) <http://dx.doi.org/10.1016/j.apenergy.2023.120808>.
- [40] X. Zheng, Z. Yuan, L. Tao, H. Wu, Aging-aware capacity prediction for lithium-ion batteries: A dynamic graph networks framework with spatiotemporal feature fusion, *Meas.: J. Int. Meas. Confed.* 258 (2026) <http://dx.doi.org/10.1016/j.measurement.2025.119005>.
- [41] J. Lee, S. Lee, Impact of busbar-induced joule heating on battery temperature rise in electric vehicle batteries, *Int. J. Automot. Technol.* (2025) <http://dx.doi.org/10.1007/s12239-025-00307-8>.
- [42] B. Sanchez-Lengeling, E. Reif, A. Pearce, A. Wiltschko, A gentle introduction to graph neural networks, *Distill* 6 (8) (2021) <http://dx.doi.org/10.23915/distill.00033>, URL <https://distill.pub/2021/gnn-intro>.
- [43] T. Fukumoto, Graph neural network for heat transfer problem, 2025, GitHub release v1.0.0, <https://github.com/matlab-deep-learning/Graph-Neural-Network-for-Heat-Transfer-Problem/releases/tag/v1.0.0>.