



A deep learning multi-feature based fusion model for predicting the state of health of lithium-ion batteries

Ankit Sonthalia^a, J.S. Femilda Josephin^b, Edwin Geo Varuvel^{c,*},
Arunachalam Chinnathambi^d, Thiagarajan Subramanian^e, Farzad Kiani^f

^a Department of Automobile Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Delhi – NCR Campus, Modinagar, India

^b Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Istinye University, Istanbul, Turkiye

^c Department of Mechanical Engineering, Faculty of Engineering and Natural Sciences, Istinye University, Istanbul, Turkiye

^d Department of Botany and Microbiology, College of Science, King Saud University, PO Box -2455, Riyadh, 11451, Saudi Arabia

^e Department of Automobile Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, 602105, India

^f Data Science Application and Research Center (VEBIM), Fatih Sultan Mehmet Vakif University, Istanbul, Turkiye

ARTICLE INFO

Keywords:

Deep learning
State of health
Lithium-ion battery
Machine learning
Prediction

ABSTRACT

Lithium-ion batteries have become the preferred energy storage method with applications ranging from consumer electronics to electric vehicles. Utilization of the battery will eventually lead to degradation and capacity fade. Accurately predicting the state of health (SOH) of the cells holds significant importance in terms of reliability and safety of the cell during its operation. The battery degradation mechanism is strongly non-linear and the physics-based model have their inherent disadvantages. The machine learning method has become popular for estimating SOH due to its superior non-linear mapping, adaptive, and self-learning capabilities, made possible by advances in deep learning technologies. In this study parallel hybrid neural network is formulated for predicting the state of health of lithium-ion cell. Firstly, the factors that have an effect on the cell state were analysed. These factors are cell voltage, charging & discharging time and incremental capacity curve. The features were then processed for use as input to the model. Spearman correlation coefficient analysis shows that all the factors had a positive correlation with SOH. While charging time has a negative correlation with the other features. Next the deep learning models namely convolution neural network (CNN), temporal convolution network (TCN), long-short-term memory (LSTM) and bi-directional LSTM were used to make fusion models. The number of layers in CNN and TCN were also varied. The hyperparameters used in the models were optimized using Bayesian optimization algorithm. The models were validated through comparative experiments on the University of Maryland battery degradation dataset. The prediction accuracy with CNN 3-layer LSTM was found to be the best for the training and the test dataset. The overall R^2 value, root mean squared error (RMSE) and mean absolute percentage error (MAPE) with the model was found to be 0.999646, 0.003807 and 0.3, respectively. The impact of the features on the model was also analysed by removing one feature and retraining the model with the other features. The effect of discharging time and the peak of the discharge incremental capacity curve was maximum. The analysis also reveals that either charging voltage or discharging voltage can be used. Further, the proposed model was also compared with the other studies. The comparison shows that the R^2 , RMSE and MAPE values of the proposed model was better.

1. Introduction

The incessant use of fossil fuels has resulted in the degradation of the atmospheric air causing ill effects on the human and other living beings. They are also the cause of global warming which has resulted in increase

in the Earth's temperature, along with changes in the weather patterns thus affecting the crops. Using renewable sources of energy can curb the use of fossil fuels. Solar and wind energy are two such renewable sources that can effectively reduce the emissions [1]. Countries around the world are promoting the use of clean and renewable energy. However, both solar and wind energy are intermittent and are available for a

This article is part of a special issue entitled: SpliTech2024 published in Energy.

* Corresponding author.

E-mail address: vedwingeo@gmail.com (E.G. Varuvel).

<https://doi.org/10.1016/j.energy.2025.134569>

Received 21 October 2024; Received in revised form 13 January 2025; Accepted 15 January 2025

Available online 17 January 2025

0360-5442/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Abbreviation and nomenclature

ADAM	Adaptive Moment Estimation
API	Application Programming Interface
Bi-LSTM	Bi-directional Long-Short-Term Memory
Bi-LSTM-NN	Back propagation Long-Short-Term Memory Neural Network
BPTT	Backpropagation-through-time
CNN	Convolution Neural Network
EKF	Extended Kalman Filter
EIS	Electrochemical Impedance Spectroscopy
FM	Feature Maps
GPR	Gaussian Process Regression
IC	Incremental Capacity
LSM	Long-Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLR	Multiple Linear Regression
R ²	Goodness of Fit
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
RVM	Relevance Vector Machine
SEI	Solid Electrolyte Interphase
SOF	State of Function
SOH	State of Health
SVM	Support Vector Machine
TCN	Temporal Convolution Neural Network
TF	Tensor Flow
TFRF	Tensor Flow Record Format
$b^{(l+1)}$	bias in l+1st layer
B_s	Batch size
$\tilde{c}_t$	candidate memory cell
cht_i^k	scalar quantity denoting charging time of kth cell
c_{t-1}	memory unit last iteration state
$Ch_v_i^k$	vector denoting charge voltage of kth cell
d	Dilation Factor
dht_i^k	scalar quantity denoting discharging time of kth cell
D	size of input data
DNN	Deep Neural Network
DTC	Differential Thermal Capacity
$Dh_v_i^k$	vector denoting discharge voltage of kth cell
D_L	Dropout rate LSTM/BiLSTM
D_r	Dropout rate of CNN or TCN
$f(\cdot)$	activation function

$\vec{h}_t$	forward hidden layer
$\overleftarrow{h}_t$	backward hidden layer
h_t	output at the present moment
h_{t-1}	output of previous moment 't-1'
I	Cell Current
k kth	convolutional kernel
K	number of convolutional kernels in a layer
K_s	Kernel size of CNN or TCN
L_r	Learning rate
N_f	Number of filters in TCN or CNN
N_s	Number of strides in CNN
$o(m, n)_k$	Eigen value at position m, n
OIR	Ohmic Internal Resistance
$pdq_dv_i^k$	scalar quantity denoting peak of IC curve of kth cell
P	Padding
Q	Cell capacity
$Q_{current}$	current available capacity
$Q_{initial}$	initial maximum available capacity
$R_{i,j}$	pooling region
S	Stride
t	Sampling Time
TL-CNN	Transfer Learning CNN
TL-DNN	Transfer Learning DNN
U_D	Number of units in Dense layer
U_L	Number of units in LSTM/BiLSTM layer
$vdq_dv_i^k$	scalar quantity denoting voltage at which peak occurs of kth cell
V	Cell Voltage
x_c	combine input data
x_t	input data at current moment t
y_i	true value
$\hat{y}_i$	predicted value
$\bar{y}$	mean value
y_t	output sequence
$y(i, j)_k$	pooling operator output at position i, j
z	Standard Score
Z_l^t	neuron output at position in t in kth layer
$Z_l(i, j)$	input of lth layer
$Z^{(l+1)}(i, j)$	output of l+1st layer
μ	mean of the distribution
σ	standard deviation of the distribution
ω_k^k	weight at position k
$\omega_k^{(l+1)}$	neuron weight at the l+1st layer
ω_{hi}	weight of the output of previous moment
ω_{xi}	weight of the input at current moment

particular time of the day. The energy produced by such sources can be stored and utilized whenever required. Lithium-ion batteries are one such solution that stores the energy and gives it away easily. It is also seen that the vehicles running on fossil fuel are a major cause of pollution in the cities. By switching to lithium-ion battery operated vehicle, the pollution can be drastically reduced. The decreasing cost of lithium-ion cells and battery packs along with subsidies on electric vehicles given by the governments has drastically reduced the cost of owning an electric vehicle. The reduction in cell and pack cost over the last decade is depicted in Fig. 1. The cost of a pack in 2013 for US\$780 which has reduced to US\$139. The major reason for the reduction was decrease in the cost of raw materials and components as the production capacity worldwide increased [2]. Due to the reduction in the costs, the sales of electric cars have exponentially increased. Fig. 2 shows country wise the number of electric vehicles sold between 2010 and 2023.

During usage the batteries are continuously charged and discharged resulting in gradual decline in their performance [4]. When their performance deteriorates to a failure threshold, they reach their end of life. The threshold is generally 70–80 % of their rated capacity or 200 % of their initial internal resistance. Continuous usage of failed cells can be a security risk to the system and may even cause a catastrophe in extreme cases [5]. The accurate assessment of the state of the lithium-ion cell is therefore crucial for ensuring safety and efficiency during its usage. This way the maintenance of the system can be timely scheduled thereby maximizing its operational availability [6]. Some of the indicators used for evaluating the health status of a battery are state of function (SOF), state of health (SOH), remaining useful life (RUL) and so on. However, obtaining the values of the indicators is a difficult task [7]. Since, the cell degradation is non-linear; during operations there are intense uncertainties; and while monitoring the data, noise and outliers create a

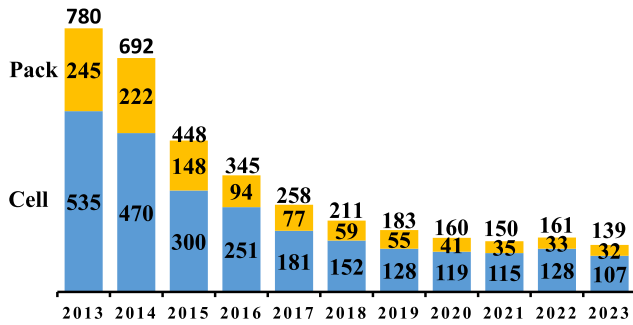


Fig. 1. Volume-weighted average lithium-ion cell and battery pack price [2]. 2024E is the estimated sales based on sales through first quarter of 2024.

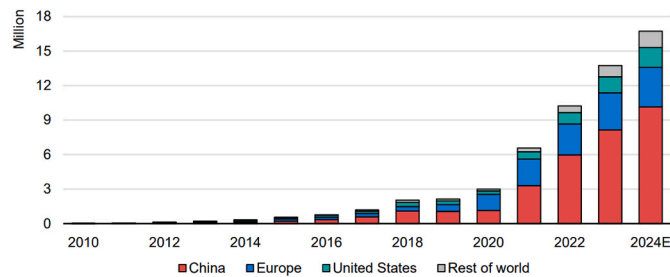


Fig. 2. Electric car sales between 2010 and 2024 [3].

challenge. Also, predicting the life of the cell in early stages is a difficult task [8].

The cell state of health is directly characterized by its capacity. As the degradation of the cell occurs their capacity decreases [9]. Many studies have been carried out to estimate the capacity of the cell. The methods used for estimation are either model-based or data-driven [10]. When the electro-chemical reaction mechanism of the cell is considered for estimation of the indicators, then the built model is known as model-based. Some of the commonly used models are electrical model, electrochemical model, empirical model, fractional-order model, fusion model and so on. Combination of discrete Arrhenius ageing model and sequential extended Kalman Filter (EKF) were used by Zheng et al. for estimating the battery capacity [11]. To estimate battery available capacity, five distinctive factors that are significantly linked to the battery aging mechanism were found and extracted from the electrochemical model [12]. Yu et al. [13] employed the RLS technique for model parameter identification and the adaptive H_∞ filter algorithm to predict battery capacity based on the Thevenin equivalent circuit model. Li et al. [14] coupled the chemical and mechanical degradation mechanisms to create a single particle-based degradation model, from which the capacity loss and voltage fluctuation are thereafter reliably predicted. Zhang et al. [15] developed a connected electro-thermal model to predict battery capacity and temperature in a hierarchical manner. Wang et al. [16] presents a fractional-order model that has been optimized for studying lithium-ion batteries and has produced precise computation results. He et al. [17] presented an analogous circuit model with several time scales and variable orders. Another popular model that can balance the complexity and accuracy of the electrochemical and equivalent circuit models is the fractional-order model. Xu et al. [18] suggested a fusion model that integrated the analogous circuit model and the electrochemical model. Wang et al. [19] created a thermoelectric coupling model that included an analogous circuit model and the thermal effect of the battery. Shao et al. [20] suggested a brand-new electrochemical model to estimate the capacity. While these models can provide lucid physical explanations, their broad applicability is somewhat hampered by the large number of parameters they often require to calculate, and

Table 1

Summary of studies on state of health prediction.

Reference	Dataset	Feature	Methodology	Result
Luo et al. [29]	SamSung ICR18650–26J cells (2.6 Ah)	EIS	SVM	RMSE≤0.001 % MAE≤2 %
Liu et al. [36]	3.2Ah NCR18650B, NASA (B05, B06, B07)	Constant voltage charging phase: charging time, charging capacity, average current, time interval with an equal charging current difference, current interval with an equal charging time difference	GR-SVM, LSTM	RMSE≤0.47 % MAPE≤0.57 %
Lin et al. [44]	Oxford dataset	DTC	SVR, GS-SVR, SA-SVR, GRU	MAE≤1 %
Hong et al. [45]	95 lithium iron phosphate batteries, total capacity 120Ah	OIR		MAE≤4 %
Wang et al. [46]	NASA dataset	Time interval during equal voltage increase and decrease	EMD, RF, GRU	MAE≤1.84 % RMSE≤2.5 % MAPE≤2.76 %
Chen et al. [38]	NCA dataset (3.5Ah), NCM dataset (3.5Ah), NCA + NCM dataset (2.5Ah)	partial constant voltage charging data	CNN, TL-CNN	RMSE≤0.85 % (NCA dataset) RMSE≤2.25 % (NCA + NCM dataset)
Li et al. [39]	12 Eunicell LR2032 lithium-ion coin cells	EIS	DNN, TL-DNN	MAPE≤0.0036 %
Cheng et al. [47]	CALCE	Charging current, voltage and capacity	B-LSTM NN	RMSE≤0.125 %
Obregon et al. [48]	12 Eunicell LR2032 lithium-ion coin cells	EIS	CAE, DNN	MAPE≤0.0128 % RMSE≤0.0129 %

the accuracy of these calculations is heavily dependent on the quality of the model [21].

The models can be coupled with advanced filtering techniques such as particle filter [22], Kalman filter [23] and their deviations namely unscented Kalman filter [24] and unscented particle filter [25] for precise prediction of cell status. Although model-based solutions have shown significant improvement, there are still certain drawbacks. These models require a lot of computing and prior knowledge which is known for some specific cells only. Due to the intricate chemical interactions occurring inside of cell, it is challenging to create a model that can precisely depict aging features of batteries at every stage of their existence. Lastly, the models are not universal and cannot be used for wide range of applications [26].

These model based methods have high accuracy for estimating the parameters. However, prior knowledge is required for estimation. Also, the models are not universal and are specific to a particular cell chemistry. The data-driven models on the other hand are not required to

master the kinetics of the lithium-ion cell, instead the state of health can be estimated only through data. In real case scenarios only cell current, voltage and in some cases temperature and internal resistance, etc. may be known. Some of the methods that can be used for accurate estimation of state of health are regression methods, deep-learning methods, neural network methods and graphs-based methods etc. Table 1 shows the summary of the work carried out by the researchers. A generic approach for estimating SOH was proposed by Shu et al. [27] utilizing a least squares support vector machine (SVM). Kullback-Leibler distances were determined from the charging voltage profile [28]. The distance was used as a feature for training a radial basis function network which can be used for understanding the dynamic characteristics of cell ageing. Luo et al. [29] used electrode impedance spectroscopy data to predict the state of health of the batteries using SVM model. The authors achieved an accuracy of 80 % with good fitting of life-time tracking. However, the SOH output was affected by environmental noise and load conditions. An improved firefly algorithm – Gaussian process regression model was used for selecting and adjusting the hyper-parameters [30]. The prediction accuracy of the model was improved by 91.64 %. The machine learning approaches discussed here are based on single-source characteristics. Hence, they have limits in estimating SOH accuracy due to local optimization. Moreover, there are issues with using regression models like SVM, GPR and multiple linear regression (MLR). A generic supervised learning technique that benefits from probabilistic issues is called Gaussian process regression [31]. When the feature space is high dimensional, it is more likely to fail. Support vector regression is a technique for achieving a linearization process by lifting the dimensionality of complex nonlinear problems [32]. Nevertheless, it is susceptible to kernel functions, parameters, and missing data. Multiple linear regression has the ability to tackle complex nonlinear problems. However, it is limited in its ability to swiftly calculate the degree of linear connection among the components [33].

To address the above-mentioned issues with single source features, studies with multiple features was carried out. Wang et al. [34] introduced a differential thermal voltammetry feature that takes into account temperature and voltage characteristics. The features were then used to evaluate the battery SOH using the GPR model. Dai et al. [35] extracted features from current, voltage, temperature, incremental capacity and differential thermal voltammetry curves. The features were used to train a dual-kernel Gaussian process regression model. The hyperparameters were optimized with northern goshawk algorithm. The RMSE value with NASA dataset was 0.4 % and with Oxford dataset it was 0.64 %. Liu et al. [36] used support vector regression and LSTM were used for predicting the SOH of a battery ageing dataset and NASA dataset. The authors found an RMSE and MAPE value of 0.22 and 0.24 %, respectively with NASA dataset, while the values were 0.47 and 0.57 %, respectively with their laboratory test dataset.

Apart from the studies described previously, neural network-based algorithms have shown great accuracies in predicting the battery health. Convolutional neural networks were used for estimating the capacity of a lithium-ion cell [37]. The authors achieved high accuracy in estimating the cell capacity by optimizing the hyperparameters. Partial constant voltage charging data was used as input for a CNN model and transfer learning based CNN model [38]. The authors found that transfer learning lead CNN model resulted in lower RMSE and MAE. Li et al. [39] trained a transfer learning based-deep neural network using electrode impedance spectroscopy data of lithium-ion batteries. The dataset was collected at 25, 35 and 45 °C. The authors observed that the data trained at 25 °C could be used for transfer learning for data at 35 and 45 °C. The transfer learning could be achieved with only 6 % of the data. The MAPE was reduced by 48 % and 55 % with transfer learning for the dataset at 35 and 45 °C.

Recently, convolutional neural network with LSTM has been gaining popularity for state of health determination and lifetime prediction. An auto-CNN-LSTM was used by Ren et al. [40] for predicting the remaining life of the cell. Stacked uni- and bi-directional LSTM along with CNN was

used by An et al. [41] in predicting the milling life of a tool. For predicting the life of rotating machineries, Ma et al. [42] also utilized CNN based LSTM. Chen et al. [43] predicted the lifetime of lithium-ion cells using a two-dimensional and one-dimensional parallel hybrid convolutional neural network along with long short-term memory network. Li et al. [4] compared the temporal convolutional network, convolutional neural network along with long short-term memory network for predicting the state of health and life-cycle of the cell.

1.1. Novelty and objectives

Studies have revealed that hybrid models of LSTM and CNN can achieve better results than single models. Moreover, it is seen that CNN is unable to capture the time-dependent information due to fixed convolutional kernel size resulting in errors in estimation of remaining useful life and state of health [49]. Residual block structure and causal dilated convolution was added to CNN, known as TCN, to make it suitable for processing time series [50]. Temporal convolutional network (TCN) characterises convolutional calculations which allows local temporal feature extraction with high accuracy. Thus, it can extract local temporal features from the ageing data [51]. While LSTM can extract global temporal features from the TCN results. In this study, a novel hybrid model based on TCN and uni- and bi-directional LSTM is proposed for estimating the state of health of lithium-ion cell. Other models namely CNN with uni- and bi-directional LSTM were also adopted for the comparison. To optimize the hyperparameters for all the models, Bayesian optimization algorithm was utilized. The dataset provided by University of Maryland was used for verifying the performance of the models in state of health estimation. The novelty of the study lies in the fact that a parallel hybrid network was developed for the prognosis of the battery state of health. Most of the studies have stacked single data in the CNN/TCN network along with LSTM. Also, the features used in this study are derived from current and voltage during the cycling of the cell. Hence, complex battery testing equipment is not required for SOH estimation. This method allows accurate prediction by the model due to comprehensive feature extraction from the data. The major contributions from this study are:

- Pre-processing the dataset for extracting features that are suitable for predicting the state of health. The dataset was derived only from voltage and current. The dataset is divided into scalar and vector data, where the vector data is time-based.
- The parallel network architecture of CNN/TCN extracts features from the scalar and vector data which is then fed to the LSTM network.
- Optimizing hyperparameters, plays a major role in the prediction accuracy of the model. In this study all the models were optimized using Bayesian optimization algorithm.
- Experiments were performed using the University of Maryland dataset of prismatic type lithium-cobalt (LiCoO₂) cells.

The remaining of the study is organized as follows. In section 2, the dataset used was described followed by the features used in the study and their preprocessing. Section 3 discusses the various models used for the experiments. In section 4, the methodology used for the experimentation and the evaluation criteria is discussed. The results are analysed in section 5. The performance of the proposed model is compared with the benchmark models in section 6. Advantages and limitations of the study are discussed in section 7. The final conclusions are drawn in section 8.

2. Experimental data

2.1. Degradation dataset

The ageing data of seven prismatic cells with a nominal capacity of

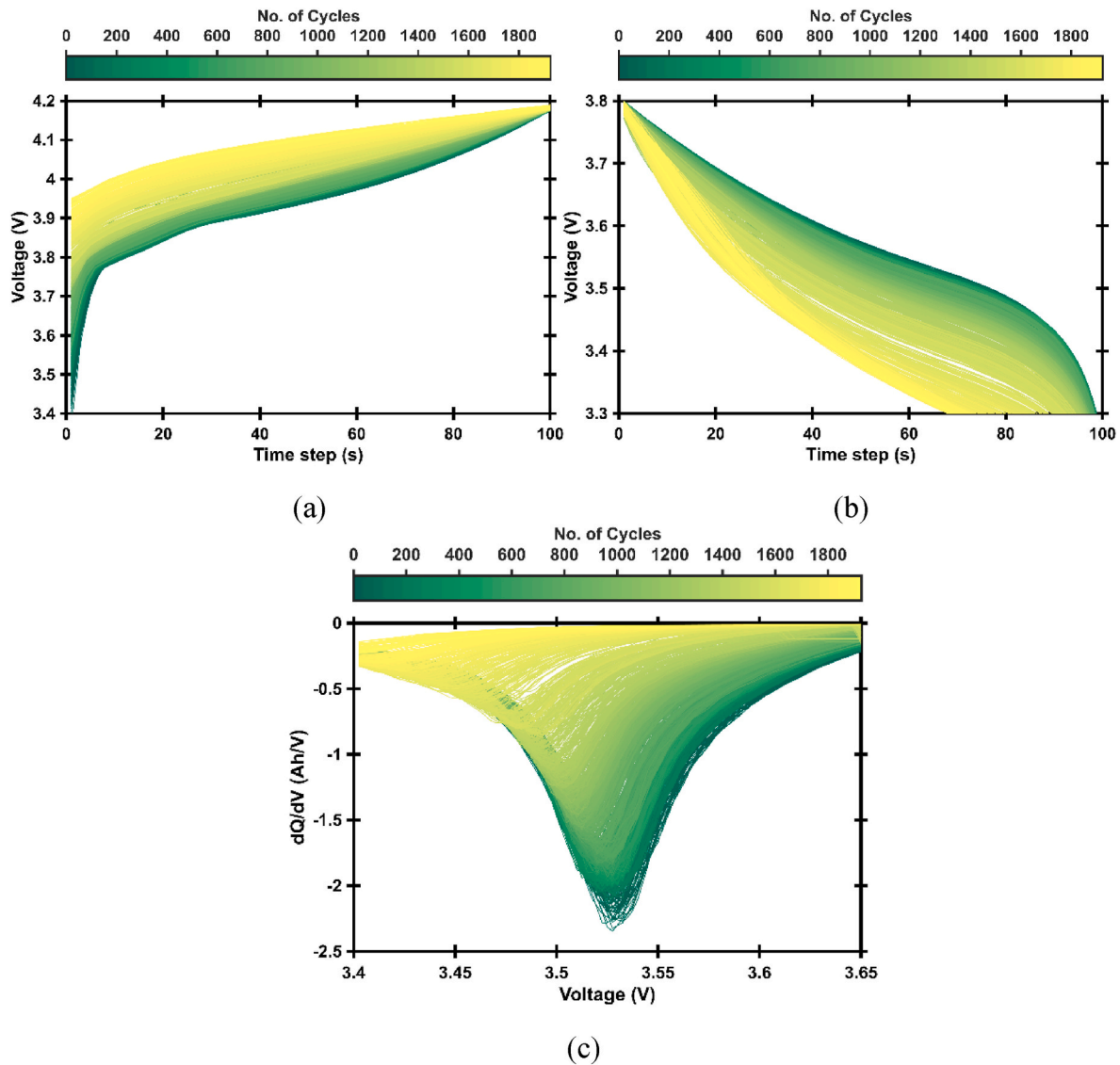


Fig. 3. (a) charge voltage, (b) discharge voltage, and (c) ICA curve during discharging.

1350mAh, noted as cell 1 to 7 were used in this study. The dataset was provided by Centre for Advanced Life Cycle Engineering, University of Maryland [52]. The cell had a graphite negative electrode, while lithium cobalt oxide (LiCoO_2) was used as a positive electrode material. The dimension of the cell was $6.6 \times 33.8 \times 50$ mm. During the ageing tests, the cells were charged with a constant current/constant voltage charging protocol with a constant charging rate of 0.5C. The constant current charging was carried out till the voltage reached 4.2V, then the protocol was shifted to constant voltage charging which was continued till the current dropped to 0.05A. The discharging of the cells was also carried out at a constant rate of 0.5C, under constant current discharge condition. All the parameters of the cell were recorded using Arbin Battery Tester.

2.2. Voltage feature

The lithium-ion cell is charged in two-stages; constant-current charging and constant voltage charging. The cell was cycled (charge/discharge) till it reached its end of life. Fig. 3a shows the terminal voltage curves during charging for cell number 3. It is seen that the slope is less in the initial cycles, however, when the cell is cycled for long, the voltage curve has a greater slope. Thus, the terminal voltage of the cell

during charging can be selected as a feature for estimating the cell state of health. Similarly, Fig. 3b shows the cell terminal voltage during discharging. The slope of the curve during initial cycles is high, which decreases with increase in cycling of the cell. Therefore, the cell terminal voltage during discharging was also selected as a feature.

The entire voltage range cannot be used for the prediction. Since, during constant voltage charging, the changes in cycles are miniscule. Similarly, during constant current charging, the changes in voltage below 3.4V were insignificant. Hence, a voltage range of 3.4–4.2V, shown in Fig. 3a, was selected for training the model. Likewise, a voltage range of 3.3–3.8V, shown in Fig. 3b, was selected for training the model. The data segment selected had hundreds or in some cases thousands of points which increases the computational burden during the training. Hence, the input length of the feature vectors was taken as 100. The vector was generated by linearly interpolating the data with defined sampling interval.

The time required for charging and discharging the cell was also considered for training the model. Since, the time required decreases with increase in number of cycles.

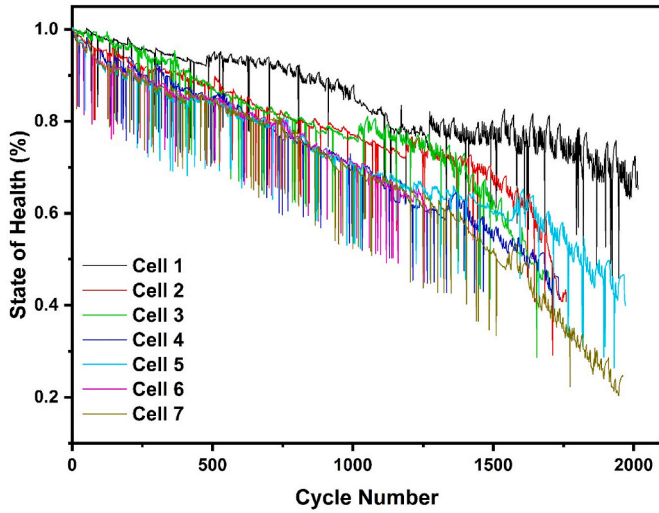


Fig. 4. State of health of the cells.

2.3. Capacity increment feature

The incremental capacity (IC) feature is widely used for describing the battery ageing process. The analysis by the feature is carried out at the electrode level. The features of the battery's phase change during the insertion and delamination of the lithium-ion active material can be explained by incremental capacity curve. The significant advantage of the feature is that the peak of the curve is formed due to the plateauing of the cell voltage [53]. During constant current charging/discharging mode, the IC is calculated using eq. (1).

$$IC = \frac{dQ}{dV} = \frac{I \cdot dt}{dV} \quad (1)$$

where, 'Q' is the capacity of the cell, 'V' is the cell voltage, 't' is sampling time and 'I' is the cell current.

The IC curve with constant current discharging of the cell is illustrated in Fig. 3c. With increase in cycles, the curve becomes flat, i.e., the peak gradually decreases. The distinctive shape, height, and placement of the peak represents the electrochemical reactions that occur when batteries are discharged. The loss of active material is the cause of decrease in the peak. As the cell ages, the active material is deposited on the electrolyte resulting in the formation of solid electrolyte interphase (SEI) layer. Thus, the active material can no longer participate in the charging/discharging process, thereby decreasing the peak of the IC. Hence, peak of the IC curve is an important feature for describing the capacity of the cell. Similarly, the voltage at which the peak occurs reduces with increase in cycles. This feature also has an important bearing on the capacity of the cell.

2.4. State of health

The state of health of the cell is defined as the ratio of available capacity at an instance to the initial maximum capacity. Equation (2) is used to calculate the state of health of a cell. A decrease in battery capacity and an increase in internal resistance can result from lithium-ion battery depletion. It is commonly accepted that a battery has reached the end of its useful life when its current available capacity falls to 80 % of its original capacity. Fig. 4 shows the state of health of the cells used in this study.

$$SOH = \frac{Q_{current}}{Q_{initial}} \quad (2)$$

where 'Q_{current}' and 'Q_{initial}' are the current and initial maximum available capacity.

2.5. Data preprocessing

The data gathered during cycling of the cell may be noisy which will result in improper training of the model, thereby decreasing its prediction accuracy. It is therefore necessary to process the data before feeding it to the model. In this study, the data was first unified, then normalized, partitioned and finally stored in a format that can be easily fed to the model.

The lengths of the timestamp-based data can vary depending on the signals that are collected several times in each cycle. Thus, it is necessary to unify the data benchmark. As discussed previously, the charge voltage and discharge voltage are turned in matrices of size 100×1 . The scalars, namely, charge time, discharge time, peak of the discharge IC curve, voltage at which the peak of the IC curves, were measured only once per cycle. Also, the state of health of the cell was recorded only once per cycle.

Thus, for the single cycle of each cell, the following expression can be derived:

$$DM_i^k = [Ch.v_i^k, Dh.v_i^k, cht_i^k, dht_i^k, pdq.dv_i^k, vdq.dv_i^k, SOH_i^k],$$

$$k \in \{1, 2, 3, \dots, 7\}$$

$$(3)$$

Where $Ch.v_i^k$, and $Dh.v_i^k$ are the vectors denoting the charge and discharge voltage, respectively. cht_i^k , dht_i^k , $pdq.dv_i^k$, $vdq.dv_i^k$, and SOH_i^k are the scalar quantities denoting charging time, discharging time, peak of the IC curve, voltage at which the peak occurs and the state of health of the cell, respectively.

If the model is fed directly with the numerical data, it may lead to large gradients which may slow down the learning ultimately resulting in divergence. Moreover, the ability to generalize about other data is diminished when the model learns too much about the distribution properties of non-normalized data. Thus, the input data was converted into dimensionless data so that the influence of the dimensions and the magnitude of the data can be neutralized. This way the relative relationship while generalizing the statistical distribution of the data can be preserved and still make them comparable. Hence, mean - standard deviation algorithm was used for standardizing the input data using equation (4).

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

where 'z' is the standard score, 'μ' is the mean of the given distribution and 'σ' is the standard deviation of the given distribution.

Next the existing dataset was divided into training and testing dataset. The training set was used to training the model by calculating the gradients and updating biases and weights used in the model. While the test set can be used to assess how well the model generalizes; it is mutually exclusive with the training set. In the present study, the cells with cell-id, Cell1, Cell3, Cell4, Cell6 and Cell7 were used for training the models while Cell2 and Cell5 were used for testing the model.

Lastly, the processed data must be stored on the hard disk in a file format which can be read by the model at high speeds. Thus, the data of each cell was stored as a linearly readable file in tensor flow record format (TFRF). In order to make more effective use of memory, the TFRF represents a '{string: value}' mapping and is capable of serializing structured data effectively. A noteworthy advantage of this format above others is that large amount of data can be parallelly fed to the model.

3. Methodology

In this study, the models were developed with CNN, TCN and LSTM, the relevant theories behind these algorithms is described in the upcoming sub-sections.

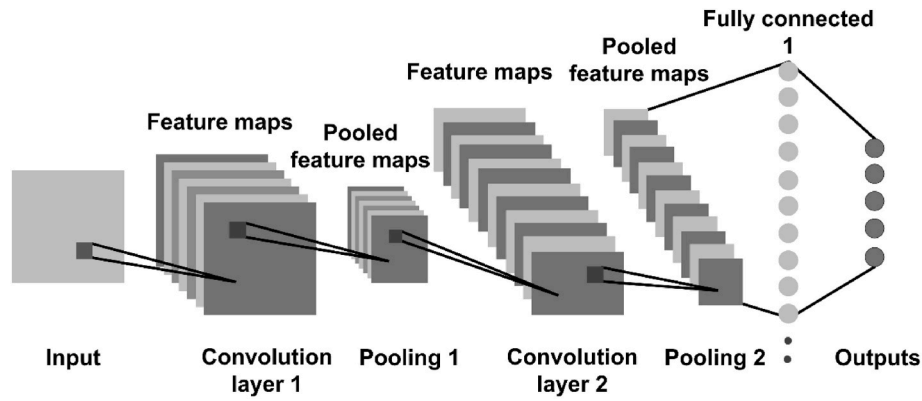


Fig. 5. A general convolution network hierarchy.

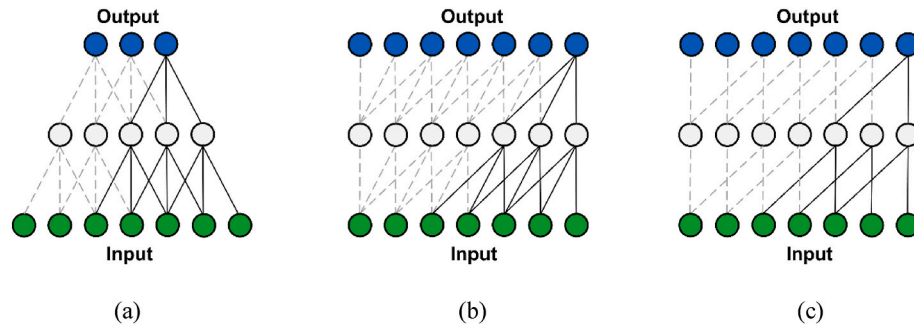


Fig. 6. (a) Standard convolution block with '3' kernel size; (b) causal convolution block with '3' kernel size and (c) dilated causal convolution '2' kernel size and '2' dilation rate [58].

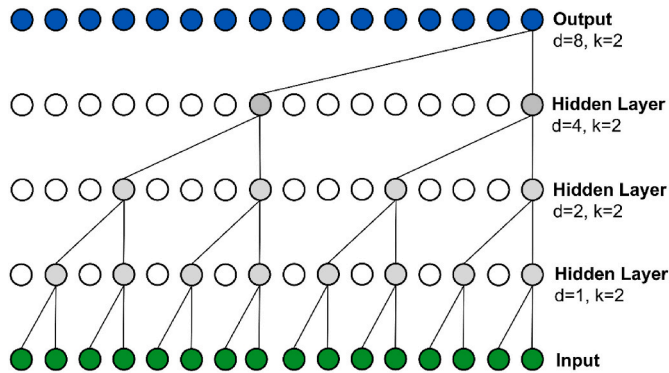


Fig. 7. Extended convolution of TCN network [4].

3.1. Convolutional neural network

It is a feed-forward deep neural network containing one or more convolutional layers. It is a classical deep-learning algorithm with applications in image recognition, computer vision and natural language processing, etc. [54]. In general the network consists of one input layer, multiple convolutional layers, sub-sampling layers, fully connected layers and an output layer.

3.1.1. Convolution layer

The core of the neural network is the convolutional layer that extracts features from the input layer or lower-level feature maps (FM) (Fig. 5). Convolutional kernels are present in the convolutional layer. Every element of the kernel has a bias and a weight coefficient. By connecting each convolutional kernel to the previous layer a receptive field of certain size can be obtained. Then the elements of the input

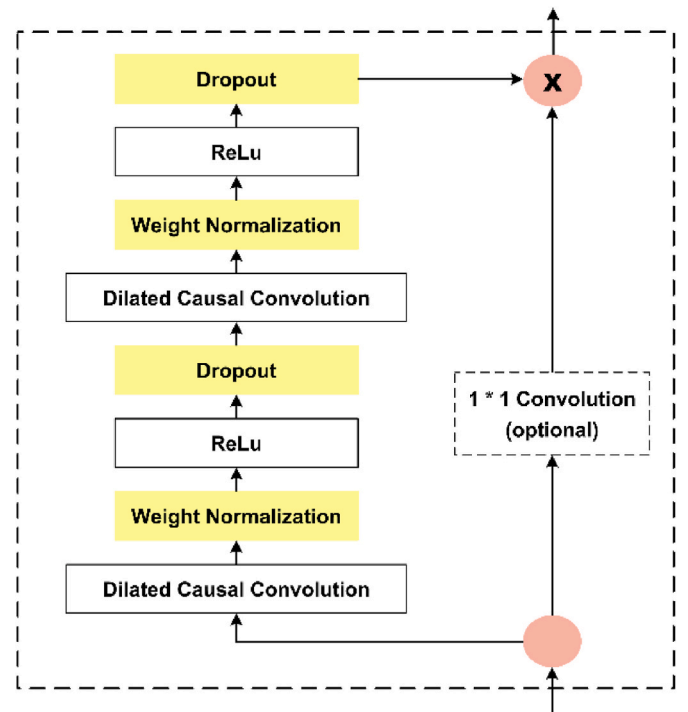


Fig. 8. Residual block [64].

features present within the feature map are matrix multiplied with the weights and summed up, which in turn is finally added to a bias. Equation (5) represents the calculation carried out.

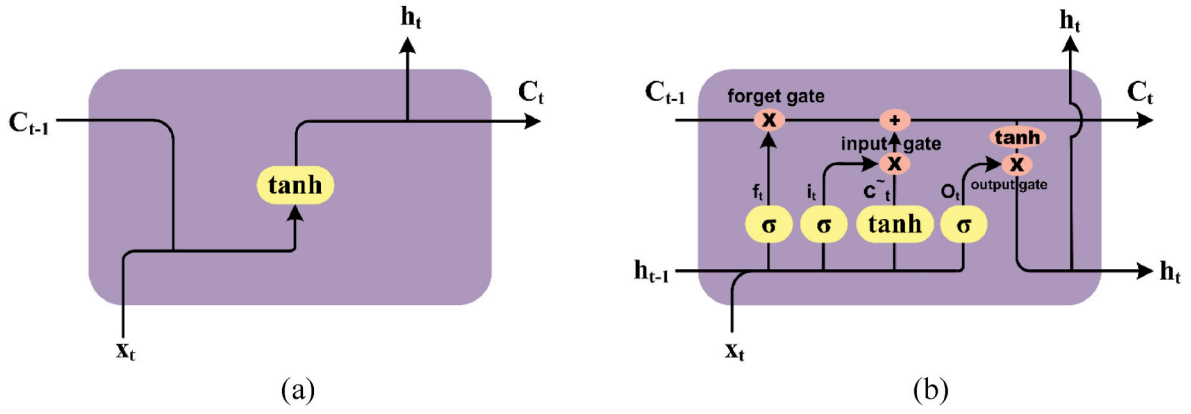


Fig. 9. (a) RNN and (b) LSTM [4].

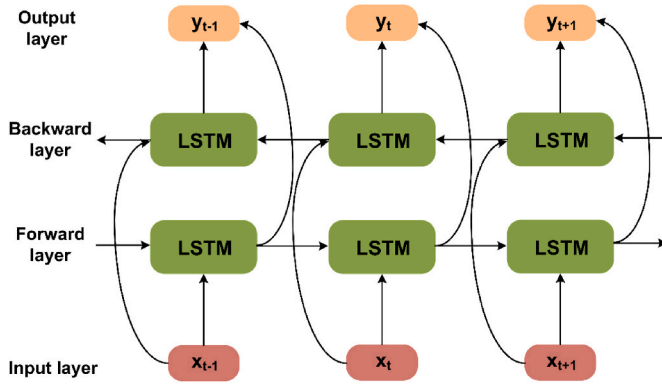


Fig. 10. Schematic of BiLSTM network [67].

$$Z^{(l+1)}(i,j) = f \left(\sum_{k=1}^K Z^l(i,j) \otimes \omega_k^{(l+1)} + b^{(l+1)} \right) \quad (5)$$

where ' $Z^l(i,j)$ ' and ' $Z^{(l+1)}(i,j)$ ' denote the input and output of the ' l 'th and ' $l+1$ 'th layer, respectively. ' $\omega_k^{(l+1)}$ ' is the neuron weight at the ' $l+1$ 'th layer. The convolutional kernels are actually these neurons. The subscript ' k ' denotes the ' k 'th convolutional kernel. ' K ' represents the number of the convolutional kernels in a layer and ' $b^{(l+1)}$ ' is the bias in the ' $l+1$ 'th layer. The symbol ' $\otimes$ ' represents the convolutional operation while ' $f(\cdot)$ ' denotes the activation function. In this study rectified linear unit (ReLU) was used as the activation function.

The size of the kernel, stride and padding determine the size of the feature map at the output of the convolution layer. Any value less than the size of the input data can be used to specify the size of the convolution kernel. Kernel with large size allows for more complicated input feature extraction. The kernel's stride determines how far it travels during each sweep. As the convolution layers are stacked, the feature

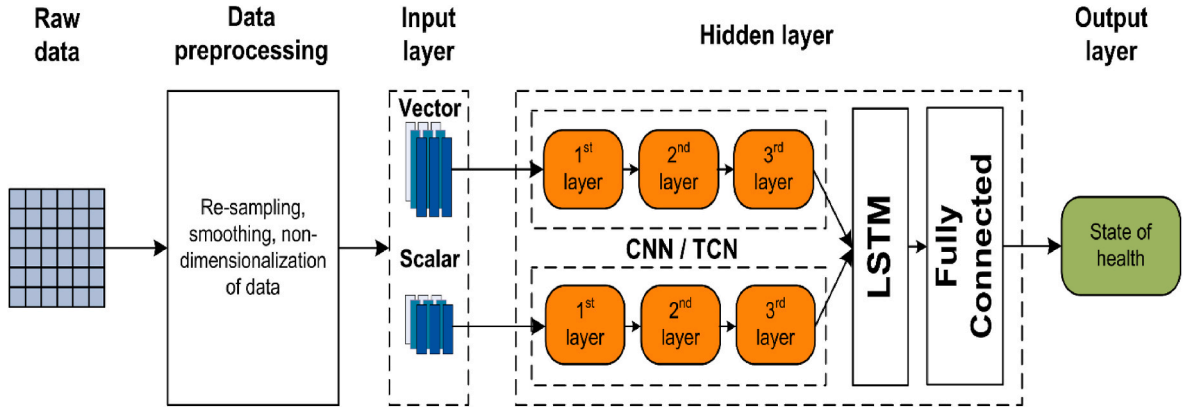


Fig. 11. Proposed framework.

Table 2

Range of Hyperparameters for optimization.

Hyperparameters	CNN	TCN	CNN-LSTM	TCN-LSTM	CNN-biLSTM	TCN-biLSTM
Number of filters in TCN or CNN (N_f)	20, 40	20, 40	20, 40	20, 40	20, 40	20, 40
Kernel size of CNN or TCN (K_s)	4, 10	2, 10	4, 10	2, 10	4, 10	2, 10
Number of strides in CNN (N_s)	1, 4	—	1, 4	—	1, 4	—
Number of units in the LSTM/BiLSTM layer (U_L)	—	—	100, 140	100, 140	100, 140	100, 140
Number of units in the Dense layer (U_D)	20, 40	20, 40	20, 40	20, 40	20, 40	20, 40
Dropout rate CNN or TCN (D_r)	0.3, 0.6	0.3, 0.6	0.3, 0.6	0.3, 0.6	0.3, 0.6	0.3, 0.6
Dropout rate LSTM/BiLSTM (D_L)	—	—	0.3, 0.6	0.3, 0.6	0.3, 0.6	0.3, 0.6
Batch size (B_s)	28, 38	28, 38	28, 38	28, 38	28, 38	28, 38
Learning rate (L_r)	0.0001, 0.00015	0.0001, 0.00015	0.0001, 0.00015	0.0001, 0.00015	0.0001, 0.00015	0.0001, 0.00015

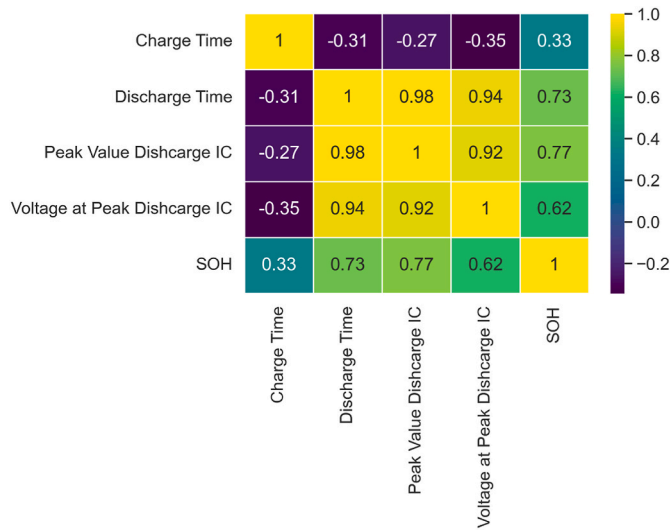


Fig. 12. Spearman correlation analysis.

map will become smaller. Hence, it is necessary to increase the feature map size before convolution which can be carried out by providing padding. The padding can be either ‘same’ or ‘valid’. The term ‘same

padding’ refers to the use of sufficient padding to maintain the same size feature maps for the input and output. On the other hand, the ‘valid padding’ only covers the actual input and does not include any artificial zero-padding.

Now, suppose the size of the input data is ‘ $D \times D$ ’, the size of the convolution kernel is ‘ $K \times K$ ’, the stride is ‘ S ’, the padding is ‘ P ’ and output feature map size after convolution is ‘ $F \times F$ ’, then equation (6) must be satisfied.

$$F = \left\lfloor \frac{D - K + (2 * P)}{S} + 1 \right\rfloor \quad (6)$$

In the above equation, when the internal number is a decimal, the fractional component is eliminated and just the integer part is kept, as indicated by the rounding down symbol $\lfloor \cdot \rfloor$.

3.1.2. Pooling layer

Between two consecutive convolution layers, a sub-sampling layer also known as pooling layer is included. By sub-sampling the feature map derived from the convolution, it can decrease the number of network parameters, speed up calculation, and along with overfitting it can also reduce the dimensions of the data. ‘Max’ and ‘mean’ pooling are the generally used pooling methods. Equation (7) represents the expression for the pooling method.

Table 3
Optimized hyperparameters of CNN and TCN.

Hyperparameters	CNN	CNN	CNN	TCN	TCN	TCN
	1 layer	2 layer	3 layer	1 layer	2 layer	3 layer
N_f	39	31	39	28	20	32
K_s	7	7	7	4	9	8
N_s	3	3	3	–	–	–
U_D	28	24	28	22	38	30
D_r	0.3711	0.5634	0.3711	0.5848	0.4072	0.3160
B_s	34	34	34	34	34	29
L_r	0.0001452	0.0001208	0.0001452	0.0001225	0.0001454	0.0001287

Table 4
Optimized hyperparameters of CNN and TCN with LSTM.

Hyperparameters	CNN	CNN	CNN	TCN	TCN	TCN
	1 layer LSTM	2 layer LSTM	3 layer LSTM	1 layer LSTM	2 layer LSTM	3 layer LSTM
N_f	38	38	38	35	29	31
K_s	8	8	8	8	10	9
N_s	1	1	1	–	–	–
U_L	110	110	110	107	114	110
U_D	20	20	20	23	38	20
D_r	0.3078	0.3078	0.3078	0.4193	0.5253	0.3085
D_L	0.3085	0.3085	0.3085	0.5422	0.4871	0.3078
B_s	33	33	33	31	37	33
L_r	0.0001430	0.0001430	0.0001430	0.0001464	0.0001135	0.0001430

Table 5
Optimized Hyperparameters of CNN and TCN with biLSTM.

Hyperparameters	CNN	CNN	CNN	TCN	TCN	TCN
	1 layer biLSTM	2 layer biLSTM	3 layer biLSTM	1 layer biLSTM	2 layer biLSTM	3 layer biLSTM
N_f	23	28	40	31	38	39
K_s	5	4	7	9	5	9
N_s	3	3	2	–	–	–
U_L	135	123	111	110	125	130
U_D	39	33	31	20	38	31
D_r	0.394	0.454	0.523	0.3085	0.3008	0.3757
D_L	0.507	0.583	0.501	0.3079	0.4721	0.3609
B_s	29	29	32	33	33	34
L_r	0.0001447	0.0001451	0.0001033	0.0001430	0.000116	0.000109

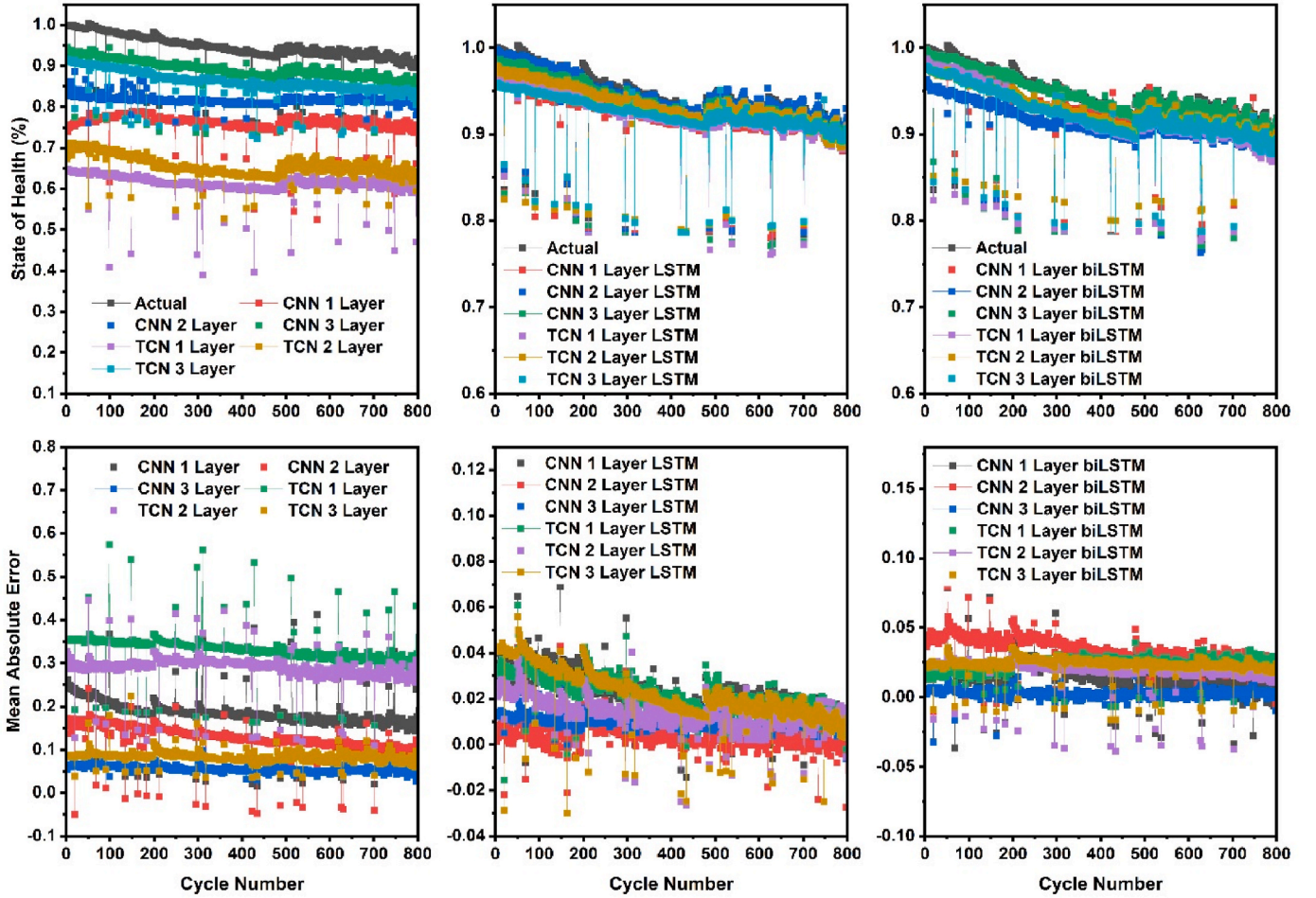


Fig. 13. Performance of various models for Cell 1.

$$y(i, j)_k = \left[\sum_{(m, n) \in R_{i, j}} (o(m, n)_k)^p \right]^{1/p} \quad (7)$$

where ' $y(i, j)_k$ ' is the pooling operator output at the position ' i, j ' of the ' k th' feature map. ' $o(m, n)_k$ ' is the eigen value at the position ' m, n ' of the pooling region ' $R_{i, j}$ ' in the ' k th' feature map. When ' p ' tends to infinity, the output is 'max' pooling, while when ' p ' is equal to '1', it is known as mean pooling.

Suppose the size of the input feature map of the pooling layer is ' $D \times D$ ', the size of the convolution kernel is ' $K \times K$ ', the stride is ' S ', and output feature map size after pooling is ' $F \times F$ ', then equation (8) must be satisfied.

$$F = \left\lfloor \frac{D - K}{S} + 1 \right\rfloor \quad (8)$$

As discussed earlier, the symbol ' $\lfloor \cdot \rfloor$ ' refers to rounding of the output to the nearest integer.

3.2. Temporal convolutional network

Using a hierarchy of convolutional filters, the TCN technique was first created to investigate long-range patterns (Fig. 6a) [55]. The important attributes of TCNs are: (1) it makes use of causal convolutions and (2) the network may map an input pattern of any length to an output pattern of the same length, much like in an RNN. Causal convolution is employed to prevent previous data from being lost (Fig. 6b) [56]. Let us suppose an input sequence $X = [x_1, x_2, x_3, \dots, x_T]$ and output sequence Y

$= [y_1, y_2, y_3, \dots, y_T]$, the prediction of the output ' y_t ' can be made by the causal convolution using the inputs from previous and present time steps, i.e., $x_1, x_2, x_3, \dots, x_t$. In other words, the output ' y_t ' cannot be predicted using the future inputs, i.e., $x_{t+1}, x_{t+2}, x_{t+3}, \dots, x_T$. Zero padding is used to make certain that the dimensions of the output and the input are equal. To determine causality, padding is done asymmetrically to the left.

For modelling time-series data having long-term dependencies, large receptive field is typically needed. If the size of the receptive field is to be increased, then either very large filters or an extensive array of convolution layers is required. Both the options, however, are computationally inefficient [57]. Dilating the filters is one of the ways of overcoming the problem (Fig. 6c). The kernel can be enlarged by dilated convolution, by creating gaps between the inputs. A ' k ' sized convolution filter and ' d ' dilation factor convolution layer can look ' $d \cdot (k-1)$ ' time steps into the past. Here is the formulation of the whole dilated causal convolution process over successive layers [58]:

$$Z_t^k = f \left(\sum_{k=0}^{K-1} Z_{(t-1)}^{(t-(k \cdot d))} * \omega_t^k + b_t \right) \quad (9)$$

Where ' Z_t^k ' is the neuron output at position ' t ' in the ' k th' layer, ' K ' is the convolutional kernel size, ' ω_t^k ' is weight at position ' k ', ' d ' is the dilation factor and ' b_t ' is the bias term. ' $f(\cdot)$ ' denotes the activation function. In this study rectified linear unit (ReLU) was used as the activation function. Another way to increase the receptive field of the network is to join several TCN blocks (Fig. 7) [59]. For the first layer, the dilation factor is set to 1, which is exponentially increased for the next layer. This is to

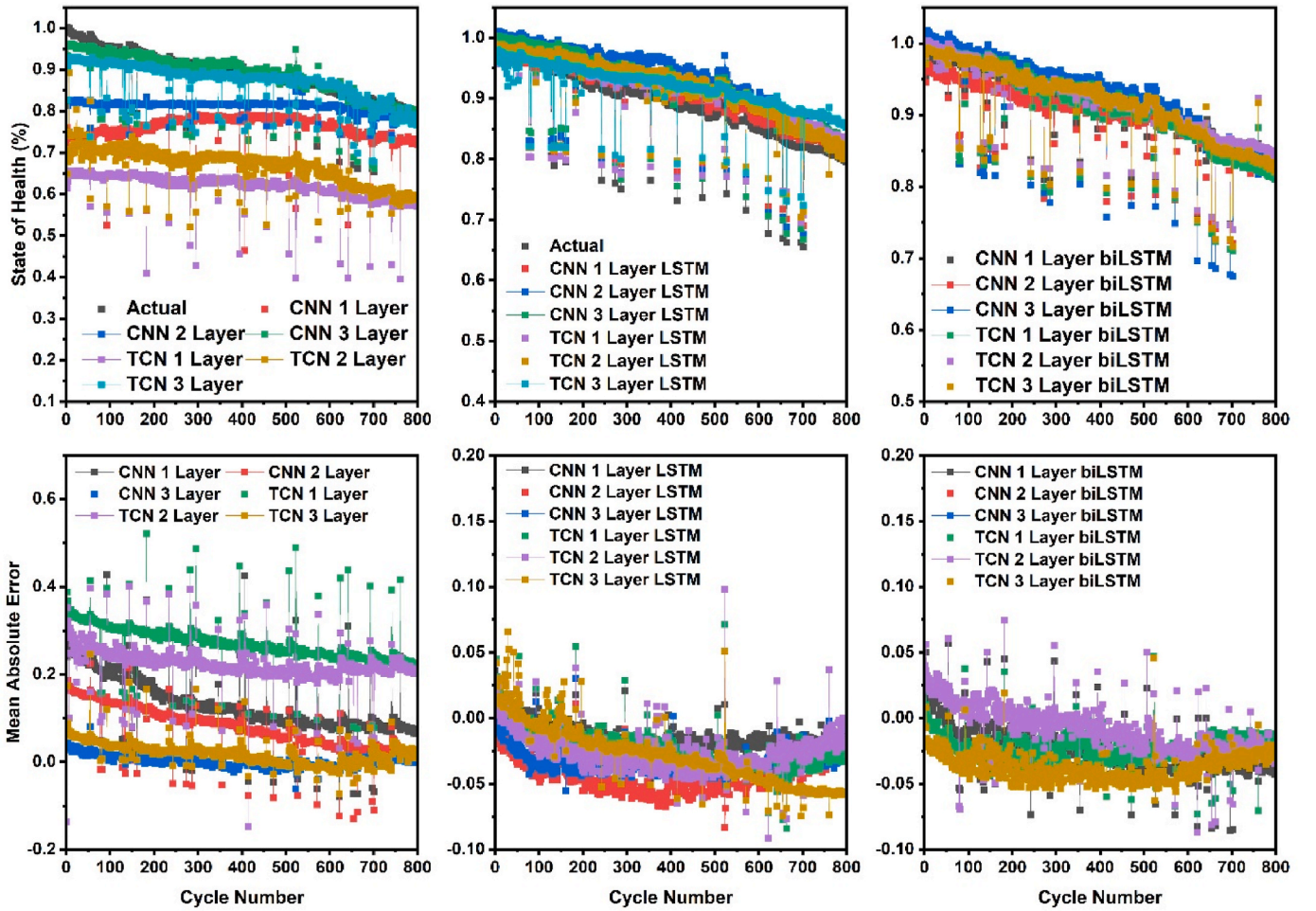


Fig. 14. Performance of various models for Cell 2.

guarantee that the receptive field can handle all inputs. Furthermore, the receptive field's size grows exponentially with depth. Due to this, fewer layers are needed than a traditional convolution to attain the same receptive field size. Nevertheless, this results in deeper structures with a lot more parameters, which makes the learning process more difficult because of the increasing/vanishing gradient issue. Connecting the output of the TCN block to a residue block will result in the improvement in the training of the model [60]. The residue block's inputs are supplemented with the outputs of the last dilated convolution layer (Fig. 8). The subsequent residue block then uses the results as its inputs. Abdi and Nahavandi [61] utilized skip connections in residue blocks resulting in the enhancement of the training stability by maintaining the gradient's norm. To guarantee that the residual block inputs and outputs on the skip connection have the same size, the inputs can optionally be transformed using a 1×1 convolution. Two sub-blocks are present in every residue block, layered on top of one another [62]. A stacked dilated causal convolution layer, a 'ReLU' activation function, and a dropout layer make up each sub-block. Batch or weighted normalization can also be included in the dilated causal convolution layer.

TCNs are ideal for deep learning complicated time series problems due to the above-mentioned unique properties. Similar to RNN, TCN can manage inputs with varied lengths by sliding the causal convolutional kernel in one dimension. Furthermore, due to their shared convolution architecture, which enables them to analyze lengthy sequences in parallel, TCNs are more memory efficient than RNNs. Higher computation times are caused by RNNs' sequential processing of the input sequences. Additionally, TCNs are trained using the normal backpropagation technique, which spares them from the gradient issues associated with

RNNs' usage of the backpropagation-through-time algorithm (BPTT) [63].

3.3. Long short-term memory network

The LSTM network is a type of RNN that is often used for modelling time-series data. The creation of gate-controlled memory units in LSTM addresses the gradient disappearing and erupting problems that are frequently present in conventional RNNs (Fig. 9a). Longer time series can also be remembered by the network due to this design. The LSTM network along with an external recurrent has an internal memory recurrent unit, which consists of gating unit. The network is depicted in Fig. 9b. During training, the forget, input, and output gate combine to allow the network to write data to the memory cell while forgetting some irrelevant details.

In LSTM, the input data ' x_t ' at the current moment ' t ' is combined with ' h_{t-1} ', which is the output of the previous moment ' $t-1$ '. The combine input data is known as ' x_c '. The hyperbolic tangent function denoted as ' $\tanh$ ' layer then squeezes the amalgamated input. The resultant squeezed value, designated as candidate memory cell ' $\tilde{c}_t$ ', next will be multiplied by the input gate's computed value. The result is then sent to the memory unit. Concurrently, an additional combined input is supplied to the forget gate and multiplied by the memory unit's last iteration state, ' c_{t-1} '. Similarly, this result is also transmitted to the memory unit. After the memory unit processes these two inputs, a ' $\tanh$ ' layer will squeeze them once more. In the end, the data obtained from the squeezed layer is multiplied with the input data processed by the output gate resulting in the yield of the output ' h_t '.

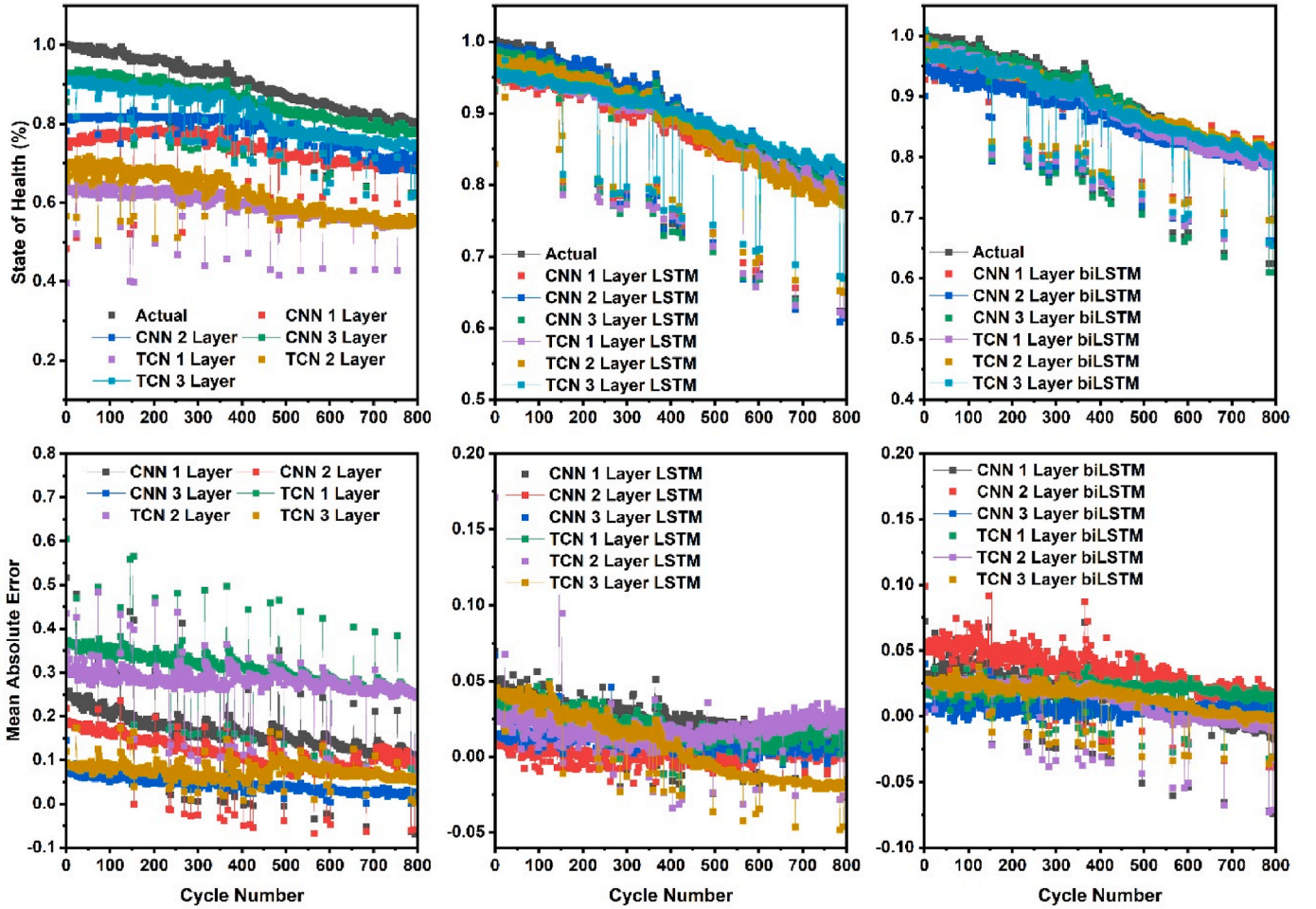


Fig. 15. Performance of various models for Cell 3.

The input gate makes the decisions about which input values should be used and whether or not to accept fresh inputs. The input gate, expressed by eq. (10), is a sigmoid activation function adopted layer.

$$i_t = \sigma(\omega_{xi} * x_t + \omega_{hi} * h_{t-1} + b_i) \quad (10)$$

where ' σ ' is the sigmoid function, ' ω_{xi} ' and ' ω_{hi} ' are the weights of the input at the current moment and the output of the previous moment, respectively, at the input gate, while ' b_i ' is the bias at the input gate. The memory cell candidate is calculated using eq. 11

$$\tilde{c}_t = \tan h(\omega_{xc} * x_t + \omega_{hc} * h_{t-1} + b_c) \quad (11)$$

The forget gate determines the relevance of information and whether it should be iterated to the next loop, as well as how many pieces to retain. The layer also uses sigmoid function as the activation function whose output is between 0 and 1. Zero in forget gate means that the information is not useful and can be rejected, while one means that the information is used in its entirety. Equation (12) is used for the calculating the output of the forget gate.

$$f_t = \sigma(\omega_{xf} * x_t + \omega_{hf} * h_{t-1} + b_f) \quad (12)$$

The memory unit can now be calculated using equation (13).

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (13)$$

where ' $\odot$ ' symbolises the multiplication of the elements one-by-one. Sigmoid function is also used by the output gate. Eq. (14) is used for calculating the output of the output gate.

$$o_t = \sigma(\omega_{xo} * x_t + \omega_{ho} * h_{t-1} + b_o) \quad (14)$$

The output of the LSTM can be calculated using equation (15).

$$h_t = o_t \odot \tanh(c_t) \quad (15)$$

where ' h_t ' is the output of the LSTM at the current moment. It can thus be seen that the memory cell and output gate work together to provide this combined operation outcome.

3.4. Bi-directional LSTM network

Although LSTM outperforms RNN, it can only handle past information sequentially. In contrast, BiLSTM (Bi-directional LSTM) also makes use of future knowledge. BiLSTM architecture, depicted in Fig. 10, shows that it has both forward and backward LSTM layers. The output layer processes inputs from both the forward and backward levels simultaneously. The network training can be considered as a standard uni-directional LSTM since the neurons in the both backward and forward networks are not coupled to one another [65].

Consider that the input data for a time duration ($1 \leq t \leq T$) is run through the network. The BiLSTM updates the output layer by computing the forward hidden layer ($\vec{h}_t$), backward hidden layer ($\overleftarrow{h}_t$), and output sequence (y_t). The backward layer is iterated from time $t = T$ to 1, while the forward layer is iterated from time $t = 1$ to T . The updating of the network can be calculated using equations, 16, 17 and 18.

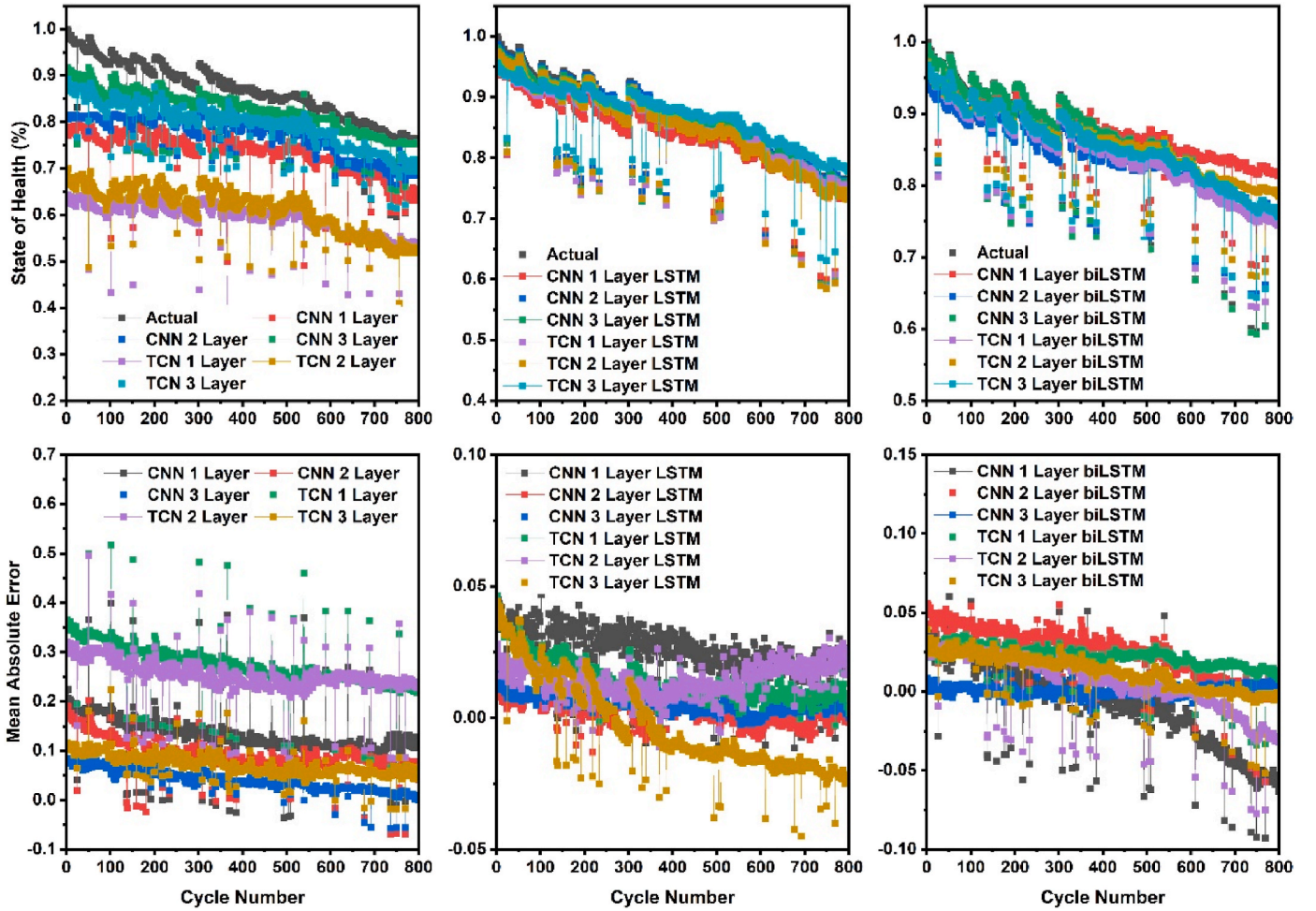


Fig. 16. Performance of various models for Cell 4.

$$\vec{h}_t = H(W_1 x_t + W_2 \vec{h}_{t-1} + \vec{b}) \quad (16)$$

$$\overleftarrow{h}_t = H(W_3 x_t + W_5 \overleftarrow{h}_{t-1} + \overleftarrow{b}) \quad (17)$$

$$y_t = W_4 \vec{h}_t + W_6 \overleftarrow{h}_t + b_y \quad (18)$$

Where, W 's are the weight coefficient and ' $\vec{b}$ ', ' $\overleftarrow{b}$ ' & ' b_y ' are the bias vectors for equations (16)–(18), respectively. BiLSTM topologies can outperform other network designs in specific problem areas. BiLSTMs [66] have proven to be effective in speech processing tasks with high content importance.

3.5. Fully connected layer

After the last convolution-pooling layer/TCN layer/LSTM layer, a fully connected layer is placed in the network. All the features extracted earlier are connected and the values are sent to the output layer. Due to this connection many feature maps will be generated as the convolutional kernels can be of different sizes and numbers. These feature maps will be a higher-dimension matrix. Therefore, with the help of flattening layer, the feature maps are flattened to a 1D shape and fed to the fully connected layer.

4. Experiments and analysis

4.1. Proposed framework

The proposed framework, shown in Fig. 11, contains three modules. First is the pre-processing of the data as discussed in the previous sections. Next is the optimization of the hyperparameters using Bayesian optimization algorithm, which will be discussed in the next section. The last module is the estimation of the cell state-of-health using CNN/TCN models along with LSTM network, one dropout layer and dense layer.

The deep learning framework, TensorFlow (TF) was used to build the network and its high-level API, Keras, was used for implementing the corresponding functions.

4.2. Hyperparameters

The hyperparameters of the network have to be initialized for optimum training of the model. For CNN, the number of filters, kernel size, number of strides and dropout rate needs to be optimized. The padding and pooling methods are also required. For TCN, the number of filters, kernels, dropout rate and dilation factor are required. While for LSTM, the number of neurons and dropout rate is required. The number of neurons in the dense layer, the learning rate and batch size for the model are also required to be optimized. All the parameters were optimized using Bayesian optimization algorithm. The range of the parameters used for optimization is given in Table 2. The Bayesian optimization algorithm tunes the hyperparameters of a machine learning model using the Gaussian process. It uses the Bayesian formula to derive probabilistic

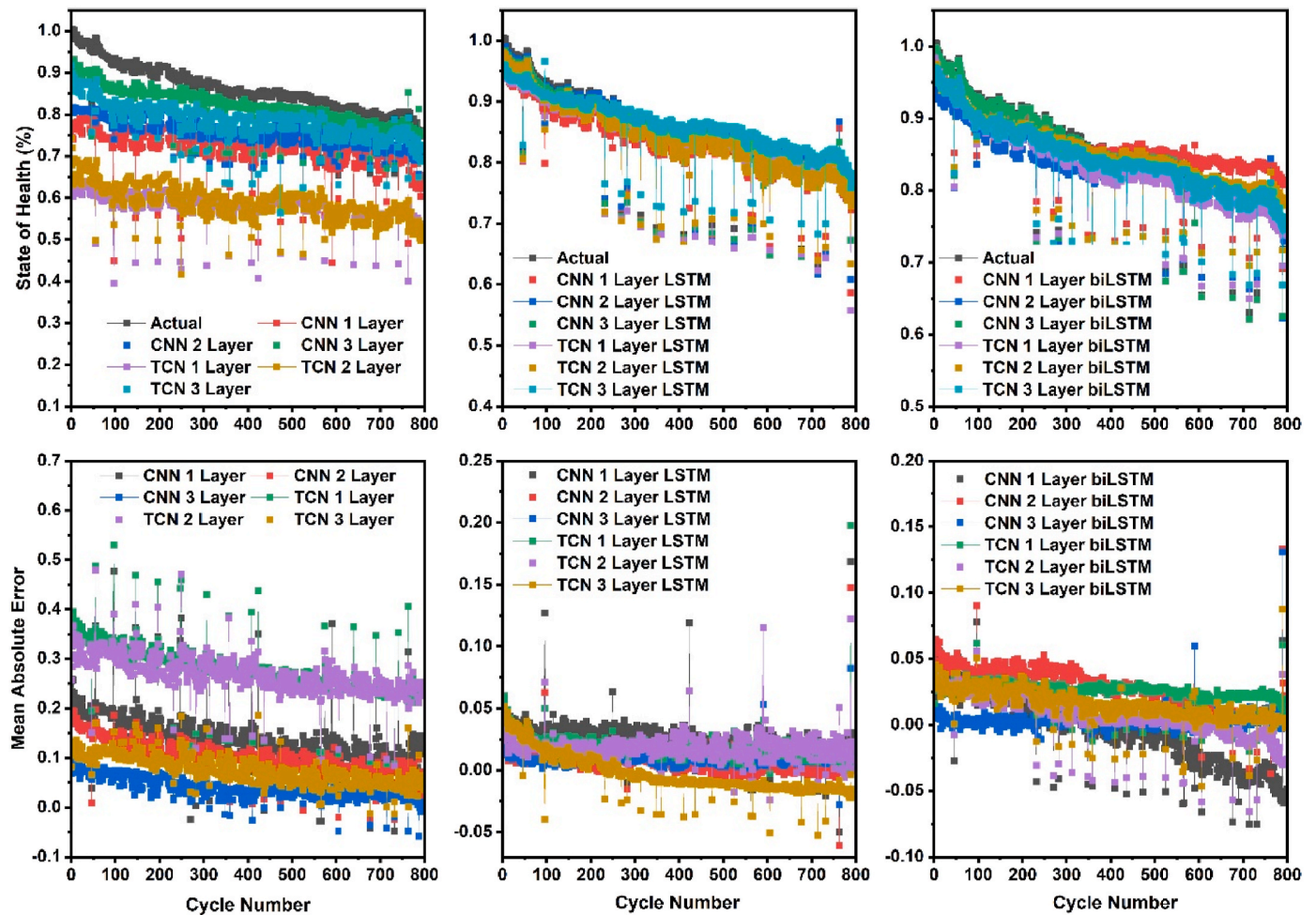


Fig. 17. Performance of various models for Cell 5.

understanding of the function distribution by combining sample data with prior knowledge about the unknown function. The location of the function's optimal value can then be inferred using this probabilistic data. Studies have shown that as compared to other global optimization techniques, the Bayesian optimization algorithm performs better [68].

To assess the error between the actual and the predicted value, 'mean square error' loss was used as the loss function. The learning rate drives the network convergence and governs the loss function's gradient update step at each iteration. In the present study, the learning rate for optimization was varied from 0.0001 to 0.00015. Another variable that may impact the rate and precision of network convergence is the optimizer, which uses the gradient descent computation method. More recently, in the field of deep learning, the Adaptive Moment Estimation (Adam) optimizer has produced improved practical results. The training dataset must be split into multiple batches because it cannot be put into the network all at once. The quantity of data in each batch is referred to as the batch size. In the present study the batch size was varied from 28 to 38 for optimization. The network after reading all the batches modifies the model parameters by calculating the gradient of the loss function using the optimizer. It is not feasible to obtain the precise model parameters in a single update; so, the data must be repeatedly iterated. 1200 iterations were carried out for each model. As the iteration continues, overfitting will inevitably occur. Here, the dropout technique is used to avoid overfitting. The dropout rate of CNN, TCN, LSTM and biLSTM was varied from 0.3 to 0.6 for optimization. A model's performance is typically expressed as the validation set's loss, with a smaller loss signifying superior performance.

CNN/TCN can extract features from the data that are hidden. Adding

more layers improves prediction accuracy by extracting deeper correlations between data. However, adding layers increases network complexity and increases the likelihood of local optimum or even overfitting of the model. Therefore, appropriate number of layers should be chosen for the model. In this study, the number of layers in CNN and TCN were varied from 1 to 3.

4.3. Evaluating the model

After completion of training of the model, both the training and the testing dataset are fed to the model for estimating the state of health of cell. The mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE) and goodness of fit (R^2). Equations (16)–(19) were used to calculate the MAE, MAPE, RMSE and R^2 , respectively.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (16)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (17)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (18)$$

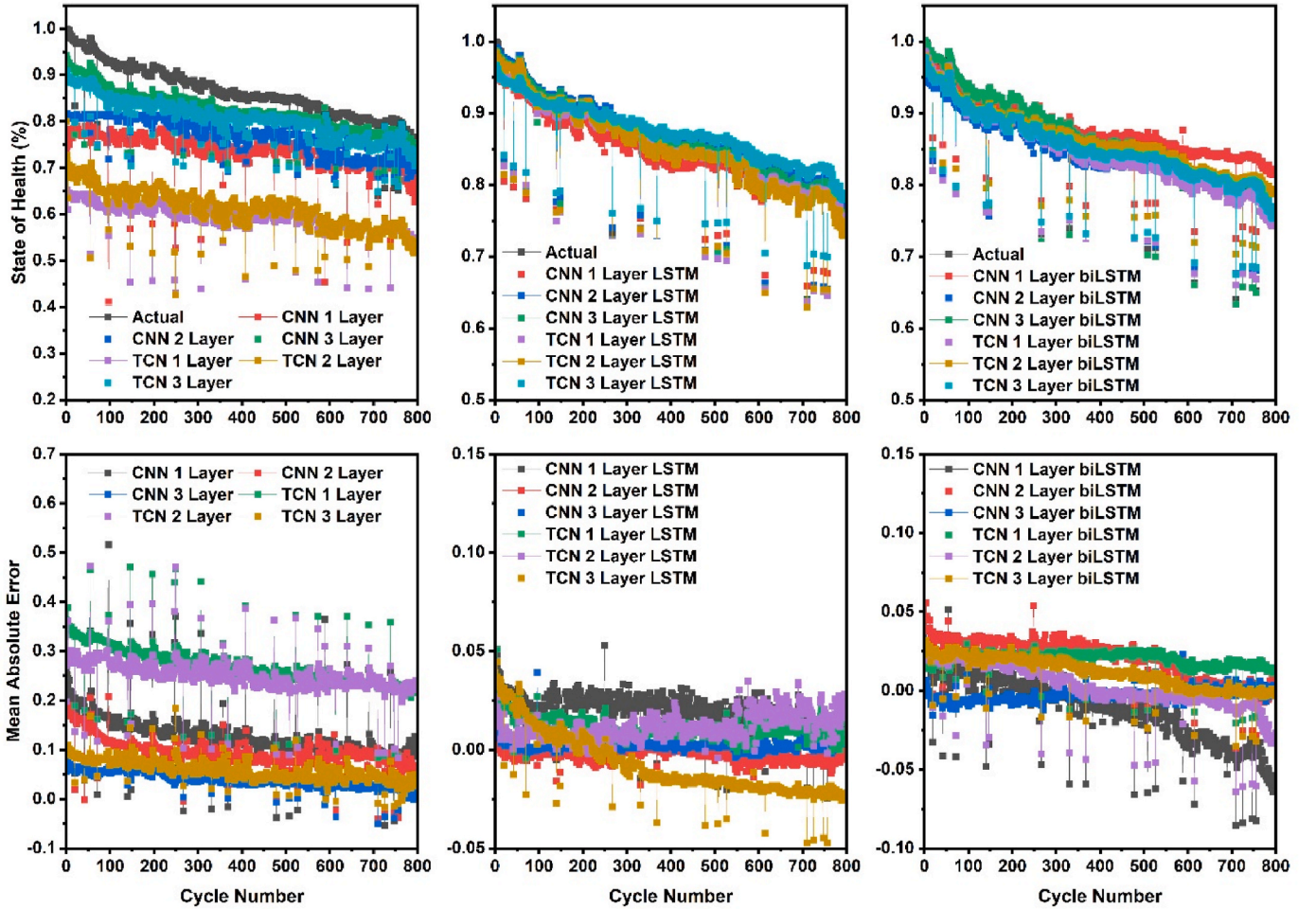


Fig. 18. Performance of various models for Cell 6.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (\bar{y} - \hat{y}_i)^2} \quad (19)$$

Where 'n' denotes the number of samples, ' y_i ' denotes the true value, ' $\hat{y}_i$ ' denotes the predicted value and ' $\bar{y}$ ' denotes the mean value.

5. Results and discussion

5.1. Feature analysis

While testing a lithium-ion cell, data for a number of parameters is generated. If all the data is utilized for training a model then its efficiency may be greatly reduced. Therefore, it is necessary to analyze the parameters and select the ones which have a strong correlation with the state of health of the lithium ion battery. The features used for the analysis were charge time, discharge time, peak of the discharge IC curve and the voltage at which the peak occurs.

Selecting the correct analysis method for correlation is also necessary. Commonly used methods are Pearson and Spearman. Pearson method is useful for linear correlations, while Spearman is useful for non-linear correlations. The relationships among the parameters of a lithium-ion battery are non-linear. Hence, the Spearman correlation method was used in this study. The Spearman correlation coefficient can be calculated using equation (20).

$$R = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (20)$$

Where 'n' is sample size and ' x_i ', ' y_i ', ' $\bar{x}$ ' and ' $\bar{y}$ ' are the values and means of the parameters, respectively. Fig. 12 shows the coefficients for the parameters used in this study. For calculating the coefficients both the training and the test set was considered. All the parameters have positive correlation with the state of health of the battery, i.e., changes in these parameters will have an effect on the health of the battery. Another interesting fact is that the discharge time and both the parameters of IC curve are highly correlated with each other. This shows that if the peak of the IC curve decreases or the voltage at which the peak occurs decreases, the discharge time decreases. On the other hand the three parameters have a negative correlation with battery charging time, i.e., the charging time increases with decrease in peak of the IC curve.

5.2. Hyperparameter optimization

To get the best performance from all the models trained in this study, the hyperparameters need to be optimized individually for the models. The hyperparameters namely number of filters, kernel size, number of strides, number of units in LSTM/biLSTM, number of units in dense layer, dropout rate, batch size and learning rate were optimized using Bayesian optimization algorithm. A maximum of 24 evaluations were carried out for finding the optimized hyperparameters of all the models.

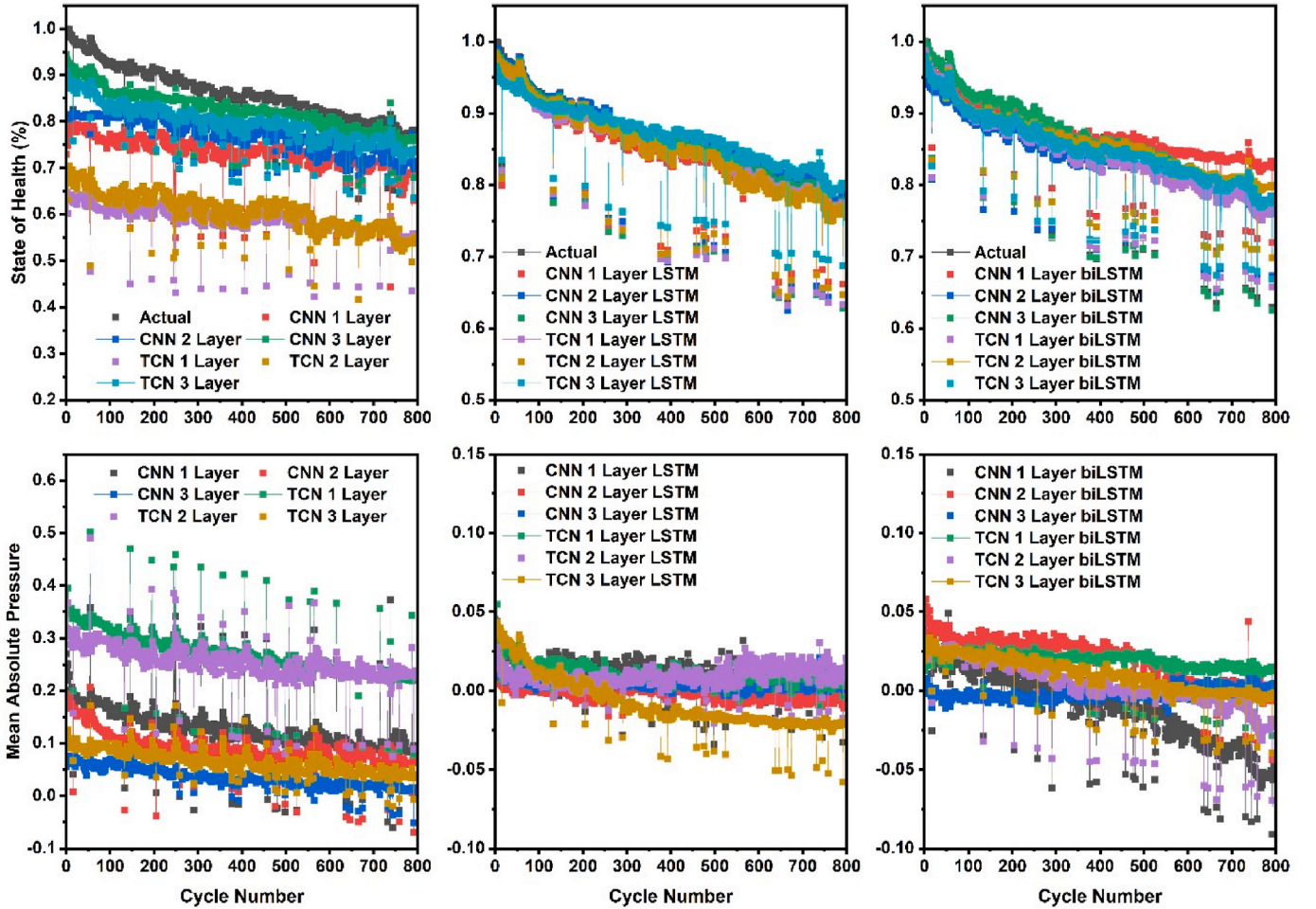


Fig. 19. Performance of various models for Cell 7.

Tables 3–5 tabulates the optimized hyperparameters for CNN/TCN without and with LSTM/biLSTM. The mean absolute percentage error of the whole dataset was chosen as the criteria for optimizing the hyperparameters. To avoid overfitting, every epoch the data was randomly shuffled. Also, the TCN's dilation factor was manually optimized to Refs. [1,2,4,8,12].

5.3. State of health estimation

The prediction of state of health of the cells using the different models is shown in Figs. 13–19. While all models can capture the fluctuation characteristics of SOH, the absolute error increases when SOH fluctuates due to the difficulty in identifying features under these situations. The least fluctuations were found in CNN 3-layer biLSTM model. The results of R^2 , RMSE and MAPE with the six tested cells for all the models is given in Table 6. The R^2 value for training cells varied from -4.74 (Cell 6) with TCN 1-layer to 0.9994 (Cell 7) with CNN 3-layer biLSTM. While the value varied from -2.138 (Cell 2) with TCN 1-layer with to 0.992 (Cell 5) with CNN 3-layer biLSTM for the test set. The negative value indicates that the model is poorly trained and is not suitable for prediction. The values of R^2 closer to 1 are considered the best. The model trained with CNN 3-layer LSTM and CNN 3-layer biLSTM architecture resulted in R^2 values closer to 1. The mean absolute percentage error for the training set varied from 0.317 (Cell 1) with TCN 1-layer to 0.0040 (Cell 6) with CNN 3-layer biLSTM. For the test set the MAPE varied from 0.265 (Cell 5) with TCN 2-layer to 0.011 (Cell 5) with CNN 3-layer biLSTM. The lower the value of MAPE, the better the model. It can be seen that CNN 3-layer LSTM and CNN 3-layer biLSTM

resulted in lowest values of MAPE for both the training and the test set. The RMSE for the test set varied from 0.219 (Cell 2) with TCN 2-layer to 0.010 (Cell 5) with CNN 3-layer LSTM. While the RMSE for the training set varied from 0.262 (Cell 1) with TCN 2-layer to 0.0032 (Cell 6) with CNN 3-layer LSTM. Overall, the accuracies of the cells used for training and testing is nearly similar. The results further reveal that the models can accurately predict the state of health below 80 % as well.

Fig. 20 depicts the RMSE of overall, training & test dataset and overall R^2 value. The overall RSME was found to be lowest with CNN 3-layer LSTM model. It was found to be $26.72, 33.55, 18.78, 37.18, 25.69, 13.28, 2.75, 0.28, 4.96, 3.29, 2.75, 21.69, 8.1, 3.1, 6.78, 12.8$ and 7.65 times lower than CNN 1-layer, CNN 2-layer, CNN 3-layer, TCN 1-layer, TCN 2-layer, TCN 3-layer, CNN 1-layer LSTM, CNN 2-layer LSTM, TCN 1-layer LSTM, TCN 2-layer LSTM, TCN 3-layer LSTM, CNN 1-layer biLSTM, CNN 2-layer biLSTM, CNN 3-layer biLSTM, TCN 1-layer biLSTM, TCN 2-layer biLSTM, and TCN 3-layer biLSTM, respectively. The overall R^2 value was found to be the highest with CNN 2-layer LSTM, CNN 3-layer LSTM and CNN 3-layer biLSTM. Moreover, the R^2 value for training and test dataset with CNN 2-layer LSTM and CNN 3-layer LSTM were found to be similar. Whereas, the R^2 value for test dataset with CNN 3-layer biLSTM was found to be low. Thus, showing that the model had lower accuracy in predicting the state of health from unknown data. The training loss and validation loss for CNN 2- layer and 3-layer with LSTM is shown in Fig. 21. The figure reveals that the training loss is lower than the validation loss i.e., the model is slightly overfitting. Also, the validation loss diverges with CNN 2-layer LSTM architecture. The maximum R^2 of 0.993 was achieved with CNN 3-layer LSTM model. The RMSE and MAPE value with the model was 0.014 and

Table 6

SOH estimation errors of various models.

Cell	Network	MAPE	R ²	RMSE
Cell 1	CNN 1 Layer	0.151647	−0.24203	0.253104
	CNN 1 Layer LSTM	0.01629	0.991459	0.020989
	CNN 2 Layer	0.102073	0.807701	0.099591
	CNN 2 Layer LSTM	0.004162	0.999056	0.006977
	CNN 3 Layer	0.040215	0.953611	0.048915
	CNN 3 Layer LSTM	0.007074	0.99851	0.008768
	TCN 1 Layer	0.317754	−0.6805	0.29441
	TCN 1 Layer LSTM	0.016726	0.984986	0.027828
	TCN 2 Layer	0.293707	−0.33448	0.262355
	TCN 2 Layer LSTM	0.016205	0.995226	0.015692
	TCN 3 Layer	0.067702	0.854956	0.086494
	TCN 3 Layer LSTM	0.025711	0.988875	0.023954
	CNN 1 Layer bilstm	0.030989	0.979414	0.032585
	TCN 1 Layer biLSTM	0.021484	0.991051	0.021484
	CNN 2 Layer biLSTM	0.021941	0.98088	0.031403
	TCN 2 Layer biLSTM	0.027641	0.984607	0.028177
	CNN 3 Layer biLSTM	0.004419	0.998912	0.007492
	TCN 3 Layer biLSTM	0.016185	0.993733	0.017979
Cell 2	CNN 1 Layer	0.104583	0.264354	0.106104
	CNN 1 Layer LSTM	0.038729	0.927596	0.033287
	CNN 2 Layer	0.08543	0.580247	0.080148
	CNN 2 Layer LSTM	0.048631	0.895475	0.039995
	CNN 3 Layer	0.044607	0.872219	0.044221
	CNN 3 Layer LSTM	0.041414	0.928305	0.033124
	TCN 1 Layer	0.247833	−2.13162	0.218918
	TCN 1 Layer LSTM	0.033518	0.949892	0.027692
	TCN 2 Layer	0.219973	−2.13806	0.219143
	TCN 2 Layer LSTM	0.029839	0.949041	0.027926
	TCN 3 Layer	0.046544	0.874392	0.043843
	TCN 3 Layer LSTM	0.07187	0.795473	0.055946
	CNN 1 Layer biLSTM	0.093096	0.611876	0.077069
	TCN 1 Layer biLSTM	0.044135	0.918442	0.035329
	CNN 2 Layer biLSTM	0.048859	0.889924	0.041043
	TCN 2 Layer biLSTM	0.077394	0.758564	0.060785
	CNN 3 Layer biLSTM	0.046847	0.911441	0.036814
	TCN 3 Layer biLSTM	0.051253	0.891557	0.040738
Cell 3	CNN 1 Layer	0.128139	−0.0371	0.130582
	CNN 1 Layer LSTM	0.021785	0.97178	0.02154
	CNN 2 Layer	0.115251	0.356955	0.102824
	CNN 2 Layer LSTM	0.005683	0.997958	0.005794
	CNN 3 Layer	0.05229	0.869939	0.046243
	CNN 3 Layer LSTM	0.006978	0.99642	0.007672
	TCN 1 Layer	0.293013	−3.0498	0.258041
	TCN 1 Layer LSTM	0.020493	0.978444	0.018826
	TCN 2 Layer	0.276508	−2.41737	0.237039
	TCN 2 Layer LSTM	0.02398	0.973373	0.020924
	TCN 3 Layer	0.069998	0.76446	0.062231
	TCN 3 Layer LSTM	0.035089	0.947529	0.029372
	CNN 1 Layer biLSTM	0.065165	0.768908	0.06164
	TCN 1 Layer biLSTM	0.027531	0.966455	0.023485
	CNN 2 Layer biLSTM	0.033789	0.935189	0.032644
	TCN 2 Layer biLSTM	0.044124	0.903493	0.039834
	CNN 3 Layer biLSTM	0.006372	0.997612	0.006266
	TCN 3 Layer biLSTM	0.022756	0.969399	0.022431
Cell 4	CNN 1 Layer	0.119367	0.443262	0.108628
	CNN 1 Layer LSTM	0.026017	0.977727	0.021728
	CNN 2 Layer	0.125915	0.556123	0.096995
	CNN 2 Layer LSTM	0.005558	0.99887	0.004894
	CNN 3 Layer	0.075763	0.851374	0.056126
	CNN 3 Layer LSTM	0.005821	0.998702	0.005245
	TCN 1 Layer	0.258304	−1.22427	0.217125
	TCN 1 Layer LSTM	0.018365	0.988164	0.015839
	TCN 2 Layer	0.260731	−1.0556	0.20873
	TCN 2 Layer LSTM	0.019524	0.988068	0.015903
	TCN 3 Layer	0.074752	0.822347	0.061362
	TCN 3 Layer LSTM	0.049175	0.939615	0.035775
	CNN 1 Layer biLSTM	0.119229	0.615059	0.090326
	TCN 1 Layer biLSTM	0.040444	0.956277	0.030442
	CNN 2 Layer biLSTM	0.041402	0.949991	0.032557
	TCN 2 Layer biLSTM	0.074103	0.85012	0.056362
	CNN 3 Layer biLSTM	0.004349	0.999315	0.003811
	TCN 3 Layer biLSTM	0.043671	0.946169	0.033778
Cell 5	CNN 1 Layer	0.129032	0.356202	0.113182
	CNN 1 Layer LSTM	0.027533	0.972405	0.023432

Table 6 (continued)

Cell	Network	MAPE	R ²	RMSE
Cell 6	CNN 2 Layer	0.132312	0.486774	0.101055
	CNN 2 Layer LSTM	0.008905	0.995507	0.009455
	CNN 3 Layer	0.076566	0.844913	0.055551
	CNN 3 Layer LSTM	0.011323	0.99425	0.010696
	TCN 1 Layer	0.258766	−1.33665	0.215625
	TCN 1 Layer LSTM	0.024172	0.977128	0.021333
	TCN 2 Layer	0.265304	−1.25455	0.211803
	TCN 2 Layer LSTM	0.027834	0.972988	0.023184
	TCN 3 Layer	0.079565	0.789357	0.06474
	TCN 3 Layer LSTM	0.048161	0.938006	0.035122
	CNN 1 Layer biLSTM	0.134827	0.509723	0.098769
	TCN 1 Layer biLSTM	0.042313	0.951023	0.031217
	CNN 2 Layer biLSTM	0.043081	0.941963	0.033982
	TCN 2 Layer biLSTM	0.077427	0.832873	0.057667
	CNN 3 Layer biLSTM	0.011154	0.992247	0.012421
	TCN 3 Layer biLSTM	0.046067	0.939991	0.034555
	CNN 1 Layer	0.142815	−0.45007	0.123254
	CNN 1 Layer LSTM	0.02055	0.967362	0.018491
	CNN 2 Layer	0.087764	0.337815	0.08329
Cell 7	CNN 2 Layer LSTM	0.004529	0.998208	0.004333
	CNN 3 Layer	0.047919	0.83952	0.041003
	CNN 3 Layer LSTM	0.003058	0.998966	0.003291
	TCN 1 Layer	0.294378	−4.7412	0.245249
	TCN 1 Layer LSTM	0.011609	0.988613	0.010922
	TCN 2 Layer	0.284836	−4.17809	0.232911
	TCN 2 Layer LSTM	0.017274	0.979297	0.014727
	TCN 3 Layer	0.053961	0.743158	0.051873
	TCN 3 Layer LSTM	0.032342	0.929225	0.02723
	CNN 1 Layer biLSTM	0.0649	0.657898	0.059866
	TCN 1 Layer biLSTM	0.022139	0.964624	0.019251
	CNN 2 Layer biLSTM	0.022031	0.955142	0.021678
	TCN 2 Layer biLSTM	0.036465	0.888791	0.034133
	CNN 3 Layer biLSTM	0.004083	0.998341	0.004169
	TCN 3 Layer biLSTM	0.021834	0.959042	0.020715
	CNN 1 Layer	0.143152	0.727939	0.105557
	CNN 1 Layer LSTM	0.02248	0.995002	0.014308
	CNN 2 Layer	0.234344	0.577292	0.131575
	CNN 2 Layer LSTM	0.006489	0.999418	0.004884
Cell 8	CNN 3 Layer	0.136293	0.861527	0.075307
	CNN 3 Layer LSTM	0.005166	0.999646	0.003807
	TCN 1 Layer	0.276393	−0.05102	0.207472
	TCN 1 Layer LSTM	0.019171	0.994845	0.014531
	TCN 2 Layer	0.243787	0.059884	0.19622
	TCN 2 Layer LSTM	0.057952	0.969096	0.035577
	TCN 3 Layer	0.112763	0.889374	0.06731
	TCN 3 Layer LSTM	0.057426	0.970504	0.034757
	CNN 1 Layer biLSTM	0.238318	0.559252	0.134353
	TCN 1 Layer biLSTM	0.06675	0.965135	0.037788
	CNN 2 Layer biLSTM	0.071527	0.958236	0.041358
	TCN 2 Layer biLSTM	0.12446	0.878092	0.070659
	CNN 3 Layer biLSTM	0.007418	0.999419	0.004877
	TCN 3 Layer biLSTM	0.080104	0.949047	0.045681

0.011. While the lowest R² (−0.689) value was obtained with TCN 2-layer model and RMSE & MAPE values were highest.

5.4. Feature analysis

In the previous section, CNN 3-layer LSTM was found to be the best model for estimating the state of health of lithium-ion cells. For understanding the effect of features on the model prediction accuracy. A feature analysis was carried out using CNN 3-layer LSTM architecture. Fig. 22(a–g) shows the comparison of the values predicted by the model and the actual value when in Fig. 22a all the features were used, in Fig. 22b except charging voltage all the features were considered. Similarly, in Fig. 22c discharging voltage was not considered, in Fig. 22d charging time was not considered, in Fig. 22e discharging time was not considered, in Fig. 22f the parameter, voltage at which peak of IC curve occurs was not considered and lastly in Fig. 22g the peak of IC curve was not considered. Complete dataset was used for the comparison. The figures show that the predicted values are similar to the actual ones, except for Fig. 22e and (g) wherein, the model was trained without using

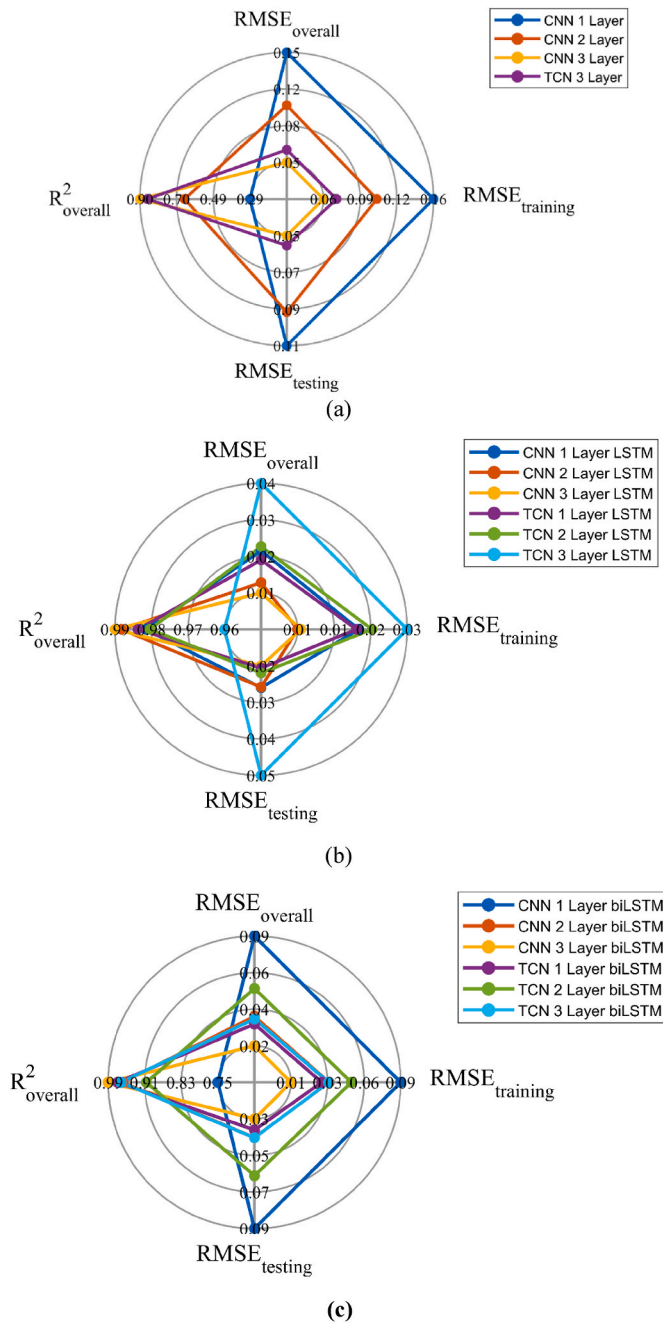


Fig. 20. Overall performance of (a) CNN/TCN; (b) CNN/TCN with LSTM and (c) CNN/TCN with biLSTM.

discharge time and peak of discharge IC curve as a feature. Fig. 23 also shows that the absolute percentage error was highest with discharging time. Moreover, the number of outliers was also maximum. As seen in Fig. 12, discharge time and peak of the IC curve had the maximum correlation with SOH. Thus, the accuracy of the model was affected when the features were not considered. Further, it is seen that without the feature discharge time the prediction accuracy of the model below an SOH of 0.7 is low. Generally, the lithium-ion cells are not used when their SOH reduces to 0.8.

Charging time had a moderate correlation with SOH. Thus, its effect on the accuracy of the model was also minimum. Fig. 24 compares the RMSE, MAPE and R^2 value of the models. The maximum R^2 value of 0.993 was achieved when all the parameters were considered. While the variation between RMSE, MAPE and R^2 for model with all the features

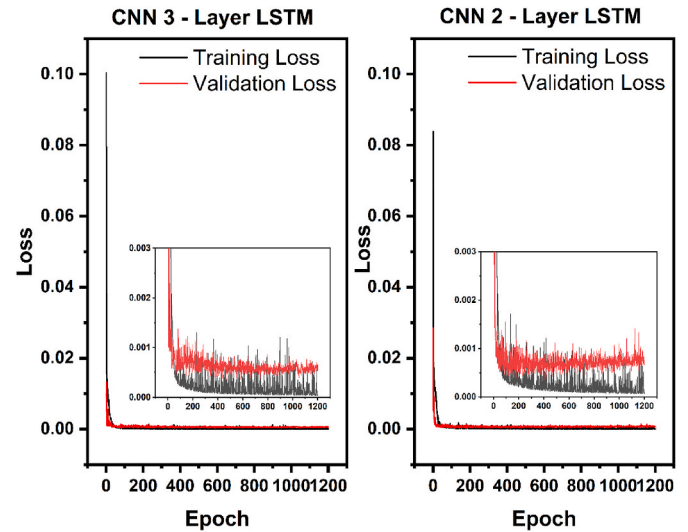


Fig. 21. Loss for CNN 3- and 2-layer with LSTM

and model without discharging voltage or charging voltage was similar. Thus, a model with either charging or discharging voltage can be used with negligible losses in accuracy. The other parameters as seen in Fig. 24 also had an effect on the statistical parameters. The lowest R^2 was observed with peak of discharge IC curve followed by discharging time which corroborates with Fig. 22.

6. Comparison with other published studies

The method proposed in this study was compared with the methods available in the literature on the basis of RMSE and MAPE values. The models used for benchmarking were given by Chen et al. [38], Li et al. [4], Lin et al. [44], Li et al. [39], and Deng et al. [69]. The models have used varied methods for predicting the state of health of lithium-ion batteries. The chemistry of the cells considered in these models was also different. The benchmark models are discussed below.

1) Benchmark 1

Chen et al. [38] considered NCA and NCM cathode materials for evaluating a CNN model. A transfer learning-based CNN model was also used by the authors for predicting the SOH. Partial current and differential current was used as input to the model. The best accuracy of the study is mentioned in Table 7.

2) Benchmark 2

NASA and Oxford University dataset was used for predicting the state of health by Li et al. [4]. Both the databases were separately used for training the model. The authors used CNN, TCN and LSTM based models. Voltage, current, temperature, capacity and time were considered as input to the model. The hyperparameters were optimized by the authors. The best accuracy of the study is tabulated in Table 7.

3) Benchmark 3

In benchmark 3, differential thermal capacity parameter based on current, voltage and temperature was used as a feature for estimating the state of health of lithium-ion battery used by the researchers of Oxford [44]. A fusion model consisting of simulated annealing algorithm and support vector regression was used for the prediction. Table 7, shows the best accuracy of the study.

4) Benchmark 4

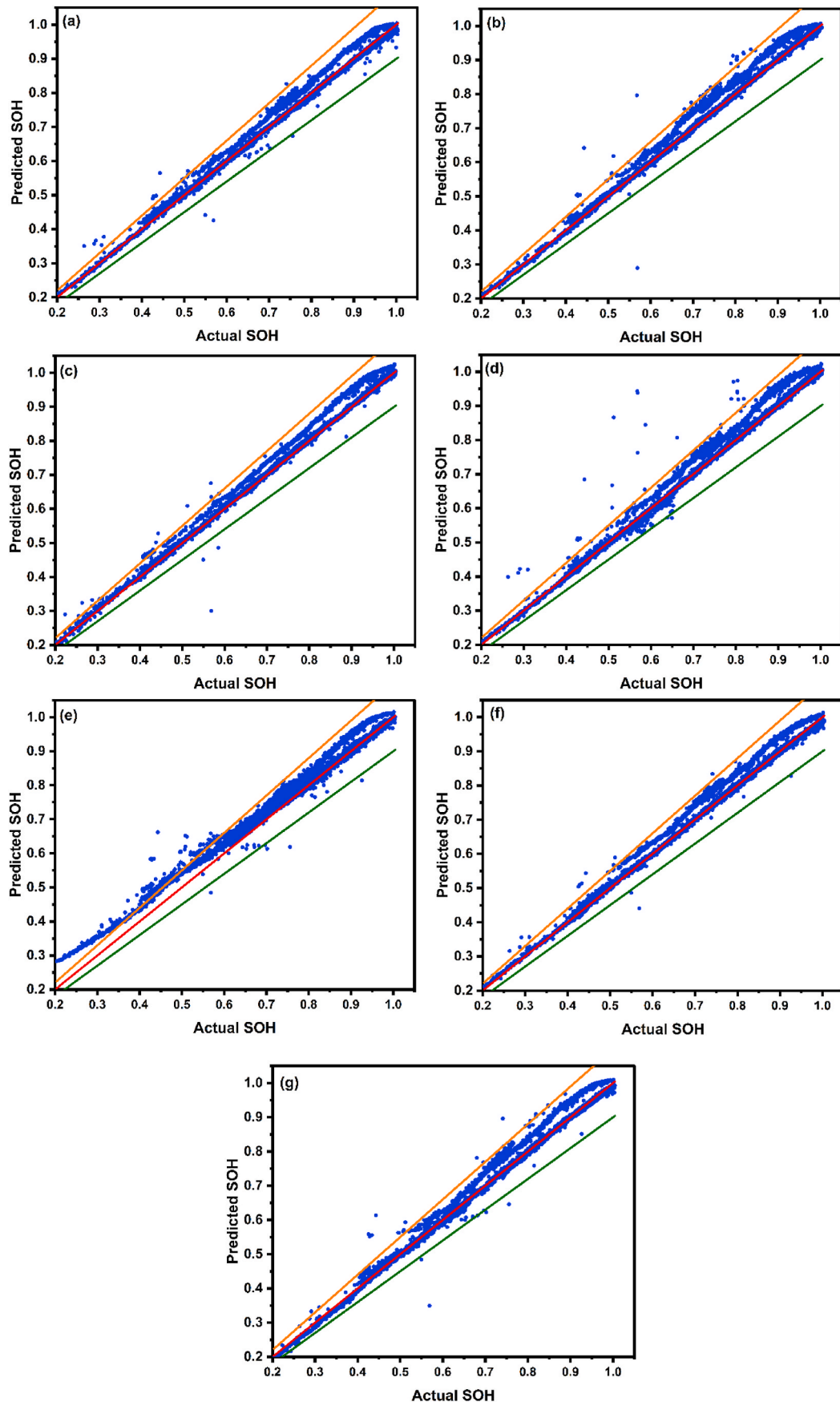


Fig. 22. Variation in predicted SOH for (a) all the parameters (b) without charging voltage (c) without discharging voltage (d) without charging time (e) without discharging time (f) without voltage at which peak ICA occurs and (g) without peak value of ICA.

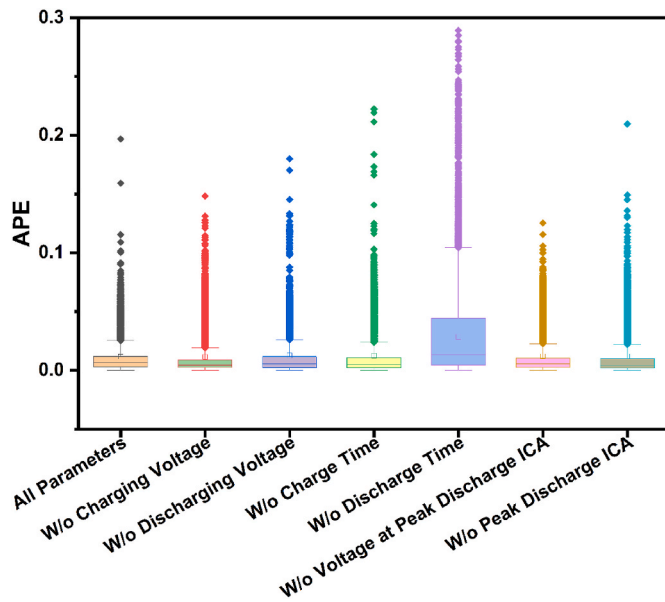


Fig. 23. Box plot of absolute percentage error with various features.

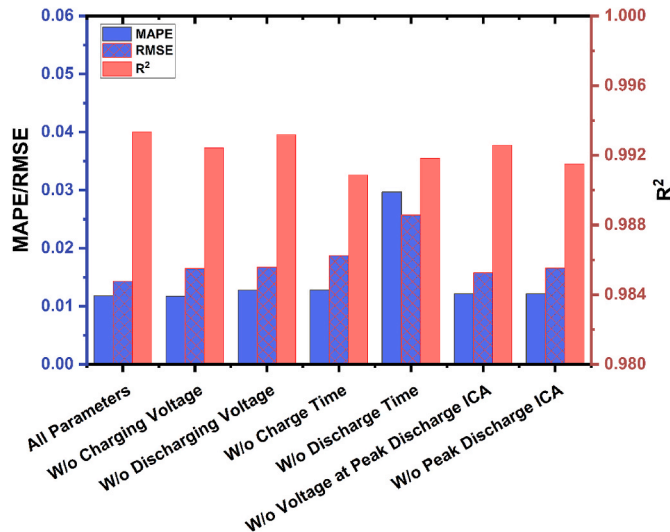


Fig. 24. Statistical comparison with various features.

Table 7

Comparison with benchmarks in terms of prediction accuracy.

	RMSE (%)	MAPE (%)
Proposed model	0.014	0.01
Benchmark 1	2.25	1.85
Benchmark 2	0.11	0.07
Benchmark 3	0.66	0.55
Benchmark 4	0.026 (MSE)	0.128 (MAE)
Benchmark 5	1.37	1.21

Li et al. [39] trained a deep neural network with electrochemical impedance spectroscopy database created at ambient temperatures of 25, 35, and 45 °C. The authors used 12 Eunicell LR2032 lithium-ion coin cells for the preparing the dataset. The best results shown in Table 7 were achieved when the deep learning and transfer learning was combined for 25, and 35 °C datasets.

5) Benchmark 5

Deng et al. [69] used the MIT dataset for predicting the state of health of LFP batteries. The authors used four features derived from the discharge capacity of the cell. The authors compared the accuracy of SVM, RVM, GPR and LSTM models. The best results were observed with LSTM model and the values are tabulated in Table 7.

Comparison between the proposed model and the benchmark model shows that the proposed model outperforms all the benchmark models by a large margin. Compared with benchmark 2, the RMSE of the proposed model was 97 % lower while MAPE was 85.7 % lower. The improvement in performance can be attributed to the combination of the use of scalar data as well as vector data of discharge and charge voltage. The combination of CNN and LSTM can well extract the features from the given parameters resulting in improved performance of the model. The other advantage of this method is that the SOH can be predicted from the first cycle itself.

7. Advantages and limitations

The method proposed in this study utilizes only voltage and current data of a cycle for accurate prediction of state of health of a lithium-ion cell. The derived incremental capacity curve used as input to the model succinctly reflects the electrochemical ageing of the cell. Most of the battery management systems used in various applications can provide only voltage and current data with respect to time. Thus, the major advantage of the proposed method is, it can be used in industrial applications. Secondly, the deep learning model was trained without the need of an expert in the field. Still very high accuracies were achieved with the error being less than 2 %.

Although the proposed method has certain advantages, it does have disadvantages. Firstly, only one dataset was utilized for training the model. Further, work has to be carried out to evaluate its generalization ability, i.e., utilizing cells of different chemistries used in various applications. Also, the dataset of the cells charged and discharged at various rates and the experiments carried out at different ambient temperatures can be used for generalization of the model. Secondly, in real-time applications, it is not necessary that the cell is completely charged and discharged in a cycle. It may so happen that the cell is partially charged or partially discharged. In such scenarios the prediction accuracy of the model may be low. Finally, such models cannot be used on low cost and power devices, thus hindering their usage in real-time applications.

8. Conclusion

In this study deep learning algorithms namely CNN, TCN, LSTM and Bi-LSTM and their combinations were compared for predicting the state of health of the prismatic cell dataset provided by the University of Maryland. A parallel hybrid network was developed for extracting the associations between time-dependent vector and scalar battery features and the cell state of health. To understand the correlation among the features Spearman correlation coefficient was used. The Spearman correlation coefficient revealed that all the parameters had positive correlation with the state of health of a lithium-ion cell. The features also had a high correlation amongst each other as well. The next objective was to find the architecture that best predicts the state of health. In this regard, the number of CNN and TCN layers were varied along with LSTM and Bi-LSTM. Further, the hyperparameters such as convolutional kernel, stride, filter, number of units in LSTM and dense layers and dropout rate after CNN/TCN layer and LSTM were optimized. Bayesian optimization algorithm was used for the optimization. The accuracy of the models in predicting SOH with multiple layers of CNN and TCN without LSTM and Bi-LSTM was very low. The lowest R^2 value of 0.292 was observed with TCN 1-layer model and the maximum R^2 value of 0.9 was observed with TCN 3-layer model. Combining CNN and TCN with bi-directional LSTM further improved the prediction accuracy. The lowest R^2 value of 0.75 was observed with CNN 1-layer model while the

highest value of 0.99 was observed with CNN 3-layer model. When LSTM was combined with CNN and TCN, the overall R^2 value for the models varied from 0.982 to 0.9996. Overall, CNN and TCN model with LSTM and biLSTM resulted in good performance for predicting the cell state of health. However, the accuracy of the CNN 3-layer LSTM model in predicting the SOH of unknown dataset was found to be the best. Thus, the CNN 3-layer LSTM was selected as the best model.

To deduce the impact of features on the model prediction accuracy a feature sensitivity analysis was carried out. CNN 3-layer LSTM model was chosen for the analysis. The model was retrained with removing one feature at a time. The accuracy of the model was affected when either discharging current or the peak of the discharge IC curve feature was removed. Since, both the features have high correlation with SOH, the accuracy was affected. Moreover, the accuracy of the model without the vectors charging voltage or discharging voltage was slightly affected. Thus, either of the two vectors can be used as an input to the model. Lastly, the present model was compared with benchmark models present in the literature. The authors found that the present model had the best accuracy in terms of RMSE and MAPE. As compared to the benchmark (Benchmark 2) with lowest values, the present model had a 97 % lower RMSE and 87.5 % lower MAPE.

The potential areas for future work are 1) generalization of the proposed model using transfer learning such that SOH prediction for any type of cell cycled under different ambient conditions can be made; 2) cases where the cell is partially charged or discharged will be explored with the proposed method; and 3) feasibility of using low-cost and power devices for implementing the proposed method in real-time applications.

CRedit authorship contribution statement

Ankit Sonthalia: Writing – original draft, Conceptualization. **J.S. Femilda Josephin:** Visualization, Validation. **Edwin Geo Varuvel:** Writing – review & editing, Supervision. **Arunachalam Chinnathambi:** Validation, Formal analysis. **Thiyagarajan Subramanian:** Writing – original draft, Data curation. **Farzad Kiani:** Visualization, Project administration.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

References

- [1] Yao J, Han T. Data-driven lithium-ion batteries capacity estimation based on deep transfer learning using partial segment of charging/discharging data. *Energy* 2023; 271:127033.
- [2] Catsaros O. Lithium-ion battery pack prices hit record low of \$139/kWh. BloombergNEF 2023. <https://about.bnef.com/blog/lithium-ion-battery-pack-price-s-hit-record-low-of-139-kwh/>. [Accessed 20 November 2024].
- [3] Gül T, Pales AF, Connelly E. Global EV outlook 2024 moving towards increased affordability. 2024.
- [4] Li C, Han X, Zhang Q, Li M, Rao Z, Liao W, et al. State-of-health and remaining-useful-life estimations of lithium-ion battery based on temporal convolutional network-long short-term memory. *J Energy Storage* 2023;74:109498.
- [5] Rauf H, Khalid M, Arshad N. Machine learning in state of health and remaining useful life estimation: theoretical and technological development in battery degradation modelling. *Renew Sustain Energy Rev* 2022;156:111903.
- [6] Liu Y, Zuo MJ, Li Y-F, Huang H-Z. Dynamic reliability assessment for multi-state systems utilizing system-level inspection data. *IEEE Trans Reliab* 2015;64: 1287–99.
- [7] Xiong R, Zhang Y, Wang J, He H, Peng S, Pecht M. Lithium-ion battery health prognosis based on a real battery management system used in electric vehicles. *IEEE Trans Veh Technol* 2019;68:4110–21.
- [8] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat Energy* 2019;4: 383–91.
- [9] Lin M, Wu D, Meng J, Wang W, Wu J. Health prognosis for lithium-ion battery with multi-feature optimization. *Energy* 2023;264:126307.
- [10] Tian J, Xiong R, Shen W, Lu J, Sun F. Flexible battery state of health and state of charge estimation using partial charging data and deep learning. *Energy Storage Mater* 2022;51:372–81.
- [11] Zheng Y, Qin C, Lai X, Han X, Xie Y. A novel capacity estimation method for lithium-ion batteries using fusion estimation of charging curve sections and discrete Arrhenius aging model. *Appl Energy* 2019;251:113327.
- [12] Xiong R, Li L, Li Z, Yu Q, Mu H. An electrochemical model based degradation state identification method of Lithium-ion battery for all-climate electric vehicles application. *Appl Energy* 2018;219:264–75.
- [13] Yu Q, Xiong R, Yang R, Pecht MG. Online capacity estimation for lithium-ion batteries through joint estimation method. *Appl Energy* 2019;255:113817.
- [14] Li J, Adewuyi K, Lotfi N, Landers RG, Park J. A single particle model with chemical/mechanical degradation physics for lithium ion battery State of Health (SOH) estimation. *Appl Energy* 2018;212:1178–90.
- [15] Zhang D, Dey S, Perez HE, Moura SJ. Real-time capacity estimation of lithium-ion batteries utilizing thermal dynamics. *IEEE Trans Control Syst Technol* 2020;28: 992–1000.
- [16] Wang Y, Zhao G. A comparative study of fractional-order models for lithium-ion batteries using Runge Kutta optimizer and electrochemical impedance spectroscopy. *Control Eng Pract* 2023;133:105451.
- [17] He X, Sun B, Zhang W, Fan X, Su X, Ruan H. Multi-time scale variable-order equivalent circuit model for virtual battery considering initial polarization condition of lithium-ion battery. *Energy* 2022;244:123084.
- [18] Xu Z, Wang J, Lund PD, Zhang Y. Co-estimating the state of charge and health of lithium batteries through combining a minimalist electrochemical model and an equivalent circuit model. *Energy* 2022;240:122815.
- [19] Wang Y, Zhou C, Chen Z. Optimization of battery charging strategy based on nonlinear model predictive control. *Energy* 2022;241:122877.
- [20] Shao J, Li J, Yuan W, Dai C, Wang Z, Zhao M, et al. A novel method of discharge capacity prediction based on simplified electrochemical model-aging mechanism for lithium-ion batteries. *J Energy Storage* 2023;61:106788.
- [21] Shen S, Sadoughi M, Li M, Wang Z, Hu C. Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries. *Appl Energy* 2020;260:114296.
- [22] Walker E, Rayman S, White RE. Comparison of a particle filter and other state estimation methods for prognostics of lithium-ion batteries. *J Power Sources* 2015; 287:1–12.
- [23] Xu X, Chen N. A state-space-based prognostics model for lithium-ion battery degradation. *Reliab Eng Syst Saf* 2017;159:47–57.
- [24] Chang Y, Fang H, Zhang Y. A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery. *Appl Energy* 2017;206:1564–78.
- [25] Cong X, Zhang C, Jiang J, Zhang W, Jiang Y, Jia X. An improved unscented particle filter method for remaining useful life prognostic of lithium-ion batteries with Li (NiMnCo)O₂ cathode with capacity diving. *IEEE Access* 2020;8:58717–29.
- [26] Khelif R, Chebel-Morello B, Malinowski S, Laajili E, Fnaiech F, Zerhouni N. Direct remaining useful life estimation based on support vector regression. *IEEE Trans Ind Electron* 2017;64:2276–85.
- [27] Shu X, Li G, Shen J, Lei Z, Chen Z, Liu Y. A uniform estimation framework for state of health of lithium-ion batteries considering feature extraction and parameters optimization. *Energy* 2020;204:117957.
- [28] Lin M, Zeng X, Wu J. State of health estimation of lithium-ion battery based on an adaptive tunable hybrid radial basis function network. *J Power Sources* 2021;504: 230063.
- [29] Luo W, Syed AU, Nicholls JR, Gray S. An SVM-based health classifier for offline Li-ion batteries by using EIS technology. *J Electrochem Soc* 2023;170:030532.
- [30] Wang C, Su Y, Ye J, Xu P, Xu E, Ouyang T. Enhanced state-of-charge and state-of-health estimation of lithium-ion battery incorporating machine learning and swarm intelligence algorithm. *J Energy Storage* 2024;83:110755.
- [31] Wang S, Jin S, Deng D, Fernandez C. A critical review of online battery remaining useful lifetime prediction methods. *Front Mech Eng* 2021;7.
- [32] Li R, Xu S, Li S, Zhou Y, Zhou K, Liu X, et al. State of charge prediction algorithm of lithium-ion battery based on PSO-svr cross validation. *IEEE Access* 2020;8: 10234–42.
- [33] Nathans LL, Oswald FL, Nimón K. Interpreting multiple linear regression: a guidebook of variable importance. *Pract Assessment, Res Eval* 2012;17:9.
- [34] Wang Z, Yuan C, Li X. Lithium battery state-of-health estimation via differential thermal voltammetry with Gaussian process regression. *IEEE Trans Transp Electrif* 2021;7:16–25.
- [35] Dai H, Wang J, Huang Y, Lai Y, Zhu L. Lightweight state-of-health estimation of lithium-ion batteries based on statistical feature optimization. *Renew Energy* 2024; 222:119907.
- [36] Liu G, Zhang X, Liu Z. State of health estimation of power batteries based on multi-feature fusion models using stacking algorithm. *Energy* 2022;259:124851.
- [37] Qian C, Xu B, Chang L, Sun B, Feng Q, Yang D, et al. Convolutional neural network based capacity estimation using random segments of the charging curves for lithium-ion batteries. *Energy* 2021;227:120333.
- [38] Chen S-Z, Liang Z, Yuan H, Yang L, Xu F, Fan Y. A novel state of health estimation method for lithium-ion batteries based on constant-voltage charging partial data and convolutional neural network. *Energy* 2023;283:129103.
- [39] Li Y, Maleki M, Banitaan S. State of health estimation of lithium-ion batteries using EIS measurement and transfer learning. *J Energy Storage* 2023;73:109185.

- [40] Ren L, Dong J, Wang X, Meng Z, Zhao L, Deen MJ. A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life. *IEEE Trans Ind Informatics* 2021;17:3478–87.
- [41] An Q, Tao Z, Xu X, El Mansori M, Chen M. A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. *Measurement* 2020;154:107461.
- [42] Ma M, Mao Z. Deep-convolution-based LSTM network for remaining useful life prediction. *IEEE Trans Ind Informatics* 2021;17:1658–67.
- [43] Chen D, Zhang W, Zhang C, Sun B, Cong XW, Wei S, et al. A novel deep learning-based life prediction method for lithium-ion batteries with strong generalization capability under multiple cycle profiles. *Appl Energy* 2022;327:120114.
- [44] Lin M, Yan C, Meng J, Wang W, Wu J. Lithium-ion batteries health prognosis via differential thermal capacity with simulated annealing and support vector regression. *Energy* 2022;250:123829.
- [45] Hong J, Li K, Liang F, Yang H, Zhang C, Yang Q, et al. A novel state of health prediction method for battery system in real-world vehicles based on gated recurrent unit neural networks. *Energy* 2024;289:129918.
- [46] Wang X, Hu B, Su X, Xu L, Zhu D. State of health estimation for lithium-ion batteries using random forest and gated recurrent unit. *J Energy Storage* 2024;76:109796.
- [47] Cheng G, Wang X, He Y. Remaining useful life and state of health prediction for lithium batteries based on empirical mode decomposition and a long and short memory neural network. *Energy* 2021;232:121022.
- [48] Obregon J, Han Y-R, Ho CW, Mouraliraman D, Lee CW, Jung J-Y. Convolutional autoencoder-based SOH estimation of lithium-ion batteries using electrochemical impedance spectroscopy. *J Energy Storage* 2023;60:106680.
- [49] Tong C, Zhang L, Li H, Ding Y. Temporal inception convolutional network based on multi-head attention for ultra-short-term load forecasting. *IET Gener Transm Distrib* 2022;16:1680–96.
- [50] Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv* 2018.
- [51] Yin Y, Lin J, Sun N, Zhu Q, Zhang S, Zhang Y, et al. Method for detection of unsafe actions in power field based on edge computing architecture. *J Cloud Comput* 2021;10:17.
- [52] He W, Williard N, Osterman M, Pecht M. Prognostics of lithium-ion batteries based on Dempster-Shafer theory and the Bayesian Monte Carlo method. *J Power Sources* 2011;196:10314–21.
- [53] Lin M, Wu D, Meng J, Wu J, Wu H. A multi-feature-based multi-model fusion method for state of health estimation of lithium-ion batteries, vol. 518; 2022.
- [54] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern Recognit* 2018;77:354–77.
- [55] Lea C, Flynn MD, Vidal R, Reiter A, Hager GD. Temporal convolutional networks for action segmentation and detection. 2017 IEEE conf. Comput. Vis. Pattern recognit.. 2017. p. 1003–12.
- [56] Hewage P, Behera A, Trovati M, Pereira E, Ghahremani M, Palmieri F, et al. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Comput* 2020;24:16453–82.
- [57] Goay CH, Ahmad NS, Goh P. Temporal convolutional networks for transient simulation of high-speed channels. *Alexandria Eng J* 2023;74:643–63.
- [58] Lara-Benítez P, Carranza-García M, Luna-Romera JM, Riquelme JC. Temporal convolutional networks applied to energy-related time series forecasting. *Appl Sci* 2020;10:1–17.
- [59] Oord A van den, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, et al. WaveNet: a generative model for raw audio. *ArXiv* 2016.
- [60] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE conf. Comput. Vis. Pattern recognit.; 2016. p. 770–8.
- [61] Abdi M, Nahavandi S. Multi-residual networks: improving the speed and accuracy of residual networks. *ArXiv* 2016.
- [62] Hahn T Von, Mechefske CK. Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder. *Int J Hydromechatronics* 2021;4:69–98.
- [63] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. 2012.
- [64] Shao Y, Tang J, Liu J, Han L, Dong S. Multivariable system prediction based on TCN-LSTM networks with self-attention mechanism and LASSO variable selection. *ACS Omega* 2023;8:47798–811.
- [65] Yildirim Ö. A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. *Comput Biol Med* 2018;96:189–202.
- [66] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Network* 2005;18:602–10.
- [67] Kulshrestha A, Krishnaswamy V, Sharma M. Bayesian BiLSTM approach for tourism demand forecasting. *Ann Tour Res* 2020;83:102925.
- [68] Wu J, Chen X-Y, Zhang H, Xiong L-D, Lei H, Deng S-H. Hyperparameter optimization for machine learning models based on bayesian optimizationb. *J Electron Sci Technol* 2019;17:26–40.
- [69] Deng Z, Lin X, Cai J, Hu X. Battery health estimation with degradation pattern recognition and transfer learning. *J Power Sources* 2022;525:231027.