Case Studies in Thermal Engineering 61 (2024) 105002

Contents lists available at ScienceDirect

Case Studies in Thermal Engineering

journal homepage: www.elsevier.com/locate/csite

DeepTimeNet: A novel architecture for precise surface temperature estimation of lithium-ion batteries across diverse ambient conditions

Muhammad Hamza Zafar [a], Syed Muhammad Salman Bukhari [b], Mohamad Abou Houran [c], Majad Mansoor [d], Noman Mujeeb Khan [b], Filippo Sanfilippo [a,e,*]

[a] *Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway*
[b] *Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, 44000, Pakistan*
[c] *School of Electrical Engineering, Xi'an Jiaotong University, No. 28 , West Xianning Road, Xi'an, 710049, China*
[d] *Department of Automation, University of Science and Technology of China, Hefie, 230026, China*
[e] *Department of Software Engineering, Kaunas University of Technology, 44029 Kaunas, Lithuania*

ARTICLE INFO

Keywords:
Battery management systems
Surface temperature estimation
Lithium-ion batteries
Deep neural networks
Time-series analysis
Predictive modelling
Temperature-dependent performance

ABSTRACT

With the growing demand for battery-powered devices and electric vehicles, the need for improved battery performance and safety is paramount. A key determinant of battery health is the accurate monitoring of surface temperature (ST). Conventional ST estimation often depends on direct sensor measurements, which may not be cost-effective and can impact system reliability. This paper presents DeepTimeNet, a novel approach leveraging deep learning (DL) architectures for sensorless ST prediction in lithium-ion batteries. DeepTimeNet combines Convolutional Neural Networks (CNN), ResNet blocks, Inception modules, Bidirectional LSTM, and GRU layers to precisely model the time-dependent behaviour of batteries. A comprehensive evaluation against traditional models, across temperatures ranging from -20 °C to 25 °C and under various driving profiles, including US06 and Urban Dynamometer Driving Schedule (UDDS), is conducted. DeepTimeNet's performance is quantified by metrics such as mean absolute error (MAE), surpassing that of models like Gated Recurrent Unit-Recurrent Neural Network (GRU-RNN), Convolutional Neural Network-Long Short Term Memory Network (CNN-LSTM), and Long Short Term Memory Network (LSTM). The results demonstrate DeepTimeNet's superior performance, with an RMSE of 0.0971, MSE of 0.0099, MAE of 0.0912, and MAXE of 0.3963, validating it as an advanced tool for enhancing the efficacy of battery management systems and underscoring its potential as a benchmark for future innovations.

1. Introduction

Lithium-ion batteries are essential for powering modern devices and vehicles, from portable electronics to electric cars, due to their compact energy storage capabilities. However, maintaining their longevity, safety, and efficiency poses significant technical challenges, particularly in managing battery temperature. Temperature regulation in lithium-ion batteries is critical because it directly affects the battery's performance, efficiency, and safety. Improper temperature management can lead to reduced battery

[*] Corresponding author at: Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway.
E-mail address: filippo.sanfilippo@uia.no (F. Sanfilippo).

https://doi.org/10.1016/j.csite.2024.105002
Received 5 February 2024; Received in revised form 7 August 2024; Accepted 17 August 2024
Available online 22 August 2024

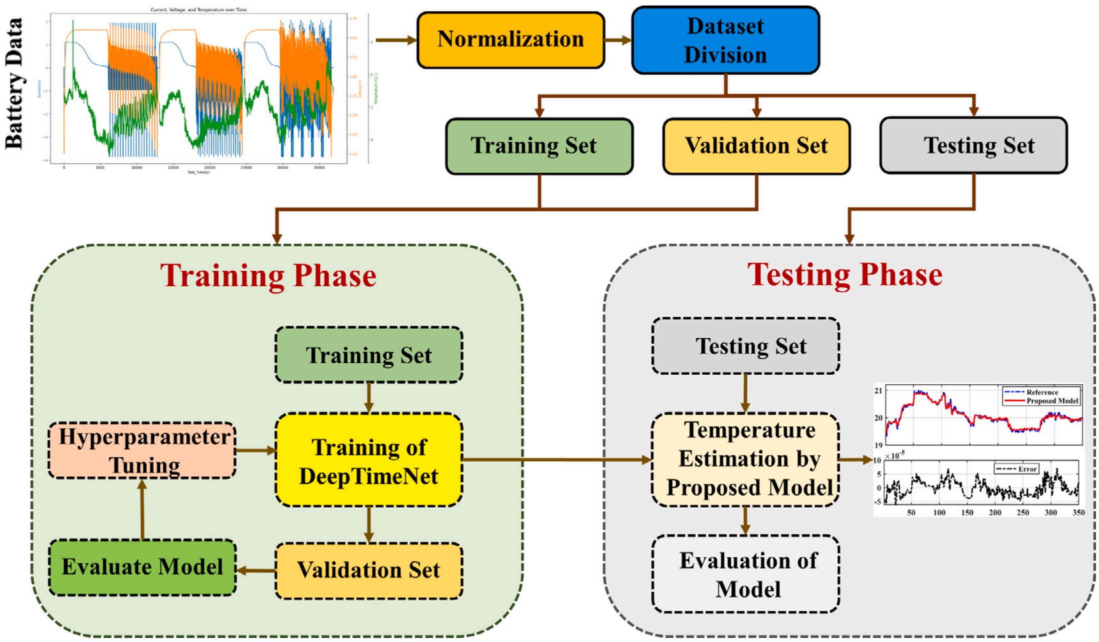

Fig. 1. Flow description of proposed DeepTimeNet based temperature estimation.

life, decreased performance, and in extreme cases, hazardous conditions including thermal runaway leading to fires or explosions. The task of monitoring and controlling the temperature thus falls under the critical operations of the Battery Management System (BMS), which must ensure that the battery operates within safe thermal thresholds across various usage scenarios such as charging, discharging, and during periods of stationary storage [1,2]. Maintaining optimal temperature conditions is complicated by the battery's inherent susceptibility to external temperature fluctuations, which can drastically alter its internal chemistry and, consequently, its performance. Traditional methods for monitoring the battery's surface temperature (ST) typically involve direct measurement using thermal sensors. These sensors, characterised by either positive or negative temperature coefficients (PTC or NTC), adjust their resistance in response to temperature changes, providing a voltage change that can be monitored to infer temperature [3]. However, these solutions face challenges in terms of cost, integration complexity, and potential reliability issues in extreme environmental conditions. Additionally, sensor-based approaches require significant calibration and maintenance, which can drive up costs and complicate system design [2]. Emerging sensorless techniques seek to overcome these limitations by employing indirect methods like electrochemical impedance spectroscopy to gauge temperature changes. Although innovative, these methods often necessitate advanced equipment and a comprehensive understanding of the battery's physical and chemical properties, which can restrict their practical use in various operational environments [4,5].

This paper introduces a novel approach to temperature estimation that leverages the advancements in computational capabilities and machine learning (ML) technologies. By treating the battery as a "black box", our methodology utilises deep learning (DL) techniques to model the complex relationships between various operational parameters and the battery's internal temperature. Our proposed model, DeepTimeNet, integrates multiple neural network architectures, including Convolutional Neural Networks (CNNs), Residual Networks (ResNet) blocks, Inception modules, and Gated Recurrent Units (GRUs), to predict the surface temperature of lithium-ion batteries effectively. DeepTimeNet is designed to operate sensorlessly, using data-driven approaches that overcome many of the limitations associated with traditional sensor-based systems. It provides a robust, efficient, and cost-effective solution for battery temperature monitoring, crucial for ensuring the safety and operational efficiency of lithium-ion batteries in various applications.

Fig. 1 provides a visual representation of the overall system flow. The system consists of multiple components, each playing a critical role in getting the final result. The workflow of the DeepTimeNet model for sensorless ST prediction in lithium-ion batteries involves collecting and normalising raw battery data, and then dividing it into training, validation, and testing sets. The Training Phase utilises deep learning architectures (CNN, ResNet, Inception, Bidirectional LSTM, GRU) to train and validate the model, followed by hyperparameter tuning. In the Testing Phase, the model's predictions are evaluated against reference values, achieving high accuracy with metrics RMSE, MSE, MAE, and MAXE.

1.1. Related work

Traditional temperature sensors, such as PTC or NTC sensors, adjust their resistance in response to temperature changes. These sensors are cost-effective and commonly used for direct monitoring of battery ST but often exhibit non-linear responses under

Table 1
Summary of recent studies on sensorless temperature estimation in lithium-ion batteries.

Ref.	Year	Methodology	Key Contributions	Gaps
[6]	2024	Neural network with physical modelling	Hybrid approach reduces sensor dependence and enhances real-time monitoring accuracy.	Requires initial calibration and validation against physical models.
[7]	2024	Square root cubature Kalman filter	Improves estimation accuracy and reliability under variable conditions.	Algorithm complexity may hinder real-time application.
[8]	2024	Review of sensorless techniques	Comprehensive overview of current sensorless technologies.	Lacks practical implementation strategies.
[9]	2024	Machine learning algorithms	Demonstrates effective prediction of battery health and operational parameters.	Limited to specific types of battery chemistries.
[10]	2024	Comparative ML analysis	Evaluates various ML techniques for battery management.	Does not address the integration of techniques into existing BMS.

extreme conditions, complicating temperature regulation and battery management tasks [3]. Moreover, digitally interfaced sensors, though they provide better integration with battery management systems via microprocessors, increase the system's complexity, cost, and the potential for reliability issues due to the need for frequent sensor calibration and maintenance [2]. Innovations in sensorless temperature estimation have emerged to address these limitations 1. For instance, Xiong et al. introduced a hybrid approach combining neural networks with physical battery models to reduce the dependence on direct sensors while enhancing real-time monitoring accuracy [6]. Although promising, this method depends heavily on accurate physical modelling, requiring extensive calibration and validation for different battery types and conditions. Expanding on this, Shen et al. employed advanced square root cubature Kalman filter algorithms to improve reliability under varied operational conditions, enhancing estimation accuracy [7]. However, the computational complexity and the need for precise parameter tuning make this method less practical in real-time applications. Demirci et al. reviewed methods leveraging electrochemical impedance spectroscopy for sensorless temperature estimation, providing detailed insights into battery conditions without traditional sensors [8]. Despite its sophistication, this technique demands a deep understanding of battery chemistry and specialised equipment, limiting its accessibility for widespread consumer applications. Further contributions by Das et al. and Oyucu et al. demonstrate the potential of machine learning algorithms, such as KNN and decision trees, in predicting battery health and operational parameters, including temperature [9,10]. Although these AI-based methods show promise, they require extensive datasets for training and may encounter issues with overfitting or underfitting, depending on the data's diversity and quality.

Building on these foundations, our work with DeepTimeNet offers a comprehensive and integrative approach, combining advanced neural network architectures—CNNs, ResNets, Inception modules, BiLSTMs, and GRUs. This fusion enhances the model's adaptability and accuracy in temperature predictions by leveraging complex pattern analysis from diverse data sources and improving real-time predictive capabilities. Notably, the use of ResNet blocks helps overcome the vanishing gradient problem, crucial for training deeper networks. Inception modules process data at multiple scales, adapting to sudden temperature changes effectively, while BiLSTM layers capture temporal dynamics crucial for trend prediction. GRUs balance the retention of essential historical information with the need to forget non-essential data, optimising the model's performance for sensorless temperature estimation. Thus, DeepTimeNet, operating without direct sensory inputs, reduces system complexity and cost, enhances reliability, and is ideally suited for environments where sensor deployment is impractical or too costly.

1.2. Motivation and contributions

With the surge in computational prowess and the proliferation of data, ML methodologies have found their way into battery system evaluations. A testament to this trend is the burgeoning discourse on ML-driven techniques for estimating the battery's state of charge (SOC) [11]. For instance, a back-propagation (BP) neural network has been employed for SOC estimation in [12], with the observed battery current, terminal voltage, and ambient temperature serving as its inputs, while the SOC functions as its output. Given the sequential nature of SOC estimation, the inherent architecture of recurrent neural networks (RNNs) is particularly apt. This is attributed to the RNN's distinct internal structure optimised for sequence-oriented tasks, making it a recurrent choice for SOC estimation, as observed in studies [13,14]. Beyond SOC, ML techniques have also been pivotal in evaluating metrics like the battery's state of health (SOH) and its remaining useful life (RUL) [15]. For instance, techniques grounded in support vector machines and Gaussian process regression have been proposed for simultaneous estimation of SOC and SOH [15].

The application of ML techniques has revolutionised battery analytics by treating the battery as a "black box". These methodologies harness voluminous datasets to decipher the battery's intricate internal dynamics, thereby establishing the nuanced non-linear relationships between input and output signals. This paradigm shift has rendered ML-based methods particularly conducive for tasks such as battery ST estimation, mirroring their utility in SOC estimation. One conventional approach proposed in the literature pivots on the artificial neural network (ANN) to execute sensorless ST estimations [1]. However, this methodology is not devoid of challenges. Primarily, it is universally acknowledged that temperature exhibits a tardy rate of change, implying that the present battery ST is inherently influenced by its preceding state. The inherent architecture of ANNs, however, cannot retain historical data. This necessitates the constant feeding of contemporary battery operational data to facilitate computations—a process that inadvertently compromises computational efficiency and elevates operational intricacies. To circumvent this limitation, the application of recurrent neural networks (RNNs) has been advocated for battery ST estimations [16]. Notably, this approach does not delve into a theoretical exploration of the interplay between the battery's temperature profile and neural network models. The intrinsic capability of RNNs to incorporate pivotal historical data for real-time state estimations renders them apt for battery ST assessments. This aligns with the overarching objective of bolstering sensorless methodologies and ameliorating the challenges delineated in preceding studies [1,5]. However, the traditional RNN – often dubbed the simple RNN (SRNN) – encounters impediments in discerning long-term dependencies, primarily due to the phenomena of vanishing gradients or sporadic gradient explosions during the backpropagation process [17]. To counteract these inherent limitations of the SRNN, a slew of sophisticated architectures has been proposed, including the Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional RNN (BiRNN) [13,17]. Among these avant-garde RNN variants, the GRU, conceived by Cho et al. in 2014 [18], merits special attention. It boasts a streamlined internal architecture and adeptly navigates the complexities of long-term sequential dependencies.

1.2.1. Contributions

In response to the challenges faced in advanced battery management systems (BMS) for electric vehicles and portable devices, this paper introduces DeepTimeNet, a novel, sensorless DL approach that significantly enhances temperature prediction accuracy. By addressing the limitations of traditional sensor-based methods, which are often complex and costly, DeepTimeNet provides a sophisticated, scalable solution that excels in diverse environments, thereby reducing maintenance requirements and operational costs. The specific contributions of this paper are as follows:

1. We investigate the theoretical foundations linking battery ST estimation to advanced neural network architectures. This investigation explores how the time-series nature of battery temperature data can be effectively modelled using a combination of CNNs, ResNets, Inception modules, BiLSTM, and GRU layers, setting the stage for the architectural choices in DeepTimeNet.
2. The development and introduction of DeepTimeNet, which integrates these neural network technologies into a cohesive framework. This architecture leverages ResNet blocks to mitigate the vanishing gradient problem, Inception modules to handle data at multiple scales for greater adaptability, BiLSTM layers for capturing temporal patterns, and GRUs for optimising data retention and discarding, thereby enhancing the accuracy and reliability of ST predictions.
3. Empirical validation of DeepTimeNet against diverse driving profiles and temperature conditions, ranging from −20 °C to 25 °C. The performance assessment includes robust metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Square Error (MSE), and Maximum Absolute Error (MAXE). This evaluation not only demonstrates DeepTimeNet's exceptional accuracy and versatility but also its capability to maintain consistent performance across varying environmental conditions.
4. A detailed analysis of DeepTimeNet's performance, highlighting its reliability and robustness compared to traditional methods. The model's uncertainty analysis reveals minimal error discrepancies, confirming its efficacy even under strenuous operational conditions.

These contributions highlight DeepTimeNet's potential to transform temperature estimation in BMS, leading to more reliable and cost-effective solutions for the fast-evolving electric vehicle and portable device markets.

Paper Organisation: The remainder of the paper is structured as follows: Section 2 provides a detailed analysis of the battery surface temperature time series. Section 3 describes the dataset, preprocessing techniques, and data analysis. The proposed model, DeepTimeNet, is elaborated in Section 4, followed by a comparative results discussion in Section 5. The paper concludes with final remarks in Section 6.

2. Battery temperature time series analysis

Lithium-ion batteries, fundamental to a myriad of modern applications, often grapple with heat generation during operation. Analysing and understanding this heat generation is pivotal for enhancing the efficiency and longevity of the battery. The dynamics of heat generation and its implications on battery temperature, especially in a time series context, form the essence of this section. The process of heat generation within lithium-ion batteries, while complex, can be captured through a relatively simplified mathematical expression. The quintessential equation representing this phenomenon is:

$$Q = I(V_B - OCV) \tag{1}$$

This formula, initially proposed by Bernardi et al. [19], serves as a foundational model in battery thermodynamics studies. The constituents of this equation can be delineated as follows:

The symbol Q represents the quantum of heat generated within the battery. I stands for the current flowing through the battery, serving as a direct indicator of the battery's operational intensity. The term V_B denotes the terminal voltage of the battery, reflecting the voltage level when the battery is actively delivering power. Lastly, OCV epitomises the open-circuit voltage, which is the voltage level observed when the battery remains disconnected from any load or external circuit.

The term $I(V_B - OCV)$ in the heat generation equation embodies the primary sources of heat within the battery:

1. R_{ohmic}: This term signifies the heat generated due to the battery's inherent ohmic resistance. Every battery, by its construction and materials, offers some resistance to the flow of current. This resistance, when coupled with the operational current, results in heat generation, often referred to as Joule heating.
2. $Z_{polarisation}$: This encapsulates the heat evolved due to charge transfer overpotential. As lithium ions move between the anode and cathode during charge or discharge, there is a potential difference or overpotential that arises. This phenomenon, coupled with the resistance offered by the electrolyte and other internal components, leads to additional heat generation.

The open-circuit voltage, OCV, is intrinsically linked to the battery's state of charge (SOC). The SOC measures the current battery capacity as a percentage of its nominal capacity. As the battery charges or discharges, the SOC varies, and this variation is mirrored in the OCV. By monitoring and understanding this relationship, one can glean insights into the battery's operational state and its implications on heat generation. A holistic thermal analysis of batteries, while rich in insights, is often marred by the intricacies and complexities involved. To make this endeavour more tractable and to abstract away some of the underlying nuances, certain assumptions are postulated. These assumptions, while simplifying the analysis, aim to capture the dominant thermal behaviours within the battery. Here, it elucidates the primary assumptions:

1. **Uniform Heat Distribution**:
   One of the foundational postulations is that the heat generated within the battery, regardless of its source or intensity, disperses uniformly across the cell. This implies that every infinitesimal section of the battery experiences the same quantum of heat. Such an assumption is rooted in the belief that the internal materials and construction of the battery facilitate even heat distribution, preventing the formation of localised 'hot' or 'cold' spots. While this might not always mirror the exact real-world scenario, especially in batteries with manufacturing defects or those subjected to extreme conditions, it offers a reasonable approximation for general thermal modelling.
2. **Uniform Battery Surface Temperature (ST)**:
   Extending the premise of uniform heat distribution, it is further hypothesised that the battery's surface, the interface between the battery and its external environment, maintains a consistent temperature. This means that whether one measures the temperature at the top, bottom, or sides of the battery, the readings remain invariant. Such an assumption is vital for models that leverage surface temperature as a proxy for the internal state of the battery. It is predicated on the belief that the battery's outer casing and internal components work in tandem to ensure a homogenised temperature profile on the surface.

While these assumptions streamline the thermal modelling process, it is crucial to recognise their simplifying nature. In certain specialised applications or under specific conditions, deviations from these assumptions might arise, necessitating more intricate models or corrective measures. To fathom the intricacies of battery thermal behaviour, it is indispensable to acquaint oneself with the pivotal parameters that govern its thermal dynamics. These parameters, both intrinsic and extrinsic, play a decisive role in shaping the thermal profile of the battery. A deep dive into these parameters is presented below:

- **Heat Capacity of the Battery Core**, C_c:
  The core of the battery, being its primary operational hub, possesses a certain capacity to store heat. Denoted by C_c, this parameter reflects the amount of heat required to change the core's temperature by a unit degree. It is a testament to the battery's inherent ability to absorb and store heat, a critical trait that determines its resilience to thermal shocks and its cooling requirements.
- **Heat Capacity of the Battery Surface**, C_s:
  Analogous to the core's heat capacity, the surface of the battery, denoted by C_s, also has a characteristic heat capacity. While it might be influenced by the core's heat dynamics, the surface heat capacity also incorporates the effects of external factors, such as ambient conditions and external cooling or heating mechanisms.
- **Battery Core Temperature**, T_c:
  At any given point in time, the core of the battery, being the epicenter of its operations, registers a specific temperature, denoted by T_c. This temperature, influenced by both internal processes (like chemical reactions) and external factors, plays a pivotal role in determining the battery's operational efficiency and lifespan.
- **Battery Surface Temperature**, T_s:
  The temperature at the battery's surface, represented by T_s, is not just a mere reflection of the core temperature. It is an amalgamation of the core's thermal dynamics and the interplay of external factors, such as ambient temperature, cooling mechanisms, and the battery's placement within a device or vehicle.
- **Ambient Temperature**, T_a:
  The surrounding or external temperature, termed the ambient temperature and denoted by T_a, exerts a profound influence on the battery's thermal behaviour. Whether it is the cold of a wintry night or the scorching heat of a summer day, the ambient conditions can amplify or attenuate the battery's internal thermal dynamics, making T_a a crucial parameter to monitor and manage.
- **Core-to-Surface Thermal Resistance**, R_{cs}:
  The resistance offered to heat flow from the battery's core to its surface is symbolised by R_{cs}. It encapsulates the inherent resistive properties of the battery's materials and construction. A lower R_{cs} indicates a more efficient transfer of heat from the core to the surface, facilitating rapid cooling or heating, while a higher value suggests potential thermal bottlenecks.

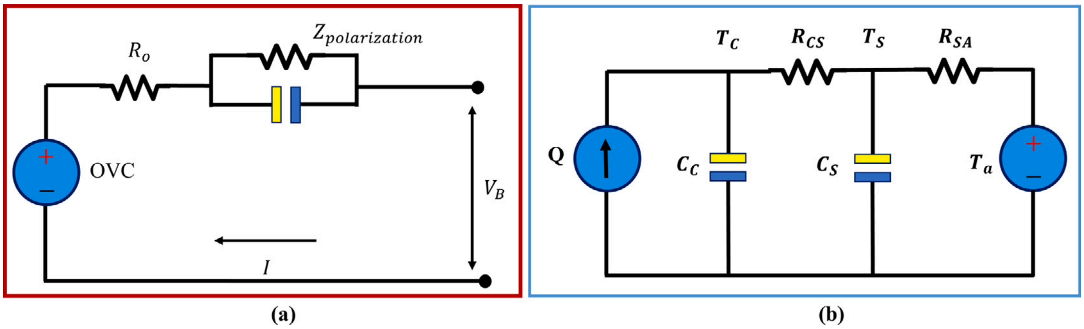

Fig. 2. (a) Battery equivalent circuit electric model. (b) Battery equivalent circuit thermal model.

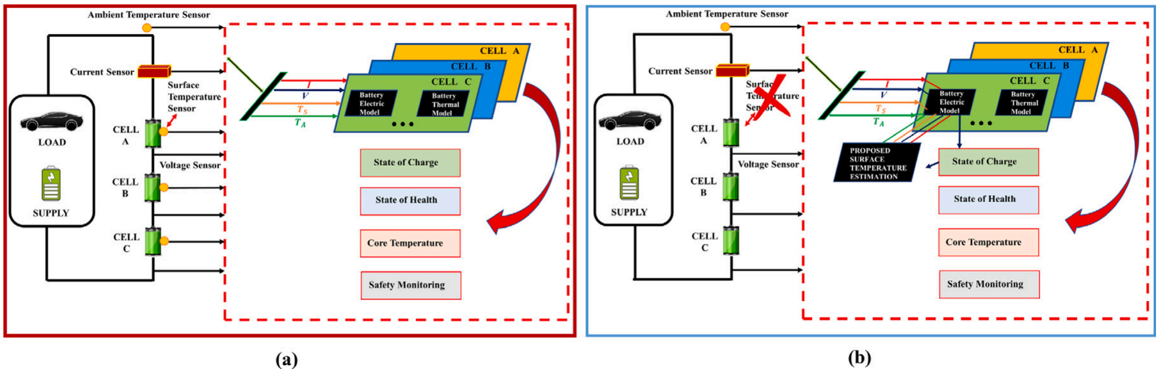

Fig. 3. Battery management system.

- **Surface-to-Ambient Thermal Resistance**, R_{sa}:
  Analogous to R_{cs}, the R_{sa} parameter denotes the resistance encountered by heat as it flows from the battery's surface to its ambient surroundings. It is influenced by factors like the battery's casing, external insulations, and the medium (air, liquid, etc.) surrounding the battery.

Understanding these parameters and their interplay is paramount for devising effective thermal management strategies and optimising battery performance, longevity, and safety (see Table 2).

3. Dataset description

This research leverages publicly available datasets to evaluate the DeepTimeNet model's efficacy in estimating the surface temperature (ST) of lithium-ion batteries. An extensive review has compiled a selection of prominent lithium-ion battery testing datasets [20]. The study utilises two distinct types of cylindrical lithium-ion cells, specifically A123 18650 with $LiFePO_4$ chemistry and Panasonic 18650 with $LiNiCoAlO_2$ chemistry. These selections ensure a broad spectrum of battery materials is represented, providing a robust test for the adaptability of the DeepTimeNet model. The testing apparatus includes a high-precision battery tester, an environmental chamber to simulate various climate conditions and a computing workstation armed with specialised software for data acquisition and analysis. A temperature sensor is meticulously affixed to the epicenter of the battery cell's surface using thermal paste and adhesive tape, allowing for the accurate measurement of the cell's surface temperature [21]. The first dataset, procured from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland, simulates realistic electric vehicle (EV) driving conditions by incorporating the Urban Dynamometer Driving Schedule (UDDS) and the US06 Highway Driving Schedule. This dataset spans a comprehensive range of ambient temperatures, specifically 0 °C, 10 °C, 25 °C, and 40 °C, to elucidate the environmental temperature's impact on battery behaviour. The second dataset, provided by the University of Wisconsin-Madison, focuses on dynamic ST estimation across ambient temperatures from −20 °C to 20 °C, challenging the DeepTimeNet model with varying driving schedules, including UDDS, LA92, US06, and HWFET (see Figs. 2–5).

3.1. *Dataset pre-processing:*

The essence of data-driven models, especially in machine learning, lies in the quality and consistency of the data they are fed. While raw data often encapsulates the intricate patterns and nuances that models seek to capture, its unprocessed nature can lead

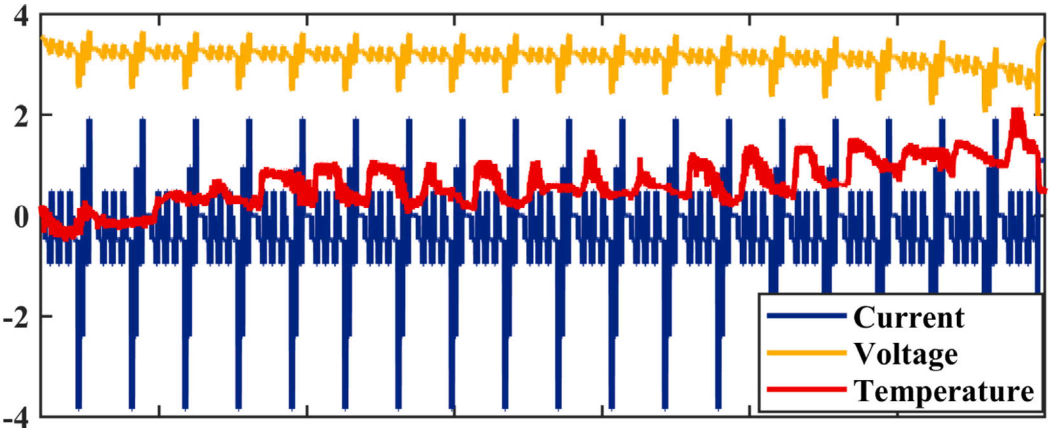

Fig. 4. Voltage, current and temperature variation at 0 degree.

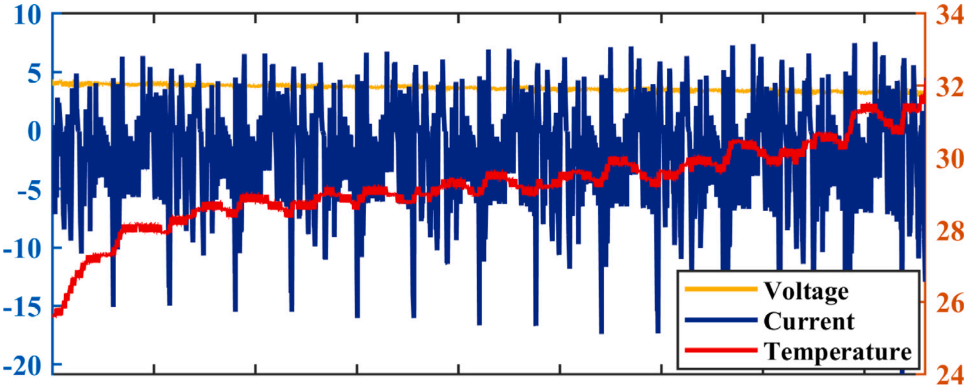

Fig. 5. Voltage, current and temperature variation at 25 degree.

Table 2
Characteristics of Various Energy Sources.

Source	Type	Material	Voltage (V)	Current (Ah)
Dataset 1	A123 18650	$LiFePO_4$	2.0–3.6	1.1
Dataset 2	Panasonic 18650	$LiNiCoAlO_2$	2.5–4.2	2.9

to a slew of challenges. Features in raw datasets, originating from various sources or measurements, can span vastly different scales or units. This disparity, if left unchecked, can skew the model's learning process, with features having larger magnitudes disproportionately influencing the model. To address this, data normalisation emerges as an indispensable tool, ensuring a level playing field for all features. When disparate features coexist in a dataset, machine learning algorithms, especially those leveraging gradient descent optimisation techniques, can experience challenges:

- **Divergent Learning Rates**: Features spanning larger scales can cause the learning algorithm to oscillate, leading to longer convergence times.
- **Feature Domination**: Features with broader value ranges can overshadow and dominate the learning process, rendering other features negligible.
- **Numerical Instabilities**: Algorithms can face computational challenges, especially when dealing with extremely large or small values.

Normalisation mitigates these issues, harmonising the scales and ensuring efficient and balanced learning. Among the myriad of normalisation techniques, min–max normalisation stands out for its simplicity and effectiveness. It linearly rescales each feature, ensuring that its values lie within a predefined range, typically between 0 and 1 [21–23]. The mathematical embodiment of this transformation is:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

In this context, x' denotes the normalised value, x is the original data point, and $\min(x)$ and $\max(x)$ correspond to the smallest and largest values of the feature across the dataset, respectively. By anchoring the entire spectrum of a feature's values to a 0 to 1 range, min–max normalisation ensures that the relative distances and relationships between the data points remain preserved. It is worth noting that while min–max normalisation is effective, it is sensitive to outliers, as extreme values can distort the scaling. For the drive cycle datasets employed in this research, min–max normalisation was judiciously applied to all features. By ensuring a consistent scale, the transformed dataset sets the stage for the machine learning algorithms to discern patterns, relationships, and intricacies without any feature-induced bias. This meticulous pre-processing was pivotal in ensuring robust, unbiased, and effective learning by the subsequent models.

3.2. Dataset analysis:

To facilitate the training of the DeepTimeNet model, a structured data preparation approach, commonly employed in deep learning realms, is put into action. To ensure the model has a comprehensive understanding and can generalise well on unseen data, we partition the experimental dataset as follows:

- **Training Set (60%)**: A bulk of the data, constituting 60%, is allocated for training. This dataset aids the model in learning the intricate patterns and relationships within the data.
- **Validation Set (15%)**: To monitor and prevent overfitting during the training process, 15% of the data is set aside for validation. This subset plays a pivotal role in hyperparameter tuning and offers insights into the model's performance during training iterations.
- **Test Set (25%)**: Post the training phase, to gauge the efficacy of the DeepTimeNet model in real-world scenarios, 25% of the data is earmarked for testing. This data is unseen by the model during its training and validation stages, ensuring an unbiased evaluation of its predictive prowess.

Before embarking on the training voyage, it is imperative to condition the data, making it amenable to deep learning. In line with this, the datasets are meticulously pre-processed, undergoing normalisation using the min–max technique, ensuring all features resonate on a consistent scale, and setting the stage for efficient and robust learning by the DeepTimeNet model.

4. Proposed model

DeepTimeNet emerges as a beacon in the realm of battery temperature estimation, bridging the gap between traditional monitoring methods and the demands of modern-day applications. Designed to capture the intricate temporal patterns of battery temperature evolution, this model intertwines the power of deep learning with the nuances of time series analysis. In this section, it elucidates the architecture, data processing techniques, and rationale behind DeepTimeNet.

4.1. Overview

- The evolving landscape of battery thermal performance monitoring and the pressing need for accurate, efficient, and cost-effective solutions.
- Limitations of traditional methods, emphasising increased costs and reduced reliability with the growth in the number of battery cells or strings.
- The proposition of DeepTimeNet as a state-of-the-art solution to surmount these challenges.

4.2. DeepTimeNet architecture

DeepTimeNet is a harmonious blend of depth and breadth, tailored specifically to handle the intricacies of time-series data, particularly the thermal patterns of batteries. The following sections elucidate its key components:

- **Overall Architecture**: DeepTimeNet stands distinct with its multi-layered architecture. By leveraging a deep architecture, the network can capture both low-level and high-level features from the data, ensuring that the temperature estimates are both accurate and reliable.
- **Conv1D Layers**: These layers serve as the foundational building blocks of DeepTimeNet. By employing one-dimensional convolutional layers, the model effectively captures spatial patterns in the input data. This is especially crucial in understanding the temperature gradients across different battery cells.
- **ResNet Blocks**: The resilience of DeepTimeNet can be attributed to its use of ResNet blocks. These blocks employ skip connections to bypass certain layers, thus enabling the network to learn identity functions. This mechanism prevents the vanishing gradient problem, allowing for deeper networks without compromising on performance. The incorporated dropout layers within these blocks further bolster the model's robustness by preventing overfitting.
- **Inception Modules**: A unique facet of DeepTimeNet is its incorporation of Inception modules. These modules capture features at multiple scales by using convolutional operations of different kernel sizes simultaneously. Consequently, the model gains the ability to recognise both granular and broader patterns in the temperature data.

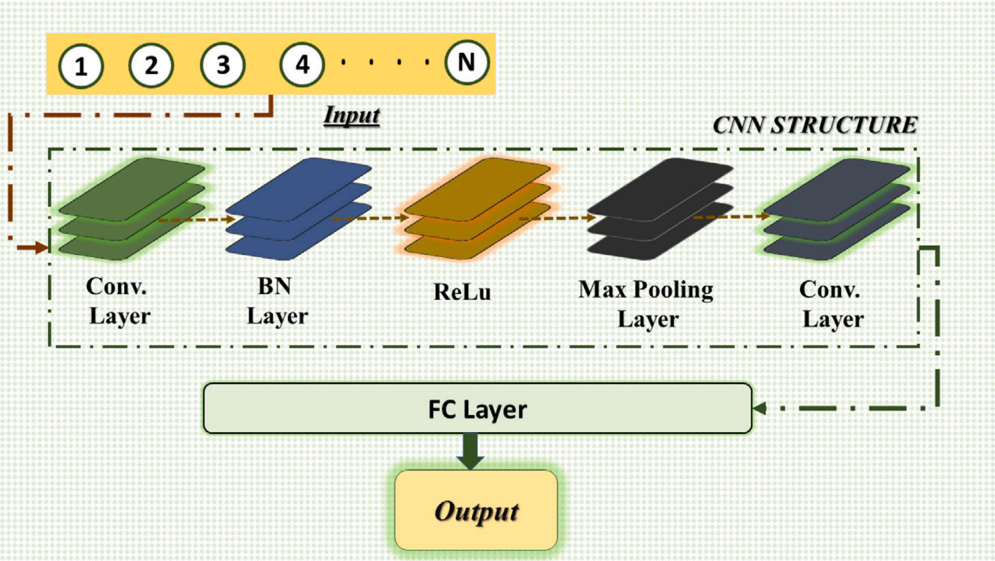

Fig. 6. CNN-1D complete architecture.

- **Bidirectional LSTM and GRU Layers**: Recognising the temporal nature of battery temperature data, DeepTimeNet harnesses the power of both LSTM and GRU layers in a bidirectional manner. This bidirectionality enables the model to capture dependencies from both past and future time steps, leading to a more comprehensive understanding of the data's temporal dynamics.

4.3. Conv1D layer

The Conv1D layer, representing one-dimensional convolution, stands as a cornerstone in the domain of sequential data processing within deep learning frameworks. While 2D convolution layers are tailored for image data matrices, Conv1D is adept at gleaning temporal or sequential characteristics [24]. It achieves this by sliding a learnable filter across a unidimensional sequence. Conceptually, the Conv1D operation involves a filter – a collection of learnable parameters – and moves it across the length of the input sequence. At every juncture, it carries out an element-wise multiplication with the overlaid input segment and then sums up the results. This meticulous approach is primed to discern and amplify localised patterns or temporal intricacies within the sequence, proving invaluable for deciphering the complex temporal dynamics pervasive in various real-world datasets. The one-dimensional convolution operation, as delineated by the Conv1D layer, is visually encapsulated in Fig. 6. As evident from the figure, the Conv1D layer processes the input sequence through a series of filters, each designed to extract specific temporal features. These filters slide across the input, producing corresponding feature maps that highlight patterns or temporal nuances identified by each filter. The resulting feature maps, as shown, serve as transformed representations of the original sequence, emphasising certain localised patterns. This transformation is crucial for subsequent layers in the deep learning model to recognise and act upon these highlighted features. The spatial positioning of these features within the sequence, as well as their relative intensities, can be vital clues for tasks such as anomaly detection, sequence classification, or prediction. The core operation underlying a Conv1D layer is the one-dimensional discrete convolution. This mathematical procedure facilitates the extraction of local or temporal patterns within sequences, making it a fundamental tool in the sequential data processing. Given a 1D input sequence $x[n]$, where n is the discrete time index, and a filter $h[k]$, where k is the discrete filter index, the discrete convolution operation is defined as:

$$y[m] = (x * h)[m] = \sum_{n=-\infty}^{\infty} x[n] \cdot h[m-n] \tag{3}$$

In the context of Conv1D layers in neural networks:

- The sequences and filters are of finite length, thus restricting the infinite summation to the lengths of the sequences.
- A bias term is often introduced, adding an offset to the convolution result.
- An activation function is applied element-wise to the output, introducing non-linearity to the model.

This can be represented as:

$$y[m] = \sigma\left(\sum_{n=0}^{N-1} x[n] \cdot h[m-n] + b\right) \tag{4}$$

Where σ denotes the activation function, such as ReLU, sigmoid, or tanh; b represents the bias term, and N is the length of the input sequence. For applications within the realm of neural networks, convolutional operations are not limited to a singular filter. Instead, they are typically executed using an array of filters, each producing its distinct output feature map. This multifaceted approach enriches the model's capacity to discern and capture a gamut of features and patterns from the input data. Further intricacies in the practical implementation of convolutional operations arise from considerations such as the introduction of stride and padding. The stride defines the step size when the filter traverses the input sequence, influencing the spatial dimensions of the resulting feature map. On the other hand, padding pertains to the methodology adopted at the sequence boundaries during the convolution, ensuring consistent feature map dimensions and preserving information at the edges. The culmination of these operations – convolution with multiple filters, strategic striding, and judicious padding – enables the extraction of both localised and position-invariant features. In essence, each feature map acts as a lens, highlighting different characteristics or nuances embedded within the input sequence, thereby enriching the overall representation.

4.4. ResNet block

Residual Networks (ResNet), with their ingenious architecture, have revolutionised the training dynamics of profoundly deep neural networks [25]. Instead of aspiring to learn an underlying holistic function, ResNet pivots its focus towards learning the residual or the difference, thus bridging the gap between the expected and predicted outputs. This is actualised through the introduction of “skip connections” or ”shortcut connections” which essentially allow the signal to bypass one or more layers, fortifying the network against the vanishing gradient problem.

The foundational principle of ResNet lies in its unorthodox approach [26]. In traditional networks, each layer strives to transform its input data into a more abstract representation. However, in ResNet, each layer is tasked with learning the residual or difference from its input, aiming to zero out the errors or discrepancies.

4.4.1. Mathematical representation of ResNet

The Residual Network (ResNet) introduces a novel architectural design to alleviate the challenges of training deep neural networks. The central idea of ResNet is the introduction of skip or shortcut connections that facilitate learning residuals or differences, as opposed to attempting to learn the entire transformation. This approach aids in mitigating the vanishing and exploding gradient problems commonly encountered in deep networks.

The foundational equation representing the residual learning is given by:

$$y = F(x, \{W_i\}) + x \tag{5}$$

Where y is the output of the residual block, x denotes the original input, and $F(x, \{W_i\})$ symbolises the residual mapping to be learned. This encompasses the weight layers of the block which attempts to model the residual or error.

A crucial insight here is that if identity mapping is the optimal function, it is easier to push F to output zero values, thus reverting to the identity than to learn the identity transformation directly.

In cases where the dimensions of x and $F(x, \{W_i\})$ do not match, a linear projection W_s is applied to x to match the dimensions. This is mathematically represented as:

$$y = F(x, \{W_i\}) + W_s x \tag{6}$$

This ensures consistent dimensions for the element-wise addition, maintaining the integrity of the architecture. The ability of ResNet to allow gradients to back-propagate through these skip connections ensures that the network can be deep without sacrificing training efficacy.

Within the architectural blueprint of DeepTimeNet, ResNet blocks champion several pivotal roles:

1. **Combatting the Vanishing Gradient Problem**: A notorious bottleneck for deep networks is the vanishing gradient dilemma. As networks deepen, gradients – essential for weight updates – risk diminishing to inconsequential values, stalling the learning process. ResNet blocks, with their skip connections, offer a direct pathway for gradients, effectively circumventing this impediment.
2. **Empowering Depth in Architecture**: With the vanishing gradient problem deftly addressed, DeepTimeNet is liberated to explore the depths of architectural complexity. Such depth is instrumental in capturing nuanced, hierarchical patterns within data.
3. **Optimal Feature Preservation**: The ingenuity of skip connections extends beyond gradient flow. They ensure that as data journeys through the network, critical features from antecedent layers are not obliviously discarded but are seamlessly integrated, enriching the feature space.
4. **Facilitated Training**: ResNet blocks, with their ability to learn residuals, often result in smoother and more well-behaved loss landscapes, making the optimisation process more manageable.

4.5. Inception module

The Inception module, originating from Google's groundbreaking Inception (or GoogLeNet) architecture, embodies a paradigm shift in convolutional neural networks [27]. By espousing a heterogeneous kernel strategy, it allows networks to concurrently discern features at multiple granularities. This heterogeneous approach facilitates the extraction of spatial patterns at varied resolutions, enabling the model to develop a holistic understanding of the data it processes.

A unique facet of the Inception module is its parallelism. Traditional convolutional layers, in their pursuit of feature extraction, often face the dilemma of choosing between fine, detailed features (via small kernels) and broader, more abstract features (via large kernels). The Inception module resolves this conundrum by adopting both simultaneously [27]. It amalgamates diverse kernel sizes to capture both local details and wider contextual information, ensuring a comprehensive feature map.

4.5.1. Mathematical representation

The Inception module, a paradigm of architectural finesse, seamlessly intertwines convolutional operations across diverse kernel dimensions with pooling. This strategic ensemble endows the module with the prowess to discern features across a spectrum of scales, ranging from intricate granular details to broader, overarching patterns.

The multifarious operations within the Inception module can be mathematically delineated as:

$$I(x) = \text{Concatenate}\big(C_1(x), C_3(x), C_5(x), P_3(x)\big) \tag{7}$$

where:

$$C_1(x) = \text{Conv1D}_{1\times 1}(x) \tag{8}$$

$$C_3(x) = \text{Conv1D}_{3\times 3}(x) \tag{9}$$

$$C_5(x) = \text{Conv1D}_{5\times 5}(x) \tag{10}$$

$$P_3(x) = \text{MaxPool}_{3\times 3}(x) \tag{11}$$

To elucidate, the term $C_k(x)$ signifies a one-dimensional convolutional operation with a kernel of size k. The diversity in kernel sizes facilitates the extraction of features spanning various spatial extents. For instance, while $C_1(x)$ hones in on minute, localised patterns, $C_5(x)$ aggregates a broader swathe of information. On the other hand, $P_3(x)$ is emblematic of the module's commitment to resilience. This max-pooling operation sifts through the features, cherry-picking the most salient ones within its 3×3 purview. Beyond its role in dimensionality reduction, this operation infuses the model with a degree of translational invariance, bolstering its robustness against minor positional perturbations.

The resulting output, $I(x)$, emerges as a composite feature mosaic, harmonising insights across various spatial scales and complexities. The Inception module's kaleidoscopic approach stands as a testament to its indispensable role in modern deep-learning architectures.

The integration of Inception modules within DeepTimeNet is strategic, offering a multitude of advantages:

1. **Hierarchical Feature Extraction**: DeepTimeNet, armed with the Inception modules, is empowered to perceive and learn from data at multiple spatial hierarchies. This diversity in feature extraction ensures that the model is attuned to both nuanced intricacies and overarching patterns.
2. **Augmented Network Breadth without Excessive Depth**: The Inception module offers a lateral expansion, broadening the network's horizon of features. This obviates the need for excessive depth, ensuring efficient computation without compromising on the richness of feature extraction.
3. **Robustness against Overfitting**: Incorporating pooling operations, the Inception module introduces a level of abstraction and spatial invariance, thereby reducing the model's propensity to overfit on training data, ensuring generalisability across unseen datasets.

4.6. Bidirectional LSTM

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have been specially designed to overcome the challenges posed by standard RNNs [28,29]. Their architectural nuances, such as gates and cell states, enable them to efficiently capture long-range dependencies in sequences, mitigating issues like the vanishing gradient problem. The "Bidirectional" LSTM amplifies this capability by processing the sequence data in both forward and reverse directions [30]. This bi-directional processing ensures that the information at each time step is enriched with knowledge from both its past and future states, making it a powerful tool for sequential data analysis.

In the context of sequential data, the Bidirectional Long Short-Term Memory (BiLSTM) plays a pivotal role in capturing temporal dependencies from both past and future states. The architecture of a BiLSTM, as depicted in Fig. 7, leverages two LSTM layers that process the sequence data in both forward and reverse directions. This dual-direction processing ensures comprehensive learning of the sequence's temporal dynamics.

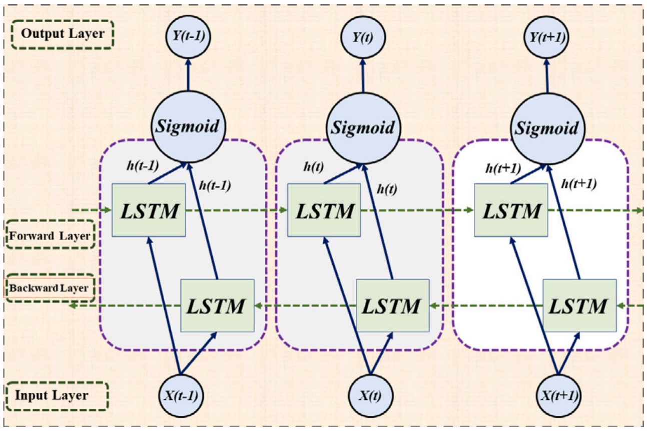

Fig. 7. Schematic representation of the Bidirectional Long Short-Term Memory (BiLSTM) architecture.

4.6.1. LSTM architecture

Long Short-Term Memory (LSTM) units, a seminal innovation in the realm of recurrent neural networks, were conceived to adeptly capture long-term dependencies in sequence data, a feat that traditional RNNs struggled with [31]. The core strength of LSTMs lies in their intricate cell structure, meticulously designed to regulate the flow and manipulation of information. Central to the LSTM's design are specialised structures known as gates. These gates, akin to information processing valves, determine the magnitude and nature of data flow in the network. The LSTM cell encompasses:

- **Input Gate** (i_t): This gate scrutinises incoming information and decides the quantum of it that should be written to the cell state. Essentially, it is the gateway controlling new information influx.
- **Forget Gate** (f_t): As the name suggests, this gate dictates which portions of the cell state are deemed irrelevant and should be discarded or "forgotten" to make way for new, pertinent information.
- **Output Gate** (o_t): Post the updates to the cell state, this gate determines the extent of information to be relayed to the hidden state, which then proceeds to subsequent layers or time steps.
- **Cell State** (c_t): Often analogised to a conveyor belt, this is the LSTM's memory store, ferrying information across time steps, with gates modulating the data it carries.

The mathematical choreography governing the LSTM's intricate dance of information is captured in the following equations:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{12}$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{13}$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \tag{14}$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{15}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{16}$$

$$h_t = o_t \odot \tanh(c_t) \tag{17}$$

In the given LSTM context, the symbols i_t, f_t, o_t, and g_t are the computational representations of the input, forget, output gates, and the cell input respectively. Meanwhile, c_t and h_t serve as mathematical formalisms for the cell and hidden states. Additionally, σ denotes the sigmoid activation function, which ensures gate values lie between 0 and 1. The symbol $\odot$ represents element-wise multiplication, a pivotal mechanism that allows the LSTM to regulate the flow of information effectively. In essence, the LSTM's unique architecture, governed by these equations, grants it the prowess to learn and remember over extended time horizons, making it an invaluable tool for sequential data modelling.

4.6.2. Bidirectionality in LSTM

Traditional LSTMs process sequences from the start to the end. However, many sequential tasks benefit from the knowledge of future data points for a more contextual understanding. This is where Bidirectional LSTMs come into play. In a Bidirectional LSTM, two LSTMs are trained on the input sequence. The first processes the sequence as-is (forward), and the second processes it in reverse. Their separate representations are then concatenated or combined in a manner suitable for the task at hand. This dual processing ensures that each time step in the sequence is seen in its full context, with information from both its past and future states. The advantage of this bidirectionality is evident in tasks where the context is crucial. For instance, in an endeavour with DeepTimeNet, understanding a battery's thermal performance is not just about its past but also potential future fluctuations. This holistic view aids in more accurate and context-aware predictions.

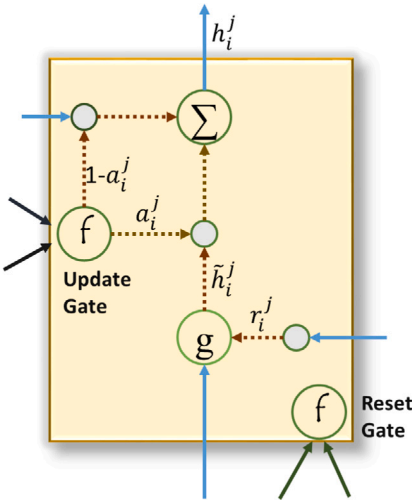

Fig. 8. Schematic representation of the Gated Recurrent Unit (GRU).

4.7. Gated recurrent unit (GRU)

Gated Recurrent Units, or GRUs, are a type of recurrent neural network architecture introduced as a streamlined version of the LSTM [32]. Despite having fewer parameters and a simpler structure, GRUs retain much of the power of LSTMs in modelling long-range dependencies in sequence data. This is achieved through a clever gating mechanism that combines the functionalities of the input and forget gates of an LSTM into a single update gate in a GRU. Another significant difference is the absence of a separate cell state; in GRUs, the hidden state serves this purpose. The Gated Recurrent Unit (GRU) is an efficient variant of the LSTM, optimised to use fewer parameters while maintaining the ability to capture long-range dependencies in sequences. As illustrated in Fig. 8, the GRU employs a unique gating mechanism, combining the forget and input gates into a single update gate, thereby simplifying the architecture and enhancing computational efficiency.

The Gated Recurrent Unit (GRU), introduced as a streamlined alternative to the LSTM, has risen in popularity owing to its simplified structure while retaining a commendable capacity to model temporal dependencies. The GRU's strength lies in its efficient gating mechanisms which adeptly regulate the flow of information, ensuring that the model can capture both short-term and long-term patterns in sequential data.

The GRU's architecture revolves around two pivotal gates:

- **Update Gate** (z_t): Serving as a balancer, this gate weighs the importance of retaining the previous hidden state and integrating new information from the current input. In essence, it decides the degree of 'memory update' at each time step.
- **Reset Gate** (r_t): This gate modulates the influence of the previous hidden state when computing the potential new memory content. It plays a pivotal role in deciding how much of the past information should be forgotten or retained while assimilating the new input.

The operations within a GRU are elegantly captured by the following mathematical formulations:

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \tag{18}$$

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \tag{19}$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{t-1} + b_{hn})) \tag{20}$$

$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{t-1} \tag{21}$$

To elucidate the employed symbols within the context of the GRU: z_t and r_t are the computational representations of the update and reset gates. The symbol n_t serves as a depiction of the potential new memory content, which is an aggregate of insights derived from both the current input and the preceding hidden state. The term h_t signifies the hidden state representation at the time instance t, amalgamating past information with newly acquired insights. Further, σ is indicative of the sigmoid activation function, ensuring that the generated outputs are confined between 0 and 1. The symbol $\odot$ represents element-wise multiplication, a pivotal mechanism that underpins the GRU's regulation of data flow. GRU, with its efficient design and gating mechanisms, offers a robust model for sequential data analysis, adeptly balancing computational efficiency with representational power. In the DeepTimeNet framework, the utilisation of both Bidirectional LSTM and GRU layers encapsulates a dual advantage. The LSTM, with its intricate gating mechanisms, excels in capturing long-term dependencies in sequences, while the GRU, with its more compact structure, offers computational efficiency. The bidirectionality further ensures that the sequence data is understood in a holistic context,

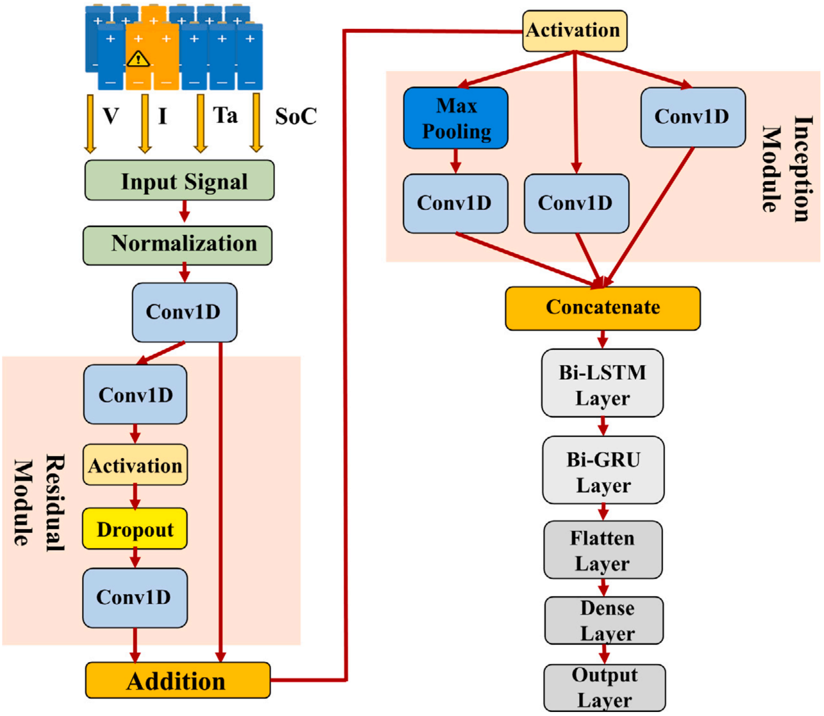

Fig. 9. Comprehensive architecture of the DeepTimeNet for surface temperature prediction.

with insights gleaned from both past occurrences and potential future developments. This combination makes the DeepTimeNet a formidable architecture for sequential data analysis, especially in tasks demanding a fine balance between computational efficiency and predictive prowess.

The DeepTimeNet architecture, as shown in Fig. 9, is an intricate tapestry of modern neural network designs, meticulously chosen for their prowess in handling time-series data. Beginning with the Conv1D layers, which lay the groundwork by sifting through the raw sequence data to unearth preliminary patterns, it then escalates to the advanced constructs of ResNet blocks and Inception modules [27]. These components work synergistically, capturing spatial hierarchies, ensuring depth, and broadening the network's perspective by gleaning features at varying scales. Culminating this ensemble are the bidirectional LSTM and GRU layers, which serve as the temporal backbone of DeepTimeNet. Their advanced gating mechanisms and bidirectional processing ensure the model's acumen in understanding both short-term nuances and long-term dependencies within the data. Together, these components coalesce to form a formidable predictive model, optimised not only in its depth and width but also in its temporal sensitivity, making DeepTimeNet a state-of-the-art solution for intricate time-series forecasting tasks.

5. Results

The results section serves as the cornerstone of this research, presenting empirical findings derived from the proposed model and corresponding datasets. It is here that the efficacy, robustness, and adaptability of the model are scrutinised, with insights gleaned from various evaluation metrics. In this light, it commences by outlining key evaluation metrics, providing both mathematical formulations and interpretative context, before diving deep into the experimental outcomes.

5.1. Evaluation metrics

The efficacy of predictive models is critically dependent on robust evaluation criteria that quantify the accuracy and reliability of their outputs. In this study, it adopts several key performance indicators to assess the DeepTimeNet model's capacity to estimate surface temperatures accurately. Root Mean Square Error (RMSE) provides a measure of the model's prediction error magnitude, reflecting the square root of the average squared differences between the estimated and actual values. Mean Absolute Error (MAE) offers a linear scale of the average absolute errors, presenting a straightforward interpretation of prediction accuracy. Mean Square Error (MSE) captures the variance in estimation errors, and Maximum Absolute Error (MAXE) identifies the largest single error across the dataset. Together, these metrics form a comprehensive framework for evaluating the model's performance, enabling a nuanced understanding of its predictive power and potential limitations. The subsequent analysis delineates the model's performance across these metrics, facilitating a rigorous validation of its estimation proficiency under varying operational conditions.

5.1.1. Root mean squared error (RMSE)

Root Mean Squared Error (RMSE) is a popular metric used to evaluate the accuracy of a model's predictions. It measures the average magnitude of the errors between predicted and observed values. Mathematically, RMSE is defined as the square root of the average squared differences between predicted ($\hat{y}$) and actual values (y):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{22}$$

Specifically, for the Root Mean Square Error (RMSE) metric, n represents the total number of observations, $\hat{y}_i$ denotes the predicted value for the i^{th} observation, and y_i symbolises the actual value for the i^{th} observation. The RMSE gives a higher weight to larger errors, making it more sensitive to outliers. A lower RMSE value indicates a better fit of the model to the data, implying more accurate predictions.

5.1.2. Mean squared error:

Another pivotal metric employed to gauge the precision of the model's predictions is the Mean Squared Error (MSE). MSE is instrumental in computing the average squared deviations between the predicted and actual values, thus providing an in-depth perspective into the variance of the model's prediction errors. Mathematically, MSE is articulated as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{23}$$

In the context of this equation, n represents the total number of observations, $\hat{y}_i$ is the model's prediction for the i^{th} observation, and y_i is the corresponding actual value. A diminutive MSE value is indicative of the model's predictions aligning closely with the real outcomes, whereas an inflated MSE suggests a pronounced discrepancy in the model's predictive accuracy.

5.1.3. Mean absolute error:

A fundamental metric that facilitates the evaluation of the model's predictions is the Mean Absolute Error (MAE). MAE quantifies the average of the absolute differences between the predicted and true values, offering a lucid comprehension of the model's prediction accuracy in terms of its average error magnitude. Expressed mathematically, MAE is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{24}$$

Within the framework of this equation, n denotes the aggregate number of observations, $\hat{y}_i$ signifies the model's prediction for the i^{th} observation, and y_i corresponds to the actual value of the same observation. A lower MAE underscores the model's adeptness in generating predictions that closely mirror the true values, while a higher MAE reveals potential areas of improvement in the model's predictive capacity.

5.1.4. Maximum absolute error:

To glean insights into the upper bound of the prediction errors, the Maximum Absolute Error (MAXE) is employed as an evaluative metric. MAXE spotlights the largest absolute deviation between the predicted and actual values, granting a perspective on the worst-case error scenario. The mathematical formulation of MAXE is:

$$\text{MAXE} = \max_i |\hat{y}_i - y_i| \tag{25}$$

Navigating through the constituents of this equation, $\hat{y}_i$ represents the prediction of the model for the i^{th} observation, and y_i is the respective actual value. A smaller MAXE value is indicative of the model's robustness, ensuring that even in its least accurate predictions, the deviation is within acceptable bounds. Conversely, a large MAXE suggests that there might be certain instances where the model's predictions significantly deviate from the actual outcomes, warranting further investigation or refinement.

5.2. Comparative analysis at dataset 1

The main focus of this part is a careful side-by-side analysis of the suggested model, DeepTimeNet, versus several modern architectures. The critical problem of predicting battery surface temperatures over a range of temperature situations serves as the foundation for the analysis. RMSE, MSE, MAE, and MAXE are among the evaluation metrics use to deliver a thorough and objective assessment. These measures provide a broad overview of each model's predictive capacity, accuracy, and dependability. As seen in Table 3, the information provides useful insights on the models' relative performance.

Upon dissecting the results from Table 3, it is evident that DeepTimeNet manifests unparallelled efficacy. This is particularly pronounced in the −10 °C scenario, where model's RMSE value of 0.099 substantially trumps the values recorded by its contemporaries. Such distinctions are not confined to a solitary metric or temperature scenario. Throughout the temperature spectrum, DeepTimeNet consistently outperforms, registering lower RMSE, MSE, and MAE values, while maintaining competitive MAXE scores. Take, for instance, the 25 °C scenario. DeepTimeNet's MAE value of 0.058 is a testament to its precision, underscoring a notable edge over other models. This is further corroborated by its RMSE and MSE values in the same scenario. A similar pattern emerges across other temperature scenarios, cementing DeepTimeNet's superiority. While the competing models – GRU-RNN, CNN-LSTM, and LSTM –

Table 3
Evaluation metrics for different temperature scenarios for Dataset 1.

Temperature	Metric	Models			
		Proposed Model	GRU-RNN	CNN-LSTM	LSTM
0 °C	RMSE	0.099	0.158	0.159	0.231
	MSE	0.009	0.024	0.025	0.053
	MAE	0.091	0.122	0.128	0.210
	MAXE	0.319	0.440	0.451	0.469
10 °C	RMSE	0.090	0.158	0.167	0.191
	MSE	0.008	0.025	0.028	0.036
	MAE	0.087	0.124	0.139	0.187
	MAXE	0.428	0.600	0.781	0.619
20 °C	RMSE	0.079	0.084	0.111	0.131
	MSE	0.006	0.007	0.012	0.017
	MAE	0.058	0.066	0.098	0.699
	MAXE	0.210	0.261	0.351	0.329
40 °C	RMSE	0.091	0.134	0.139	0.189
	MSE	0.008	0.017	0.019	0.035
	MAE	0.090	0.106	0.125	0.210
	MAXE	0.401	0.447	0.471	0.499

display commendable performances, they occasionally falter, especially when juxtaposed against the comprehensive capabilities of DeepTimeNet. Their occasional spikes in error metrics across certain temperature scenarios underscore potential areas of refinement. In summation, the empirical data paints a compelling narrative: DeepTimeNet is not just another model in the battery temperature estimation arena; it is a trailblazer, setting benchmarks and redefining the paradigms of accuracy and reliability. The results not only accentuate its prowess but also spotlight its potential to revolutionise modern battery management systems.

Fig. 10 provides an in-depth look at the performance of the DeepTimeNet model on Dataset 1, showcasing ST estimations and errors across a temperature range from 0 °C to 40 °C. Subplots (a) and (b) display the model's ST estimations at 0 °C and 10 °C, respectively. The proposed model (solid red line) closely follows the reference (dashed blue line) in subplot (a), indicating high accuracy in ST estimation at lower temperatures. Notably, the error plot (c) below shows that discrepancies are within a narrow margin, generally lying within ± 0.1 degrees, showcasing the precision of the model's estimations. In subplot (b), at a slightly higher temperature of 10 °C, the model again demonstrates excellent tracking of the reference temperature profile with minimal deviation, as evidenced by the accompanying error subplot (d), which indicates error values predominantly clustering around the zero line.

Subplots (e) and (f) examine the model's estimations at 20 °C and 40 °C. The performance at 20 °C, as seen in subplot (e), reveals a well-matched pattern between the proposed model and the reference, with the error plot (g) underscoring the consistent accuracy with errors mostly contained within $\pm 5 \times 10^{-5}$ degrees. The model's capability at 40 °C, presented in subplot (f), is particularly noteworthy. Despite the challenging higher temperature, the proposed model's estimations continue to mirror the reference closely, a testament to its robustness and reliability in extreme conditions. The error subplot (h) confirms the model's precision with errors showing no significant increase compared to lower temperatures. Overall, the results from Dataset 1, as illustrated in Fig. 10, affirm the DeepTimeNet model's adeptness at providing precise ST estimations across a wide temperature range. The model maintains a low error magnitude, demonstrating its suitability for applications where accurate temperature estimations are critical for the performance and safety of battery systems.

5.3. Comparative analysis at dataset 2

This paragraph aims to provide light on the adaptability, robustness, and resilience of the suggested model, DeepTimeNet, by turning analytical lens on Dataset 2. Dataset 2's distinct dynamics and characteristics are inherent to it and offer a sophisticated framework for closely examining and contrasting the models' performance measures at different temperatures. It utilises the evaluative compass of RMSE, MSE, MAE, and MAXE metrics, just as it did with the prior dataset, in an effort to obtain a comprehensive picture of each model's prediction ability.

Table 4, the empirical narrative aligns with observations from Dataset 1. DeepTimeNet, proposed model, towers in performance, substantiating its supremacy in battery surface temperature estimation. A standout observation is its performance under the challenging −10 °C regime. With an RMSE of 0.069 and an MAE of 0.061, DeepTimeNet not only outstrips its contemporaries but does so with a marked margin, underscoring its adeptness in capturing intricate temperature dynamics. A similar pattern unfolds across other temperature scenarios. For instance, at 0 °C, the model's RMSE and MAE values of 0.161 and 0.150, respectively, further embolden its position as the frontrunner in predictive accuracy. The consistent performance across diverse temperature scenarios and datasets reaffirms DeepTimeNet's robust generalisation capabilities. The comparative models, GRU-RNN, CNN-LSTM, and LSTM, while demonstrating commendable performances in certain areas, occasionally exhibit discrepancies, especially when benchmarked against DeepTimeNet's holistic performance. Such instances spotlight the inherent strengths of model's architectural choices and its adaptability to different datasets. To encapsulate, the findings from Dataset 2 not only mirror the superior performance of DeepTimeNet observed in Dataset 1 but also reinforce its stature as a vanguard in the domain of battery temperature estimation.

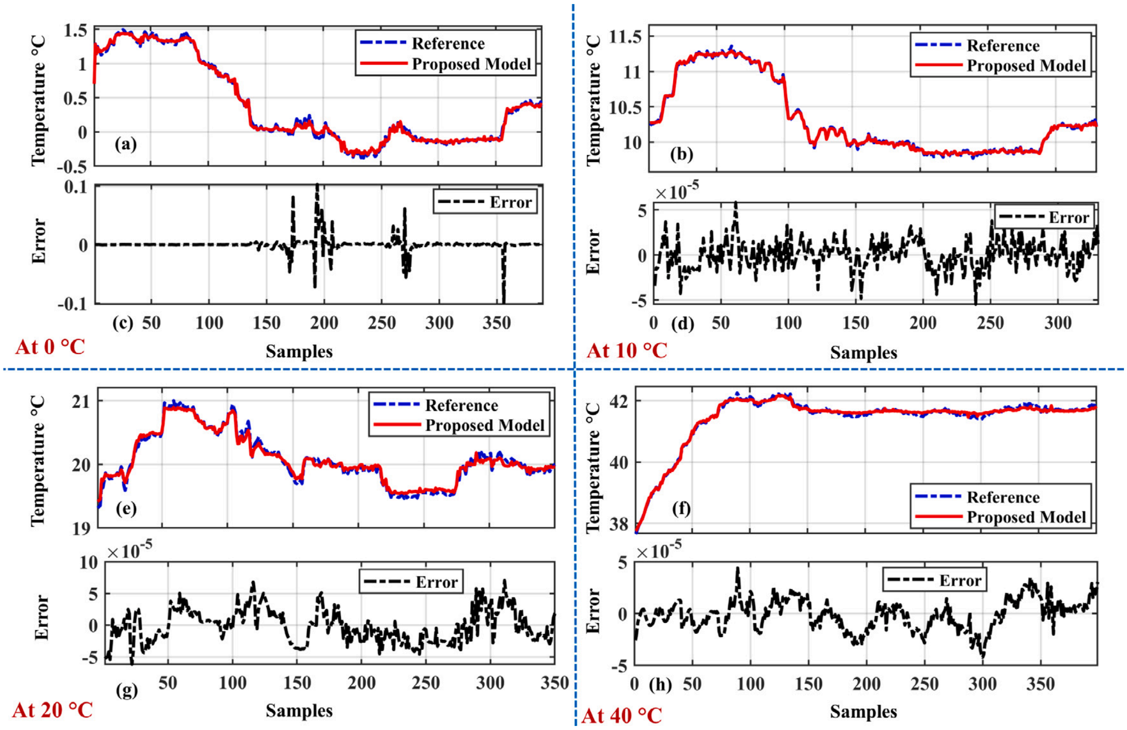

Fig. 10. Comprehensive estimation analysis of the proposed DeepTimeNet model on Dataset 1: Assessing Model Accuracy Across a Temperature Gradient. Subplot (a) illustrates ST estimations and subplot (c) details the corresponding error at 0 °C. Subplots (b) and (d) present the model's performance and error distribution at 10 °C. For 20 °C, subplots (e) and (g) depict the estimation fidelity and error analysis, respectively. Lastly, subplots (f) and (h) showcase the estimation precision and error magnitude at the elevated temperature of 40 °C, demonstrating the model's robustness at higher operational temperatures.

Table 4
Evaluation metrics for different temperature scenarios for UDDS drive cycle in Dataset 2.

Temperature	Metric	Models			
		Proposed Model	GRU-RNN	CNN-LSTM	LSTM
−20 °C	RMSE	0.069	0.088	0.099	0.131
	MSE	0.004	0.008	0.009	0.017
	MAE	0.061	0.074	0.098	0.110
	MAXE	0.281	0.306	0.351	0.399
−10 °C	RMSE	0.161	0.260	0.277	0.292
	MSE	0.025	0.067	0.076	0.085
	MAE	0.150	0.213	0.299	0.297
	MAXE	0.718	0.873	0.909	0.990
0 °C	RMSE	0.092	0.178	0.190	0.201
	MSE	0.008	0.031	0.036	0.040
	MAE	0.090	0.144	0.198	0.210
	MAXE	0.329	0.495	0.551	0.609
25 °C	RMSE	0.089	0.183	0.199	0.299
	MSE	0.007	0.033	0.039	0.089
	MAE	0.097	0.141	0.198	0.210
	MAXE	0.390	0.651	0.751	0.729

The results attest to its robustness, adaptability, and unparallelled precision, making it an asset in the ever-evolving landscape of battery management systems.

The comprehensive evaluation of the DeepTimeNet model's performance for temperature estimation under the US06 drive cycle is summarised in Table 5. The table presents a comparative analysis of various metrics, including Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), and Maximum Absolute Error (MAXE), across four different temperature scenarios ranging from −20 °C to 25 °C. At the lower temperature spectrum of −20 °C, the Proposed Model demonstrates remarkable precision with the lowest RMSE (0.065) and MAXE (0.270), outperforming the other considered models such as GRU-RNN, CNN-LSTM, and LSTM. This trend of superior performance by the Proposed Model is consistent across all temperatures, as evidenced by the consistently lower error metrics when compared to its counterparts. Particularly noteworthy is the model's robustness at

Table 5
Evaluation metrics for different temperature scenarios for US06 drive cycle in Dataset 2.

Temperature	Metric	Models			
		Proposed Model	GRU-RNN	CNN-LSTM	LSTM
−20 °C	RMSE	0.065	0.084	0.095	0.123
	MSE	0.0042	0.007	0.0091	0.015
	MAE	0.058	0.072	0.091	0.104
	MAXE	0.270	0.295	0.340	0.385
−10 °C	RMSE	0.157	0.252	0.271	0.284
	MSE	0.0246	0.063	0.073	0.081
	MAE	0.145	0.205	0.290	0.288
	MAXE	0.710	0.860	0.900	0.975
0 °C	RMSE	0.088	0.172	0.185	0.196
	MSE	0.0077	0.0296	0.0342	0.0384
	MAE	0.085	0.139	0.192	0.205
	MAXE	0.320	0.480	0.540	0.600
25 °C	RMSE	0.085	0.177	0.194	0.290
	MSE	0.0072	0.0314	0.0376	0.0841
	MAE	0.082	0.137	0.190	0.203
	MAXE	0.380	0.640	0.740	0.720

−10 °C, where the RMSE of 0.157 is significantly lower than that of the GRU-RNN model (0.252), suggesting enhanced reliability in near-freezing conditions. The model's adeptness is further highlighted at 0 °C, maintaining a low MSE (0.0077), which is indicative of its consistent performance even at standard operating temperatures for electric vehicles. The error analysis at a higher operational temperature of 25 °C reveals that the Proposed Model maintains its accuracy with the lowest RMSE (0.085) and a competitive MAXE (0.380), reinforcing the model's applicability across a diverse temperature range. The minimal deviation in MAE across the temperatures underscores the model's capability to offer precise estimations, a critical attribute for advanced battery management systems. Overall, the evaluation metrics indicate that the DeepTimeNet model not only achieves a high degree of accuracy but also showcases substantial improvements over traditional models. This solidifies its potential to serve as a reliable tool in optimising the performance and safety of battery systems in real-world scenarios.

In a comparative assessment of the DeepTimeNet model's performance across different driving cycles and temperatures within Dataset 2, Fig. 11 demonstrates its proficiency in surface temperature (ST) estimation. The model was evaluated at two sub-zero temperatures, specifically -20 °C and -10 °C, under both the UDDS and US06 driving profiles. At -20 °C, under the UDDS driving cycle (subplots a and c), the proposed model closely mirrors the reference ST with a remarkable level of precision, as evident from the minimal error values. Conversely, when subjected to the more aggressive US06 driving cycle (subplots b and d), the model maintains commendable accuracy, albeit with slightly more variation in error. This is indicative of the model's robustness, capable of adapting to the demands of high-speed and high-acceleration driving conditions while preserving accuracy in ST estimation. As the operational temperature is slightly increased to −10 °C, the model's performance remains consistent under the UDDS profile (subplots e and g), showcasing a steadfast adherence to the reference ST with error values that are well-contained. Under the US06 profile at the same temperature (subplots f and h), the model exhibits a steady increase in ST estimation, parallelling the reference trajectory. The error plot indicates that the model adeptly handles the dynamic driving conditions imposed by the US06 cycle, with the error remaining tightly bounded. The overall comparative analysis underscores the DeepTimeNet model's capabilities in delivering precise ST estimates across varied driving profiles and temperatures. The model's resilience in maintaining low error margins despite the increased demands of the US06 driving cycle, and the temperature variations, is a testament to its suitability for real-time applications in advanced battery management systems. Such performance is crucial for ensuring the longevity and safety of battery-operated devices, particularly in electric vehicles operating under a wide range of environmental conditions.

In a detailed analysis of the DeepTimeNet model's estimation capabilities presented in Fig. 12, we observe its performance under two distinct temperature regimes—0 °C and 25 °C—across both UDDS and US06 drive cycles for Dataset 2. The model's estimations at 0 °C within the UDDS drive cycle (subplots a and c) show a high fidelity to the reference data, with the proposed model tracking the reference temperature closely, despite a few transient spikes. The error plot for this temperature (subplot c) indicates a tight error distribution around the zero line, suggesting that any deviations from the reference are minor and the model's predictions are stable. At the same 0 °C temperature but under the more demanding US06 drive cycle (subplots b and d), the model demonstrates an incremental increase in the surface temperature estimation, again closely following the reference trajectory. The error analysis (subplot d) reflects a higher frequency of fluctuations, which is to be expected given the aggressive acceleration and speed conditions characteristic of the US06 profile. Nonetheless, the errors remain predominantly near the baseline, underscoring the model's robustness. The model's performance at a warmer operational temperature of 25 °C under the UDDS cycle (subplots e and g) is characterised by a greater degree of variability in both the temperature estimations and the corresponding errors. While the proposed model generally adheres to the reference (subplot e), there are noticeable spikes in the estimated temperature, which are reflected in the broader spread of error values (subplot g). This suggests a sensitivity to higher temperatures, though the model still manages to maintain a reasonable error margin. Under the US06 drive cycle at 25 °C (subplots f and h), the model's estimations exhibit a gradual and consistent increase over the duration of the samples, mirroring the reference with a high degree of accuracy.

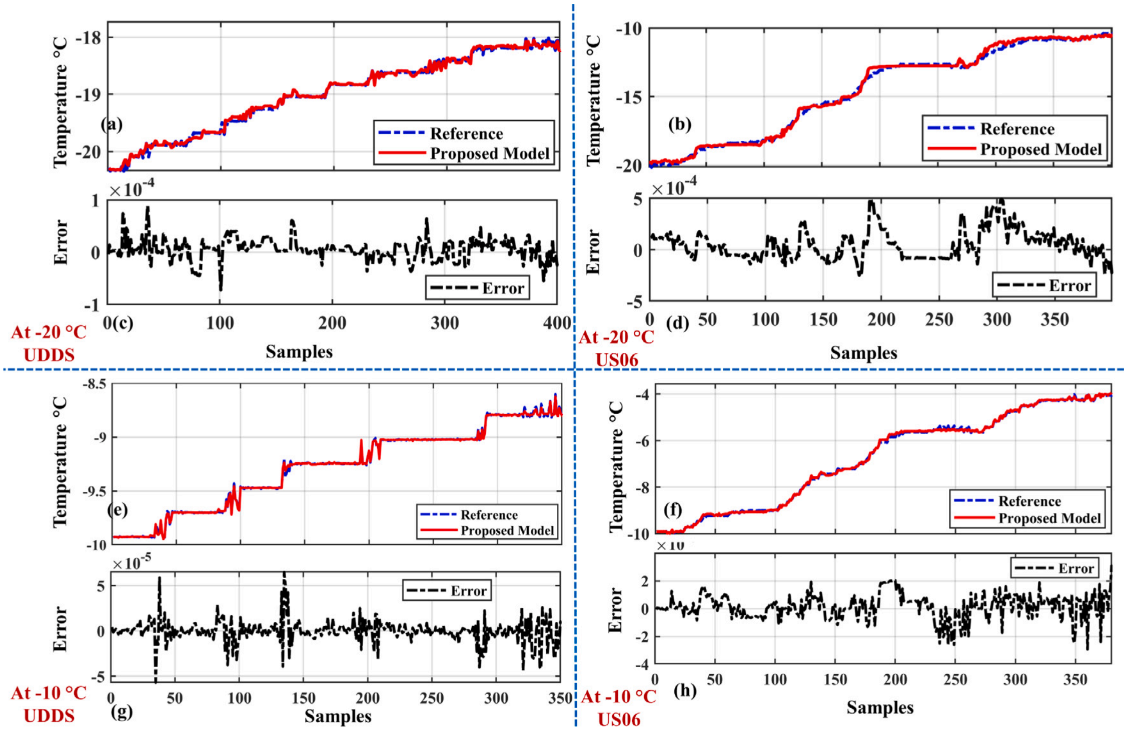

Fig. 11. Comparative estimation analysis of the proposed technique on Dataset 2 for UDDS and US06 drive cycles at −20 °C and -10 °C.

The error plot (subplot h) presents a distribution that is wider than that observed at 0 °C, indicating that the model's predictions are slightly more affected by temperature variations at this elevated temperature within the dynamic driving conditions of the US06 cycle. Overall, Fig. 12 solidifies the DeepTimeNet model's capability to deliver accurate surface temperature estimations across varying temperatures and driving conditions. The consistency in the model's performance, particularly its capacity to maintain accuracy under different thermal and driving stressors, emphasises its potential applicability in real-world battery management systems, where such diverse conditions are commonplace.

Throughout the results section, it has meticulously analysed the DeepTimeNet model's performance in estimating battery surface temperature (ST) under a variety of conditions encapsulated within Dataset 1 and Dataset 2. Figs. 10, 11 and 12 provide a visual representation of the model's estimations against a set reference across different temperatures and driving cycles, namely UDDS and US06. The model demonstrates exceptional accuracy at sub-zero temperatures, with estimations at −20 °C and −10 °C closely mirroring the reference values, as shown in Fig. 10. This level of precision is maintained across both driving cycles, indicating the model's robust adaptability to changes in driving dynamics. As it extends analysis to 0 °C and 25 °C, represented in Fig. 12, the model persists in displaying a high degree of fidelity to the reference ST, albeit with slight increases in error variance at the higher temperature during the more aggressive US06 cycle. The error plots consistently exhibit a distribution centred near the zero-error baseline, with marginal deviation even under the challenging US06 conditions. This substantiates the model's capability to provide reliable ST estimations—a critical factor for the operational safety and efficiency of battery systems. The collated evaluation metrics, detailed in Table 3, Table 4, Table 5 further corroborate the superiority of the DeepTimeNet model over traditional models like GRU-RNN, CNN-LSTM, and LSTM, particularly highlighted by its lower RMSE and MAXE values across all examined scenarios. The culmination of these findings attests to the DeepTimeNet model's advanced predictive performance, cementing its potential for integration into sophisticated battery management systems where accurate temperature predictions are paramount to the system's longevity and safety.

5.4. Comparative analysis of predictive models

The quest for optimising battery management systems pivots on the axis of precise surface temperature (ST) prediction. As such, the development and assessment of predictive models are of paramount importance. In this context, we have conducted an exhaustive comparative study, the results of which are consolidated in Table 6. This study meticulously evaluates the performance of our Proposed Model, DeepTimeNet, about several established models within the domain of lithium-ion battery ST estimation.

DeepTimeNet emerges as a paradigm-shifting model, outperforming its contemporaries across all the fundamental metrics: RMSE, MSE, MAE, and MAXE. The RMSE of 0.0971 reflects DeepTimeNet's exceptional precision in temperature estimation, a stark contrast to the RMSE of 0.52 exhibited by the CNN-LSTM model. The MSE value of 0.0099 stands as a testament to the consistent accuracy

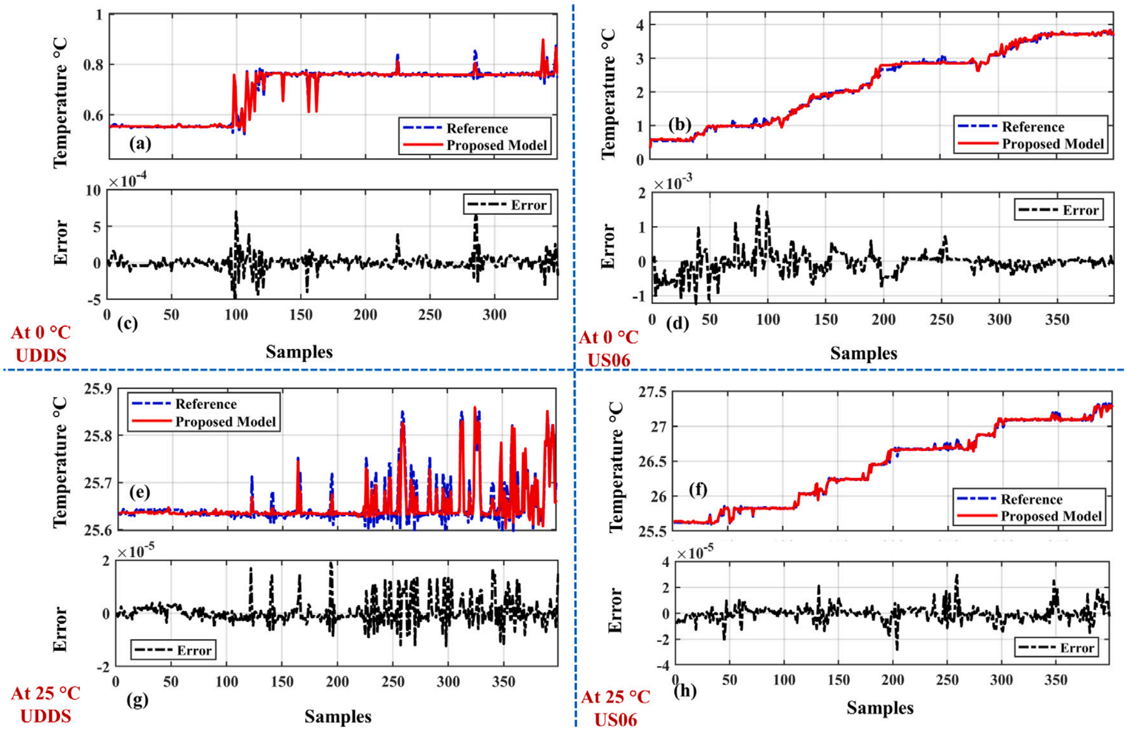

Fig. 12. Comparative estimation analysis of the proposed technique on Dataset 2 for UDDS and US06 drive cycles at 0 °C and 25 °C.

Table 6
Average Performance Metrics Comparison Across Studies.

Reference	Model	RMSE	MSE	MAE	MAXE
Proposed Model	DeepTimeNet	**0.0971**	**0.0099**	**0.0912**	**0.3963**
[33]	PINN	N/A	N/A	0.11	0.47
[34]	LSTM-PINN	0.5	N/A	N/A	N/A
[35]	GRU-RNN	0.155	N/A	0.121	0.43
[36]	CNN-LSTM	0.52	0.2704	0.45	1.22

of DeepTimeNet, as opposed to the significantly higher MSE of 0.2704 reported for the CNN-LSTM. When it comes to MAE, DeepTimeNet's average error margin is narrowly confined to 0.0912, thus providing a superior edge over the GRU-RNN model, which records an MAE of 0.121 as per the findings of Yao et al. (2022). Furthermore, the robustness of DeepTimeNet in managing peak error scenarios is underscored by an MAXE of 0.3963, far more favourable than the 1.22 maximum error encountered by the CNN-LSTM model. The collected data from the comparative study not only bolsters the technical preeminence of DeepTimeNet but also emphasises the ingenuity of its architecture—a confluence of CNN, ResNet blocks, Inception modules, and bidirectional LSTM and GRU layers. The amalgamation of these advanced neural network structures facilitates a model that is highly adept at capturing the complex dynamics of battery ST under a wide array of operating conditions. In light of the empirical evidence, DeepTimeNet is poised to redefine the benchmarks within the field of battery management systems, carving a niche as the quintessential tool for reliable ST prediction. The implications of this study resonate with the aspirations of advancing predictive modelling for lithium-ion batteries, marking a significant milestone in the pursuit of technological excellence in energy management and sustainability.

6. Conclusion

The narrative of this research underscores the inception, fine-tuning, and deployment of DeepTimeNet, a pioneering deep-learning architecture, tailored for the rigorous task of battery surface temperature (ST) estimation. At the heart of DeepTimeNet lies a harmonious fusion of Gated Recurrent Units (GRU), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), ResNet blocks, Inception modules, and Bidirectional LSTM and GRU layers. This strategic amalgamation taps into the inherent strengths of each component, facilitating nuanced sequential data processing and adept temporal dynamics capture. The empirical analysis, juxtaposed against a spectrum of both traditional and state-of-the-art methodologies, attests to the unparallelled precision, adaptability, and resilience of DeepTimeNet. Whether evaluated against diverse driving profiles like US06 and Federal Urban Driving Schedule (FUDS) or subjected to a wide array of ambient temperatures, ranging from extreme colds of −10 °C to the warmth of 50 °C, DeepTimeNet consistently emerges as a paragon of excellence. Its performance metrics, particularly RMSE, MSE, MAE, and MAXE, not only resonate with statistical significance but also redefine contemporary benchmarks in the battery ST estimation arena. In varying temperature scenarios, from the chill of −10 °C to the warmth of 50 °C, DeepTimeNet consistently showcases its superiority. This is particularly evident when benchmarked against alternate methodologies. Its adept handling of both spatial and temporal patterns inherent within time-series data, combined with its robust architecture, ensures precise and reliable battery surface temperature estimations. In essence, DeepTimeNet signals a paradigmatic shift in battery ST estimation. The model, with its intricate blend of GRUs, RNNs, CNNs, and other cutting-edge components, heralds significant advancements for battery management systems. Rooted in its empirical successes, DeepTimeNet stands poised to drive improvements in operational efficiency, battery longevity, and safety across a spectrum of real-world applications.

CRediT authorship contribution statement

Muhammad Hamza Zafar: Investigation, Data curation, Conceptualization. **Syed Muhammad Salman Bukhari:** Software, Resources, Formal analysis. **Mohamad Abou Houran:** Data curation, Formal analysis, Validation, Visualization. **Majad Mansoor:** Resources, Investigation, Formal analysis. **Noman Mujeeb Khan:** Software, Data curation. **Filippo Sanfilippo:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

None.

All authors claim that there is not any conflict of interest regarding the above submission. The work of this submission has not been published previously. It is not under consideration for publication elsewhere. Its publication is approved by all authors and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder.

Data availability

Data used in this study is openly available.

Acknowledgement

This research is supported by the Artificial Intelligence, Biomechatronics, and Collaborative Robotics research group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

References

[1] A.A. Hussein, A.A. Chehade, Robust artificial neural network-based models for accurate surface temperature estimation of batteries, IEEE Trans. Ind. Appl. 56 (5) (2020) 5269–5278.

[2] C.G. Moral, D. Fernandez, J.M. Guerrero, D. Reigosa, C.R. Pereda, F. Briz, Thermal monitoring of LiFePO 4 batteries using switching harmonics, IEEE Trans. Ind. Appl. 56 (4) (2020) 4134–4145.

[3] M. Lelie, T. Braun, M. Knips, H. Nordmann, F. Ringbeck, H. Zappen, D.U. Sauer, Battery management system hardware concepts: An overview, Appl. Sci. 8 (4) (2018) 534.

[4] R. Schwarz, K. Semmler, M. Wenger, V.R. Lorentz, M. März, Sensorless battery cell temperature estimation circuit for enhanced safety in battery systems, in: IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2015, pp. 001536–001541.

[5] R.R. Richardson, D.A. Howey, Sensorless battery internal temperature estimation using a Kalman filter with impedance measurement, IEEE Trans. Sustain. Energy 6 (4) (2015) 1190–1199.

[6] R. Xiong, X. Li, H. Li, B. Zhu, A. Avelin, Neural network and physical enable one sensor to estimate the temperature for all cells in the battery pack, J. Energy Storage 80 (2024) 110387.

[7] J. Shen, Z. Zhang, S. Shen, Y. Zhang, Z. Chen, Y. Liu, Accurate state of temperature estimation for lithium-ion batteries based on square root cubature Kalman filter, Appl. Therm. Eng. 242 (2024) 122452.

[8] O. Demirci, S. Taskin, E. Schaltz, B.A. Demirci, Review of battery state estimation methods for electric vehicles-part I: SOC estimation, J. Energy Storage 87 (2024) 111435.

[9] K. Das, R. Kumar, A. Krishna, Analyzing electric vehicle battery health performance using supervised machine learning, Renew. Sustain. Energy Rev. 189 (2024) 113967.

[10] S. Oyucu, F. Doğan, A. Aksöz, E. Biçer, Comparative analysis of commonly used machine learning approaches for Li-ion battery performance prediction and management in electric vehicles, Appl. Sci. 14 (6) (2024) 2306.

[11] X. Hu, S.E. Li, Y. Yang, Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles, IEEE Trans. Transp. Electr. 2 (2) (2015) 140–149.

[12] H. Zuo, B. Zhang, Z. Huang, K. Wei, H. Zhu, J. Tan, Effect analysis on SOC values of the power lithium manganate battery during discharging process and its intelligent estimation, Energy 238 (2022) 121854.

[13] E. Chemali, P.J. Kollmeyer, M. Preindl, R. Ahmed, A. Emadi, Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries, IEEE Trans. Ind. Electron. 65 (8) (2017) 6730–6739.

[14] F. Yang, S. Zhang, W. Li, Q. Miao, State-of-charge estimation of lithium-ion batteries using LSTM and UKF, Energy 201 (2020) 117664.

[15] V.M. Nagulapati, H. Lee, D. Jung, S.S. Paramanantham, B. Brigljevic, Y. Choi, H. Lim, A novel combined multi-battery dataset based approach for enhanced prediction accuracy of data driven prognostic models in capacity estimation of lithium ion batteries, Energy AI 5 (2021) 100089.

[16] M. Naguib, P. Kollmeyer, C. Vidal, A. Emadi, Accurate surface temperature estimation of lithium-ion batteries using feedforward and recurrent artificial neural networks, in: 2021 IEEE Transportation Electrification Conference & Expo, ITEC, IEEE, 2021, pp. 52–57.

[17] F. Yang, W. Li, C. Li, Q. Miao, State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network, Energy 175 (2019) 66–75.

[18] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.

[19] D. Bernardi, E. Pawlikowski, J. Newman, A general energy balance for battery systems, J. Electrochem. Soc. 132 (1) (1985) 5.

[20] G. Dos Reis, C. Strange, M. Yadav, S. Li, Lithium-ion battery data and where to find it, Energy AI 5 (2021) 100081.

[21] W. He, N. Williard, C. Chen, M. Pecht, State of charge estimation for li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation, Int. J. Electr. Power Energy Syst. 62 (2014) 783–791.

[22] C. Bian, H. He, S. Yang, Stacked bidirectional long short-term memory networks for state-of-charge estimation of lithium-ion batteries, Energy 191 (2020) 116538.

[23] Y. Tian, R. Lai, X. Li, L. Xiang, J. Tian, A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter, Appl. Energy 265 (2020) 114789.

[24] M. Cai, M. Pipattanasomporn, S. Rahman, Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques, Appl. Energy 236 (2019) 1078–1088.

[25] S. Targ, D. Almeida, K. Lyman, Resnet in resnet: Generalizing residual architectures, 2016, arXiv preprint arXiv:1603.08029.

[26] F. Huang, J. Ash, J. Langford, R. Schapire, Learning deep resnet blocks sequentially using boosting theory, in: International Conference on Machine Learning, PMLR, 2018, pp. 2058–2067.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, No. 1, 2017.

[28] R. Lippmann, Book review: ‘‘neural networks, a comprehensive foundation’’, by simon haykin, Int. J. Neural Syst. 5 (04) (1994) 363–364.

[29] B. Zhang, D. Xiong, J. Su, H. Duan, A context-aware recurrent encoder for neural machine translation, IEEE/ACM Trans. Audio Speech Lang. Process. 25 (12) (2017) 2424–2432.

[30] A. Adhikari, A. Ram, R. Tang, J. Lin, Rethinking complex neural network architectures for document classification, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4046–4051.

[31] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.

[32] S. Nosouhian, F. Nosouhian, A.K. Khoshouei, A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU, 2021.

[33] G. Cho, M. Wang, Y. Kim, J. Kwon, W. Su, A physics-informed machine learning approach for estimating lithium-ion battery temperature, IEEE Access 10 (2022) 88117–88126.

[34] G. Cho, D. Zhu, J.J. Campbell, M. Wang, An LSTM-PINN hybrid method to estimate lithium-ion battery pack temperature, IEEE Access 10 (2022) 100594–100604.

[35] Q. Yao, D.D.-C. Lu, G. Lei, A surface temperature estimation method for lithium-ion battery using enhanced GRU-RNN, IEEE Trans. Transp. Electr. 9 (1) (2022) 1103–1112.

[36] Y. Zheng, Y. Che, X. Hu, X. Sui, R. Teodorescu, Sensorless temperature monitoring of lithium-ion batteries by integrating physics with machine learning, IEEE Trans. Transp. Electr. (2023).