Energy 300 (2024) 131540

Contents lists available at ScienceDirect

Energy

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

Deep machine learning approaches for battery health monitoring

S. Singh, P.R. Budarapu *

School of Mechanical Sciences, Indian Institute of Technology Bhubaneswar, Bhubaneswar 752050, India

ARTICLE INFO

Keywords:
Battery management system
Deep machine learning
Predictive analytics
Time series
Forecasting
LSTM
CNN and transformers

ABSTRACT

The performance and longevity of batteries can be measured through battery parameters, like: state of charge, state of health, and remaining useful life. Therefore, accurate estimation of these parameters is essential for developing efficient battery management systems. Recently, a variety of smart model-based and data-driven artificial neural network methods have been evolved for efficient management of battery systems. However, there is space for improving the accuracy and reliability of the proposed methods, particularly in the prediction and forecasting of battery performance metrics. In this study, a deep machine learning based forecasting technique has been proposed to better estimate the battery performance parameters. Available open source data-sets on three distinct battery technologies comprising of battery cycling profiles and environmental conditions, are adopted to construct and validate models for predicting battery health. A common data-set is developed by combining the open source data, which is later supplied as input to the predictive analytics and time series analysis. Encoders and decoders are employed for the extraction of important and relevant feature information from the data-set. Three different network model-based architectures are developed to predict the state of charge of Lithium-ion batteries. The proposed models are observed to accurately capture the dynamic behavior of the battery during discharge, which is confirmed by the analysis of the recorded voltage, current, and temperature data, and the precise estimation of state of charge. Furthermore, a novel forecasting method based on the time-series analysis is also introduced. Mean average error values less than 0.2% are observed when neural basis expansion analysis for interpretable time series architecture in conjunction with encoder–decoder-based feature extraction is employed.

1. Introduction

In the modern world, electric power from batteries fuel a wide variety of equipment ranging from sensitive electronic gadgets to electric vehicles. Due to their high energy density, long-life cycle, and a low self-discharge rate, LiBs have emerged as the best option for energy storage. Recently, a rapid expansion and dramatic transformation in the electric vehicle industry is witnessed due to the development of advanced digitization, artificial intelligence, and ML based techniques [1]. The driving range, charge time, battery lifespan and cost are the key parameters of the electric vehicle customers. The effectiveness of batteries significantly influence the performance of electric vehicles. The performance and longevity of batteries is influenced by a variety of factors, including: temperature, utilization patterns, charge and discharge cycles, to name a few. Whereas, conventional battery life testing methods, including: cycling the battery and employing capacitors and resistors are still practiced to estimate their degradation profile. This is because, either the machine learning/physics-based/mathematical models [2–4] are unable to accurately forecast the remaining useful battery life due to lack of data and/or varying chemical characteristics of batteries. On the other hand, a big database addressing all relevant battery failure mechanisms has been developed as a result of more than a decade of research. The battery informatics, an interdisciplinary field of ML and battery research along with the scarcity of battery data is reviewed in [5,6].

In the realm of managing Lithium-ion batteries [7], numerous strategies have been put forth to gauge the SOC, assess the SOH, and forecast the RUL. These strategies utilize a variety of techniques, ranging from optimization approaches using neural networks to methods that are model-based and data-driven, underscoring the efficacy and potential of machine learning methodologies in the management of battery systems. Popular ML based algorithms for estimating SOC and SOH, including: shallow neural network, deep learning, support vector machine, and Gaussian process regression methods are reviewed in [8,9]. A review of ML based methods to predict the RUL of Lithium-ion batteries is provided in [10]. A discrete element method to simulate particle plastic deformation during electrode compaction of solid-state batteries was proposed [11], to simulate the deformations occur during mold compaction processes associated with solid-state battery electrodes.

* Corresponding author.
E-mail address: pattabhi@iitbbs.ac.in (P.R. Budarapu).

https://doi.org/10.1016/j.energy.2024.131540
Received 16 December 2023; Received in revised form 12 April 2024; Accepted 3 May 2024
Available online 7 May 2024

Nomenclature

Nomenclature of the acronyms used in this study

LiBs	Lithium-ion batteries
BMS	Battery management system
SOC	State of charge
SOH	State of health
RUL	Remaining useful life
NMC	Nickel Manganese Cobalt
LFP	Lithium Iron Phosphate
UDDS	Urban dynamometer driving schedule
HWFET	Highway fuel economy test
ML	Machine learning
DML	Deep machine learning
ANN	Artificial neural network
DNN	Deep neural network
RNN	Recurring neural network
GRNN	Gated recurring neural network
LSTM	Long short-term memory
SELU	Scaled exponential linear unit
RELU	Rectified linear unit
GRU	Gated recurrent unit
CNN	Convolutional neural network
NBEATS	Neural basis expansion analysis for interpretable time series
ARIMA	Autoregressive integrated moving average
TFT	Temporal fusion transformer
ELU	Exponential linear unit
GLU	Gated linear unit
BATS	BATched Sparse
TBATS	Trigonometric seasonality, Box–Cox transformation, ARMA errors, Trend and Seasonal components
MAE	Mean average error
MAPE	Mean average percentage error
RMSE	Root mean square error
IoT	Internet of Things
API	Application programming interface

Nomenclature of the symbols used in this study

RNN	
f_W	Activation function
W	Weight parameters
$\mathbf{x}_t$	Input vector at time step t
$\mathbf{h}_{(t-1)}$, $\mathbf{h}_t$	Internal state at time step $(t-1)$ and t
GRNN	
$\mathbf{h}_t$	The output of the LSTM unit at time step t
$\mathbf{c}_t$	Memory at time step t
$\boldsymbol{\sigma}_t$	Output gate to modulate the amount of memory content exposure
σ	Logistic sigmoid function in LSTM network
W_0, U_0, V_0	Weights used in LSTM network
GRU	
$\mathbf{h}_{(t-1)}$, $\mathbf{h}_t$	The activation at time steps $(t-1)$ and t
$\tilde{h}_t$	Candidate activation at time step t
$\mathbf{z}_t$	Updated gate at time step t
$\mathbf{r}_t$	Set of reset gates at time step t
CNN	
F, K, S, P	Filters, kernel size, stride and padding
X	Input sequence with length N
F	Weight tensor
$\mathbf{b}$	Bias vector

Various model-based and data-driven methods for estimating the SOC in Lithium-ion batteries are comprehensively reviewed in [12]. The important properties of the most common electrode and electrolyte materials along with their recent progress using ML based techniques is summarized in [13]. A comprehensive overview of deep machine learning techniques for modeling and forecasting the dynamics of battery systems, along with a summary of open source Lithium-ion battery test and cycle data-sets is provided in [14]. Whereas, a criteria concerning the encoding of information in the SOH forecasting model inputs, model transferability to other batteries, and the applicability to 2nd life battery applications is reviewed in [15], where the authors also highlighted the limitations of the applicability and comparability of existing models due to different data sets, metrics, output values, and forecast horizons. A neural network-based methodology to forecast the SOH and end of life, based on charge–discharge voltage profiles, with the help of LSTM and GRU neural networks is provided in [16].

A static selection model to choose the best nonlinear predictor, ARIMA model and combination function for a specific database for estimating the battery metrics is provided in [17]. The mechanism and influencing factors of battery degradation are studied with the help of LSTM model in [18]. Some data-driven and neural network based approaches to estimate the SOC in Lithium-ion batteries include: an optimization approach based on neural networks [19], a neural networks based method [20], a data-driven method to forecast the SOC based on historical battery data [21], a nonlinear fractional model [22], a deep learning-based framework [23], an independently recurrent neural network based method [24] to name a few. Furthermore, hybrid techniques by combining neural networks are also proposed to estimate the SOC. For instance, a hybrid multi-layer deep neural network based approach [25].

On the other hand, several methods for estimating the SOH of Lithium-ion batteries include: incremental capacity analysis technique [26] XGB-AKF method [27], a multi-task learning based approach [28], to name a few. In the similar lines, recent techniques to estimate the RUL include: a cubic polynomial degradation model and envelope extraction approach [29], CEEMDAN and WOA-SVR model [30], TCN-GRU-DNN and dual attention mechanism [31], an exponential model and particle filter approach [32], a hybrid architecture combining DNN and NBEATS based method [33], a combined incremental capacity and LSTM based network [34], a forecasting method based on histogram features [35], a CNN model to estimate the future values of SOH of Li-ion batteries in the early phases of qualification tests [36], a prediction method using electrochemical impedance spectroscopy feature [37], multi-health features extraction and an improved LSTM neural network based method [38], a CNN-Transformer framework combined with data pre-processing [39], to name a few. An approach by extracting the features from current, voltage and temperature curves during charge and discharge to predict SOH of a battery with the help of a LSTM neural network is introduced in [40].

An adaptive self-attention LSTM model to predict the RUL of Lithium-ion batteries is available at [41]. A DML network [42] with a sequence decomposition algorithm to decompose the capacity fading sequence into a long-term residual and short-term fluctuations, with the help of Transformer-based network is developed in [43], to estimate the battery capacity as well as RUL. A prediction model for estimation of the SOH and RUL of batteries is introduced in [44], by combining Savitzky–Golay filter with GRU neural networks. An examination of the developments in SOC estimation, particularly on data-driven estimation techniques was provided in [45]. An improved anti-noise adaptive LSTM neural network with feature extraction and parameter characterization was proposed in [46], for RUL prediction. A CNN model to estimate capacity for batteries with different degradation paths, to

estimate the battery capacity using relaxation voltage data collected for 10 s was investigated in [47].

Few recent techniques on time-series analysis are as follows: a convolutional transformer model for multi-step time series forecasting [48], a data-driven model with LSTM network for time-series regression prediction problem, by combining data-driven methods and feature signal analysis [49], an encoder–decoder model framework based on RNNs to replace the time consuming characterization tests [50], to name a few. Moreover, a chained Gaussian process framework for forecasting after integrating the available prior knowledge on derivatives is developed in [51]. A fusion prognostic framework by adopting a data-driven time series prediction model along with the extracted features for Lithium-ion battery capacity prediction, based on an auto-regression with an exogenous-variable model was introduced in [52]. A rolling-horizon optimization model integrated with a RNN driven forecasting designed to interactively forecast uncertainty and optimize battery energy storage operations in an iterative fashion was investigated in [53]. A deep learning approach with multi-head attention and a multi-scale hierarchical learning mechanism based on encoder–decoder architecture to forecast voltage and power series that can help in early detection of internal short circuits is available at [54].

The above methods leverage various techniques, including neural network optimization approaches, model-based and data-driven methods, and multi-task learning, demonstrating the effectiveness and potential of these techniques in battery management systems. However, further research is required to improve the accuracy and reliability of the proposed methods. Therefore, development of new methods that can better handle the challenges associated with battery management is required. Moreover, the prediction and forecasting of the battery performance metrics are estimated using the traditional methods. Also, the transformer based architecture for the prediction of battery performance parameters is not explored to the fullest. As a result, a DML based forecasting technique has been proposed here to better estimate the battery management metrics. Thus, the major novelties of this study are: (i) development of a technique to collect and preprocess the open-source data on batteries for electric vehicle for various charge/discharge rates. (ii) Introduction of better predictive models to estimate battery management metrics, like state of charge, with the help of the open-source data. (iii) Development of time and space-efficient forecasting models to predict the remaining useful life of batteries based on the time-series approach.

The rest of the paper is organized as follows: Section 2 provides the information on source data, data processing and flow diagrams indicating the neural network and forecasting architectures developed in this study. The predictive models like LSTM networks, CNN, transformer, apart from the forecasting models are discussed in Section 3. Analysis of results, and the estimation of MAE and the RMSE values are presented in Section 4. Key results along with the future scope is provided in Section 5.

2. Methodology

The accurate estimation of the state of charge and state of health of batteries is necessary to ensure optimal performance, maximize battery life, apart from avoiding over and under charging, and thereby premature failure. In contrast, estimation of SOH is essential for determining the deterioration of batteries over time and hence, predict their remaining useful life. On the other hand, SOC cannot be directly estimated and must be inferred from observable parameters, like: temperature, current, and voltage. Establishing the nonlinear mapping between the observable variables and SOC is a crucial issue that needs to be addressed. Additionally, the dynamic operating conditions of the battery will bring another challenge to SOC estimation due to unpredictable charging and discharging. Traditional methods of SOC estimation, such as: Coulomb counting and open-circuit voltage methods, comes with drawbacks, like: sensitivity to temperature and variations in the battery chemistry.

Therefore, ML has emerged as a potential tool for predicting the SOC and SOH of batteries. DML based methods are advantageous in terms of reduced computational complexity, enhanced accuracy, and the capacity to manage dynamic operating conditions, over the traditional methods. Accurate estimation of SOH is necessary for predicting the RUL of batteries, identifying degradation mechanisms, and devising optimal battery management systems. Traditional techniques for estimation of SOH, like: impedance spectroscopy [55] and capacity fade analysis [56], are time-consuming and require sophisticated equipment. DML based techniques overcome these limitations by utilizing data from multiple sensors, to construct the models that can accurately predict the SOH of batteries in real-time.

One technique to estimate the RUL of batteries is forecasting [30, 57], which comes with substantial implications for the design and operation of battery systems. Accurate RUL forecasting can aid in the prevention of catastrophic failures, the optimization of battery maintenance schedules, and the reduction of the total cost of battery operation. RUL forecasting utilizes historical data to predict the future degradation trajectory of batteries and estimate their expected duration. In contrast, RUL prediction estimates the time until failure based on current operating conditions. DML based RUL forecasting has several advantages over conventional methods, including: the ability to manage large amounts of data, identify intricate degradation mechanisms, and generate accurate and trustworthy predictions. ML based algorithms use historical sensor data, including voltage, current, temperature, and impedance, to construct models that can predict the SOC and SOH of batteries in real time, as well as the future degradation trajectory of batteries and estimate their RUL.

To summarize, accurate estimation of SOC and SOH, as well as RUL forecasting, are indispensable for ensuring optimal battery performance, preventing premature failures, and maximizing battery life. ML has emerged as a potent instrument for predicting these metrics, using data from multiple sensors to construct models that can accurately predict the SOC and SOH of batteries in real time and forecast the RUL.

2.1. Source data

The battery life and performance data-set comprises of information on commercial Lithium-ion batteries subjected to accelerated aging tests. In this study, the data-set related to two distinct battery technologies, namely: NMC and LFP are used for the analysis. The collected data incorporates the battery cycling profiles and environmental conditions, namely: temperature and humidity. In other words, the data includes measurements of battery voltage, current, and temperature, as well as information about the cycling protocol and initial battery capacity. The battery voltage and current data is used to construct and validate models for predicting battery health and remaining useful life.

In this study, different open-source data-sets shown in Table 1 are used to train and test the proposed neural network based predictive and forecasting models for the battery analysis. Considering the batteries for electric vehicles, the number of charge–discharge cycles must be large, i.e., at least greater than 1500 cycles. Therefore, the open-source LG 18650HG2 Li-ion battery data-set is used to predict the state of charge and state of health. Data from Mendeley is used to create the open-source LG 18650HG2 Lithium-ion battery data-set [58]. A series of tests are conducted on the adopted data-set at six different temperatures, i.e.: 40 °C, 25 °C, 10 °C, 0 °C, −10 °C, and −20 °C. Before each discharge test, the battery is charged at a rate of 1C to 4.2 V with a 50 mA cut-off, and battery temperature of more than 22 °C.

The Oxford degradation data-set [59] is a compilation of measurements by examining the behavior of tiny Lithium-ion pouch cells with aging. The data-set contains information gathered from eight different cells, where each cell was subjected to testing in a thermal chamber at a temperature of 46 °C. The data is segregated into two

Table 1
Data sources and cell types used in the present study.

S. No.	Data	Source	No. of cells	Cell type
1	LG data-set [58]	Mc Master University	1	18 650, 3 Ah, NMC
2	Oxford degradation data-set [59]	Oxford University	8	Pouch, 0.7 Ah
3	NMC time series [60]	Mendeley	18	NMC
4	LFP time series [60]	Mendeley	6	Pouch, 1.5 Ah

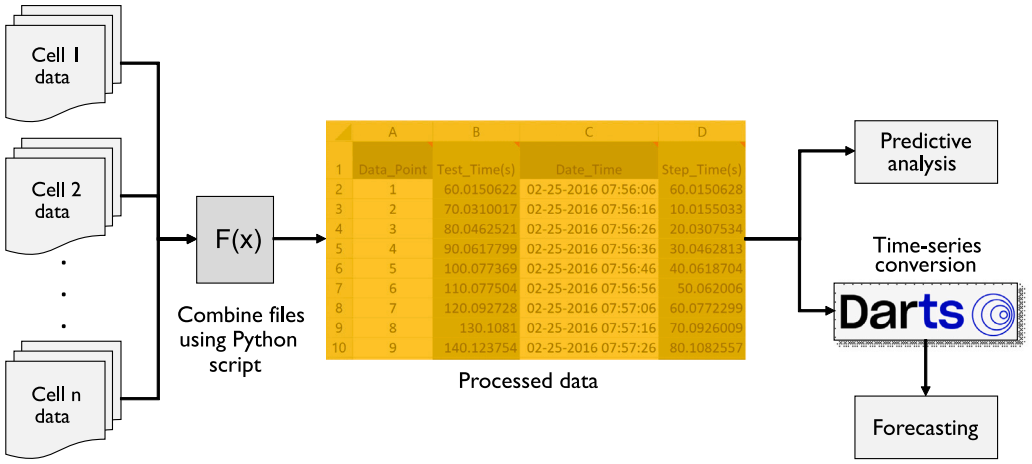

Fig. 1. Schematic indicating the data processing and workflow. A common data-set is created by appropriately combining the battery data obtained from different sources and the processed data is used as the input for predictive analytics and time series analysis.

different files: ExampleDC_C1.mat, containing voltage, current, and temperature measurements recorded during the first charge–discharge cycle only; and Oxford_Battery_Degradation_Dataset_1.mat, containing the results of the characterization tests performed at intervals of 100 cycles until the end of the battery life. The cells are subjected to a particular charge–discharge profile, consisting of a constant-current and constant-voltage charge profile, followed by a drive cycle discharge profile, derived from the urban Artemis profile. Both the above profiles are applied to the cells in a prescribed order and measurements for characterization are obtained at intervals of 100 cycles. The Oxford degeneration data-set is used in the prediction of remaining useful life.

The time series data-sets on NMC and LFP [60] are used for testing, in particular, to check the degradation and estimate the useful life, and hence forecast the battery health. The data-set consists of the experimental cell data for charge–discharge curves. This data-set is used exclusively for testing purpose during the battery health forecasting.

2.2. Data processing

The charge–discharge curves of a battery provide important information on battery health. Therefore, the charge–discharge data is crucial to develop the models for predicting battery performance and health. The performance of a battery is evaluated under variety of environmental conditions, using various test modes, such as: UDDS, HWFET, US06 cycle, to name a few. The measurement data on charge–discharge curves is used to extract features like the battery voltage, current, power, and temperature, apart from SOC and SOH. Predictive analytics uses statistics and modeling techniques to make predictions about future outcomes and hence the performance. Therefore, predictive analytics consider the present and historical data patterns to determine the possibility of those patterns to emerge again. Whereas, time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time. In time series analysis, the data points are recorded at consistent intervals over a set period of time, such that the analysis can indicate the variation of the variables over time. Time series analysis requires a large number of data points to ensure consistency and reliability.

In this study, a common data-set is created by appropriately combining the data obtained from different sources, see Table 1, with the help of the developed Python script, as shown in Fig. 1. The processed data is later supplied as input to the predictive analytics and time series analysis, see Fig. 1. The time series analysis requires splitting of sequential data, which is systematically performed here. Steps to split sequential data are as follows: the sequence length indicates the number of time steps, which depends on the problem and data time resolution. For instance, if a data is recorded at every single minute and the objective is to predict the parameters in the next 10 min, then the sequence length will be equal to 10. Later on, the sequences are created through the overlapping sequences of the sequence length defined in the previous step. Considering a sequence length of 10 and assuming that the data-set has 100 time steps, a total of 100−10+1 equal to 91 sequences are required to be created. This is followed by splitting each of the sequences into input and output pairs. In general, the information corresponding to the time step is stored in the output sequence. In other words, if a sequence has 10 time steps, the input sequence will have the information on first 9 time steps and the output sequence corresponds to the information in the 10th time step. Finally, the input and output pairs are reshaped with the help of LSTM network. The input sequence is reshaped into a three dimensional array, where the dimensions are indicated by samples, time steps, and features. The dimension of samples indicates the number of input/output pairs. Similarly, the dimensions of time steps and features represent the length of each sequence, and the number of variables in each time step, respectively. On the other hand, the output sequence consists of a two dimensional array with samples and features as dimensions.

A good data can be identified based on ‘how well the pre-processing is performed on the data’. This is expressed in terms of missing values, outlier detection, normalization and feature engineering. The above activities are performed for all the data considered in this study to ensure the supply of good data to the developed models. This is ensured by performing some pre-processing steps to check the data quality before using it in model training. Firstly, the missing data points are replaced with averaged interpolated values. This is followed by the application of outlier detection, although extreme outliers are not expected as the considered data sources in this study are reliable. Finally, the data is

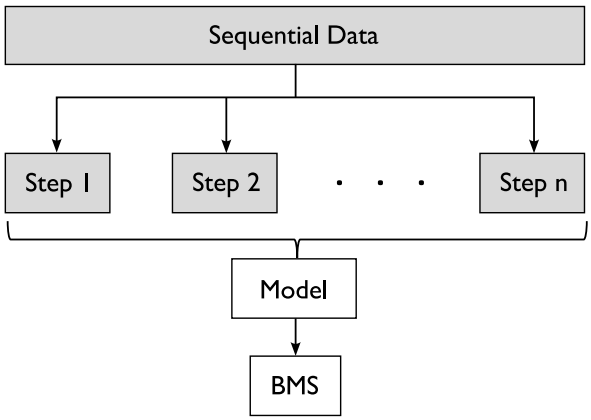

Fig. 2. A schematic highlighting the splitting of sequential data into several steps.

normalized to bring all the features to a comparable scale, enhancing stability in model training and convergence speed.

2.3. Prediction and forecasting workflow

A schematic highlighting the splitting of sequential data into several steps is shown in Fig. 2. The future time steps are later predicted with the help of the developed neural network models, using the present time steps arrived by splitting the sequential data. The number of times steps for a particular data can vary, subjected to the hyper parameters. Thus, variations are required to be tested. In this study, LSTM scheme is employed for splitting the sequential data, considering 300, 500, and 700 time steps.

In time series analysis, a covariate is a variable that is not included in the primary time series data, however, believed to influence the data. Covariates are used to forecast the time series data behavior. Two primary varieties of covariates exists: exogenous and endogenous type. Exogenous variables are external factors which are not affected by the time series data and can influence the data, for example: weather data, economic indicators, to name a few. Endogenous covariates are the internal variables derived from the time series data and can influence the time series data, for example: lagged values of the time series data, moving averages, to name a few. Covariates are used to predict the future values of a cycle through forecasting. Whereas, current, voltage, maximal charge, discharge power, internal impedance, and temperature are used as characteristics in the forecasting of a series.

Covariates enhance the accuracy of time series forecasting models by providing additional information that can be used to predict the behavior of time series data. The historical data can be used to estimate the underlying patterns and trends of time series data, such that the model can predict future values based on the estimated patterns and trends. Therefore, the accuracy of forecasts depends on the accuracy of the historical data and the employed model. Furthermore, the time series forecasting models can be separated into univariate and multivariate categories. Univariate forecasting models considers only the past values of a single variable. In contrast, multivariate forecasting models consider the past values of many variables into forecasts for a single/multiple variables. Architectures that used charge–discharge time steps as covariates for forecasting, when traditional covariates are passed onto the encoder–decoder networks are indicated in Figs. 3(a) and (b), respectively. In Fig. 3(a), the variables are based on the data from charge–discharge cycles only. Whereas, multiple data sources can be accommodated in the encoder–decoder networks shown in Fig. 3(b).

The extraction of important and relevant feature information from a data-set is performed using encoders and decoders, see Fig. 3. Auto-encoders and variational auto-encoders are frequently employed in deep learning models. Encoders are neural networks that compress the input data into a lower-dimensional representation, also known as a bottleneck/latent space, see Fig. 4. A deep machine learning model made of 12 input and output layers each and 21 hidden layers, for dimensionality reduction and features workflow is shown in Fig. 5(a).

The encoder learns to recognize and extract the essential characteristics of the data by discarding any superfluous information, see Fig. 5(b). The compressed representation is then passed on to the decoder as input. A decoder is a neural network that reconstructs the original data from the compressed representation. The decoder is taught to map the compressed representation back to the original data, thereby effectively 'decoding' the compressed information, see Fig. 5(b). Reconstruction error refers to the difference between the original data and the reconstructed data.

Encoders and decoders can be used in tandem to discover a more efficient and compact data representation during feature extraction. The compressed representation can be used as the input for other tasks, such as: classification/clustering, as it contains the essential characteristics of the data. Auto-encoders and variational auto-encoders are utilized for dimensionality reduction and anomaly detection as well.

2.4. Model parameters

In order to prevent the over-fitting, an 'Early Stopping' callback function is employed here during the training process. This function monitors the validation loss during training, particularly the RMSE values. Recently, MAE and RMSE values are used to ensure the convergence. Therefore, MAE and RMSE values are adopted in this study to test the convergence.

If the validation loss is observed to not to improve for a defined number of epochs, known as 'patience', which in this study is set to 50, the training is stopped early to prevent overfitting. Additionally, the dropout regularization and L^2 normalization techniques are utilized to enhance the data accuracy. In the dropout technique a fraction of neurons during training are randomly excluded to prevent over reliance of the model on some specific features. Furthermore, the L^2 normalization is adopted to add a penalty to the loss function based on the squared magnitude of the model weights, which helps to prevent overfitting by discouraging large parameter values.

The robustness of the developed neural network models is tested by dividing the data set is divided into three different subsets: training, validation, and testing sets. This division allows for effective evaluation of the model performance. A popular data split ratio used in this study is 70:15:15, where 70% of the data is allocated for training, 15% for validation, and 15% for testing. This helps to study the model learning patterns and hence generalize well from the majority of the data during training.

3. Predictive models to estimate the output

In this section, RNNs; GRNNs, consisting of: LSTM and GRU; Transformer network; CNNs; forecasting models for degradation, containing: NBEATS model, ARIMA model and TFT are discussed.

3.1. Recurring neural network

A RNN model is an extension of a conventional feed-forward neural network, which is particularly designed to deal with sequential data of variable length sequences. The long-term dependencies continue to exist in the process and are useful for the evaluation at a later stage. RNN applies to a recurrence relation at every time step to process a sequence of data by introducing an internal loop to allow information to be passed from one step of the network to the next. Thus, a recurrent hidden state exists in RNNs, whose activation at each time step depends on the previous time step. Therefore, the recurrence relation can be expressed as:

$$\mathbf{h}_t = f_W(\mathbf{h}_{t-1}, \mathbf{x}_t), \tag{1}$$

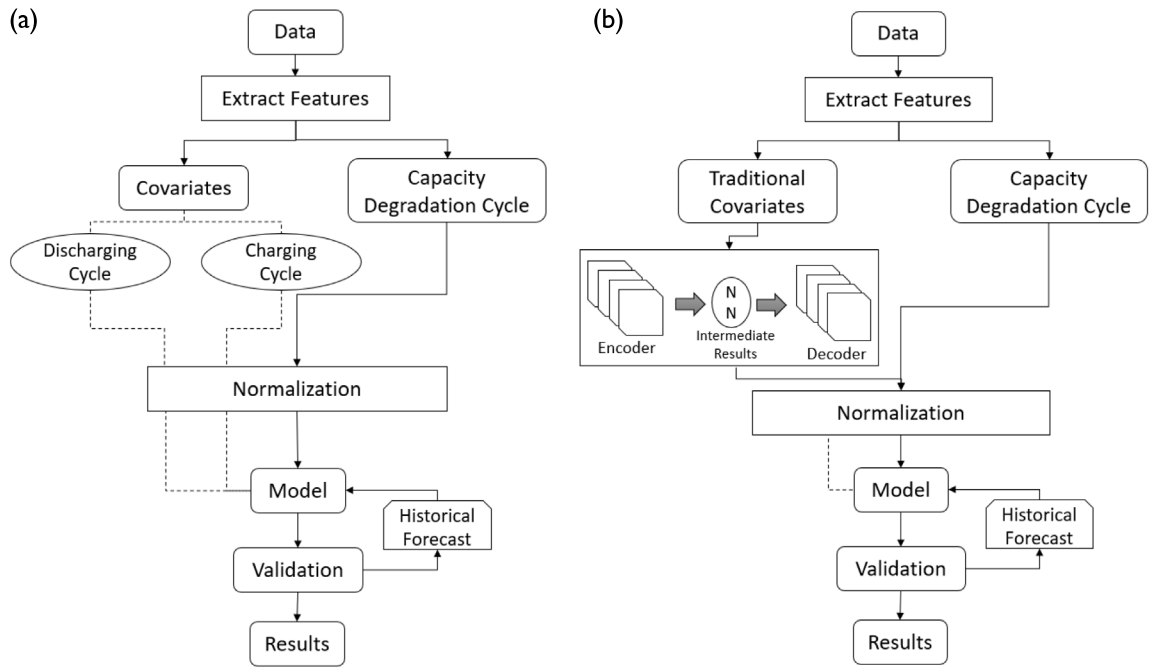

Fig. 3. Architectures used for forecasting (a) using charge–discharge time steps as covariates and (b) when traditional covariates are passed onto the encoder–decoder network.

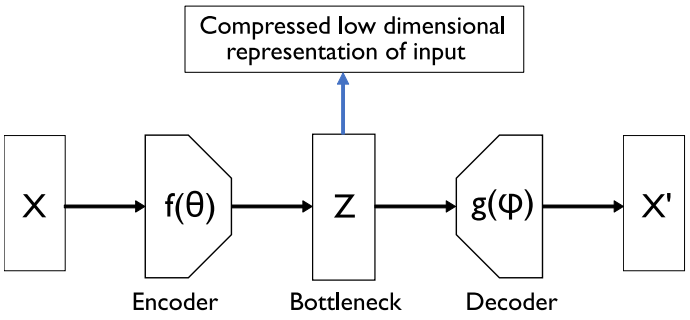

Fig. 4. A schematic highlighting the encoder–decoder for dimensionality reduction.

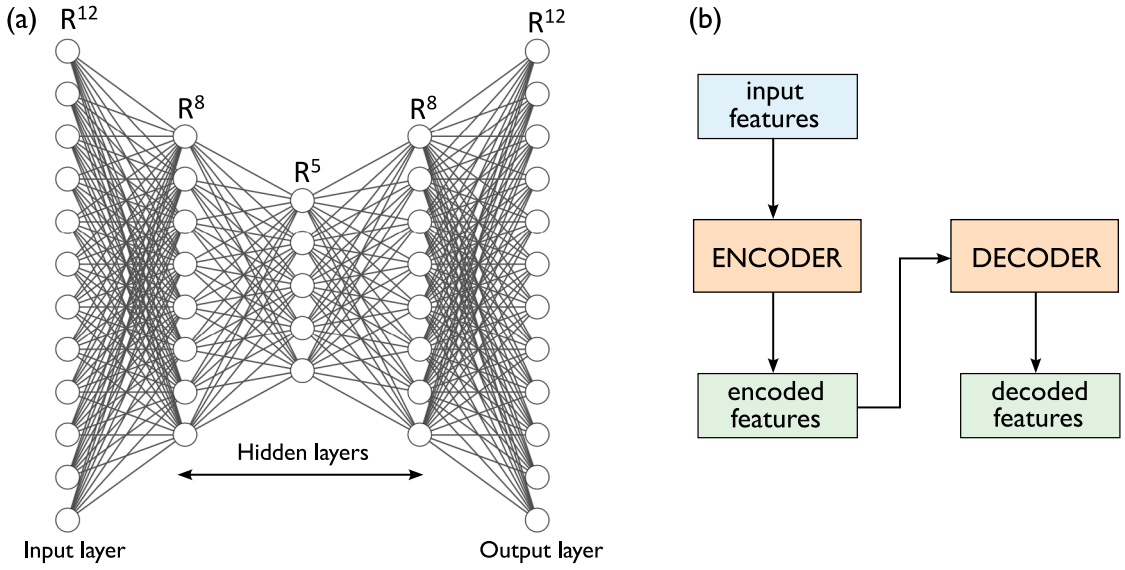

Fig. 5. (a) A deep machine learning model for dimensionality reduction and features workflow, where the flow of encoder and decoder is indicated in (b).

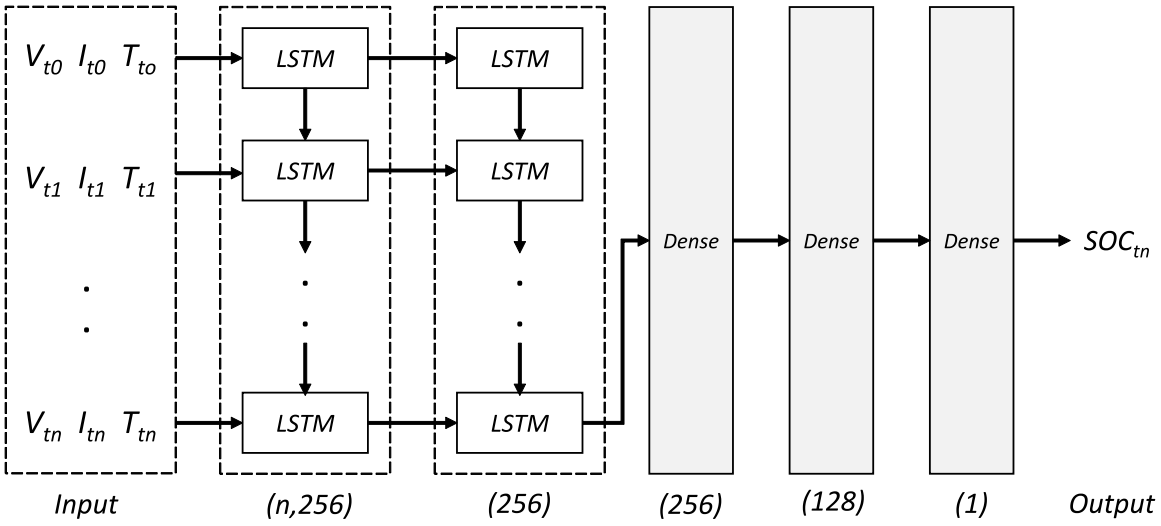

Fig. 6. A schematic of the long short-term memory architecture.

where the activation function f_W is parameterized by weights W, $\mathbf{x}_t$ is the input vector at time step t, and $\mathbf{h}_t$ and $\mathbf{h}_{t-1}$ represents the internal states at time steps t and (t-1), respectively. Therefore, according to Eq. (1), the internal state at a given time step depends on the internal state at the previous time step and the current input vector. Note that the same activation function f_W and the weight parameters W are used at every time instant t.

3.2. Gated recurring neural network

Two important gated recurring neural network models are: (i) long short-term memory unit [61] and (ii) gated recurrent unit [62]. The above approaches use a more sophisticated activation function than a usual activation function. RNNs employing the above recurrent units have been shown to perform well in tasks that require capturing long-term dependencies [63].

3.2.1. Long short-term memory unit

The LSTM network is used for time-series data prediction tasks, where the goal is to predict a future value of a time-series, for the known previous values. The LSTM network is capable of learning long-term dependencies, which is crucial for accurately predicting future values in a time-series. The originally proposed long short-term memory unit [61] has undergone several improvements to arrive at the implementation presented in [64]. The recurrent network computes a weighted sum of the input signal and applies a nonlinear function. Whereas, the LSTM unit maintains a memory (**c**) at time t. The output (**h**) of the LSTM unit is given by [63]:

$$\mathbf{h}_t = \boldsymbol{\sigma}_t \tanh(\mathbf{c}_t), \tag{2}$$

where $\boldsymbol{\sigma}_t$ is an output gate to modulate the amount of memory content exposure and computed as [63]:

$$\boldsymbol{\sigma}_t = \sigma(W_0\mathbf{x}_t + U_0\mathbf{h}_t + V_0\mathbf{c}_t), \tag{3}$$

where σ is a logistic sigmoid function and W_0, U_0 and V_0 represent the weights.

A schematic of the long short-term memory architecture consisting of an input layer, two LSTM layers, two dense layers and an output layer is shown in Fig. 6. The training data shape containing voltage, current and temperature information is provided to the input layer as input. According to Fig. 6, the first LSTM layer contains 256 units and uses the SELU activation function. The first layer returns sequences of output for each input time step through return_sequences=True. The second LSTM layer also contains 256 units and uses the SELU activation function. Unlike the first LSTM layer, the second layer returns only the final output of the sequence by setting the return_sequences=False. The output of the second LSTM layer is then forwarded to a dense layer with 256 units and a SELU activation function. Whereas, the second dense layer contains 128 units and also uses the SELU activation function. Finally, the output layer is a dense layer with a single unit and a `linear` activation function, where the target value for the next time step in the time-series is predicted.

3.2.2. Gated recurrent unit

In a gated recurrent unit each recurrent unit is made to adaptively capture dependencies of different time scales, with the help of gating units that modulate the flow of information inside the unit. However, unlike LSTM, separate memory cells does not exists in GRU. Therefore, the activation $\mathbf{h}_t$ of the GRU at time t is a linear interpolation between the previous activation $\mathbf{h}_{t-1}$ and the candidate activation $\tilde{\mathbf{h}}_t$, as shown below [63]:

$$\mathbf{h}_t = (1 - z_t)\mathbf{h}_{t-1} + \mathbf{z}_t\tilde{\mathbf{h}}_t, \tag{4}$$

where an updated gate $\mathbf{z}_t$ decides how much the unit updates its activation/content and is computed as [63]:

$$\mathbf{z}_t = \sigma(W_z\mathbf{x}_t + U_z\mathbf{h}_{t-1}). \tag{5}$$

The candidate activation is computed as [65]:

$$\tilde{\mathbf{h}}_t = \tanh(W\mathbf{x}_t + U(\mathbf{r}_t \odot \mathbf{h}_{t-1})), \tag{6}$$

where $\odot$ represents an element-wise multiplication and $\mathbf{r}_t$ is a set of reset gates, computed similar to the update gate:

$$\mathbf{r}_t = \sigma(W_r\mathbf{x}_t + U_r\mathbf{h}_{t-1}). \tag{7}$$

3.3. The transformer network

The transformer architecture is a modification of the LSTM layer for sequence modeling, where the LSTM layer is replaced with the transformer model, as shown in Fig. 7. The transformer model is a type of neural network architecture introduced in [66], which has achieved state-of-the-art results on several natural language processing tasks. The transformer model consists of an encoder and a decoder, see Fig. 7. The encoder processes the input sequence, and the decoder generates the output sequence. The encoder and decoder are composed of multiple layers of a basic building block, known as the 'transformer layer'. Therefore, the transformer layer consists of two sub-layers: the multi-head attention layer and the feed-forward neural network layer.

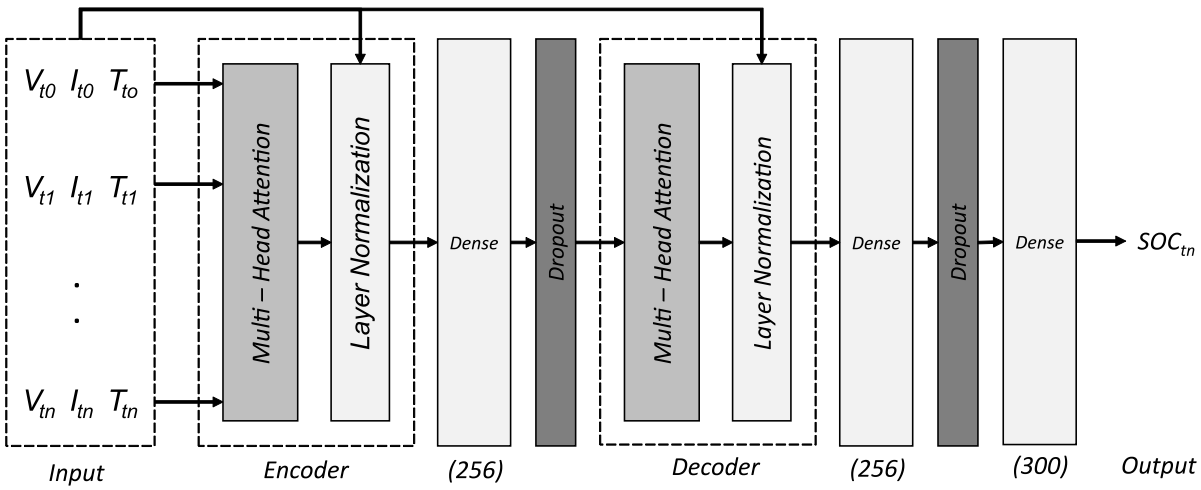

Fig. 7. A schematic of the transformer architecture.

The multi-head attention layer allows the model to attend to different parts of the input sequence and capture long-range dependencies. The feed-forward neural network layer applies a non-linear transformation to the output of the multi-head attention layer.

In the transformer model, the input layer is retained with the same shape as the input layer of the LSTM model. This is followed by addition of a multi-head attention layer with 8 heads and the key dimension and value vectors are set as 64, as the transformer encoder. The input sequence are passed to the multi-head attention layer in the sequence of: query, key, and value, as per the specification. This is followed by the addition of a normalization layer and a dense layer with 256 units and ReLU activation function, see Fig. 7. The layer normalization is added to normalize the activation of each layer to have zero mean and unit variance, which helps to stabilize the training of deep neural networks. A feed-forward neural network dense layer consisting of a single fully connected layer with 256 units and a ReLU activation function is added to the output of the layer normalization, see Fig. 7. Furthermore, as indicated in Fig. 7, a dropout layer is added to regularize the network. The dropout regularization randomly sets a fraction of the inputs to zero during training, which helps to prevent over-fitting. Later on, another multi-head attention layer with 8 heads and a key dimension of 64, apart from a normalization layer is added as the transformer decoder. Followed by the addition of a dense layer with 256 units and ReLU activation, and another dropout layer, see Fig. 7. Finally, a single unit dense layer is added to capture the output.

3.4. Convolutional neural networks

The CNNs are typically utilized for the completion of sequence modeling tasks. The selected CNN architecture in this study is a modification of one dimensional CNN architecture, which is highlighted in Fig. 8. According to Fig. 8, the Conv1D layers indicate the one-dimensional convolutional layers that apply a set of filters to the input data to extract local patterns/features. Each filter slides over the input sequence and performs a dot product with a window of the input at each position. The resulting output is a sequence of feature maps that capture different aspects of the input. The parameters of the filters are learned during training through back propagation. The shape of the output of each Conv1D layer depends on the number of filters, kernel size, stride, and the padding used. Whereas, the MaxPooling1D layers are one-dimensional pooling layers that down-sample the output from the previous layer by taking the maximum value over a fixed window of adjacent values. This operation helps to reduce the dimensionality of the data and make the model more efficient. The output shape of each MaxPooling1D layer depends on the pool size and the stride used. On the other hand, the GlobalMaxPooling1D layer is a pooling layer that takes the maximum value over the entire output sequence of the previous layer. This operation helps to capture the salient features of input sequence and reduce its dimensionality to a fixed-size vector. Finally, the dense layers are the fully connected layers that map the output from previous layer to a prediction for each time step in the output sequence.

The specifics of the architecture are follows. The output of a Conv1D layer with F filters, kernel size K, stride S, and padding P is applied to an input sequence X with length N is given by [67,68]:

$$\text{output[i,j]} = \sum_{K=0}^{K-1} (X[iS + P + K]\mathbf{F}[Kj]) \tag{8}$$

where the variable output indicates the output sequence of the layer, i is the index of the input sequence, j is the filter index, P is the padding size, and $\mathbf{F}$ is the weight tensor of shape (K, F).

The output of a MaxPooling1D layer with pool size P and stride S applied to an input sequence X with length N is given by:

$$\text{output[i]} = \max(X[iS : iS + P]) \tag{9}$$

where output indicates the output sequence of the layer, and i is the index of the pooled sequence.

Similarly, the output of a GlobalMaxPooling1D layer applied to an input sequence X with length N and F feature maps is given by:

$$\text{output[j]} = \max(X[i, j]), \quad i \in (0, N) \tag{10}$$

where output represents the output vector of the layer and j is the index of the feature map.

The output of a Dense layer with K neurons applied to an input vector X with length N is given by:

$$\text{output[j]} = \sum_{N=0}^{N-1} ((X[i]W[i, j]) + b[j]) \tag{11}$$

where output denotes the output vector of the layer, W is the weight matrix of shape (N, K), $\mathbf{b}$ is the bias vector of shape K, and j is the index of the neuron.

3.5. Forecasting models for degradation

In this study, neural basis expansion analysis for interpretable time series [69], autoregressive integrated moving average, and temporal fusion transformer [70] models, are used to forecast the health of the battery in architecture 2, see Fig. 5.

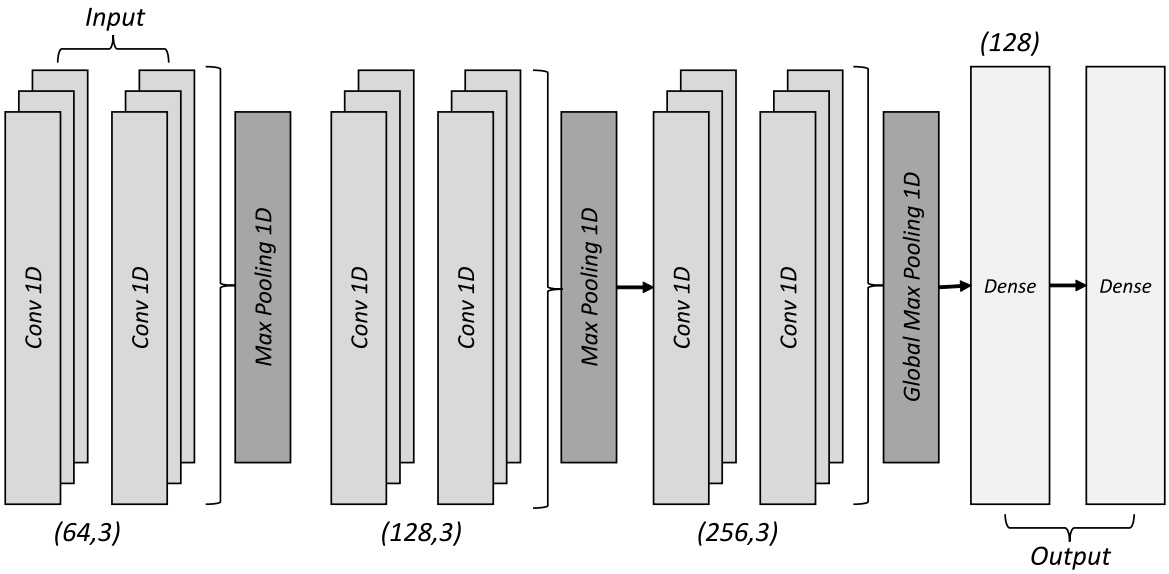

Fig. 8. A schematic of convolution based architecture. The corresponding number of units are indicated.

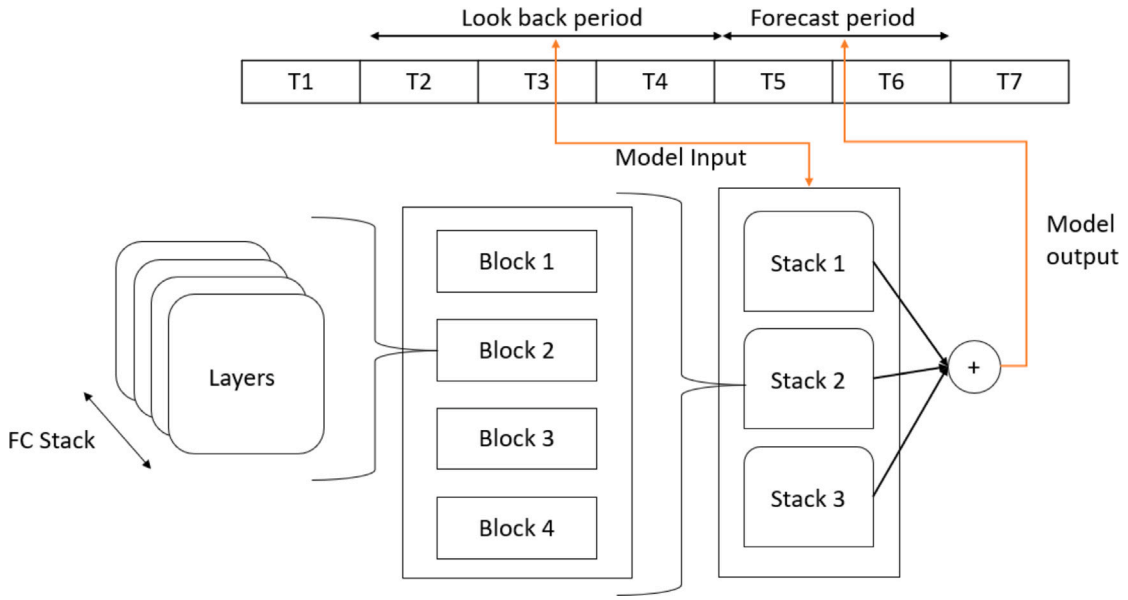

Fig. 9. Architecture of the NBEATS model used for the forecasting.

3.5.1. Neural basis expansion analysis for interpretable time series model

In this study, the neural network-based NBEATS forecast time series model shown Fig. 9 is adopted to forecast the battery health. The model comprises of the stack of completely connected layers and periodic and trend layers. The stack of completely connected layers learns input data patterns and generates high-level features that represent time series long-term interdependence. The periodic and trend layers provide forecasts by integrating high-level characteristics from the fully connected layers with periodic and trend patterns. Daily, weekly, and seasonal cycles are captured by the periodic layer. A CNN analyzes high-level characteristics from the fully linked layers and generates periodic features. Time series trends are captured by the trend layer. The fully connected layer processes periodic characteristics from the periodic layer and creates the final forecast.

As shown in Fig. 9, the top-level architecture of N-Beats contains three levels: (i) the block, which is the processing unit, (ii) the stack, which is a collection of blocks, and (iii) the architecture, which is a collection of stacks. The N-Beats model has a stack of completely connected layers with two branches for backcast and forecast signals. The doubly residual stacking is estimated as:

$$x_{l+1} = x_l + x'_l, \quad y_l = \sum_l y'_l. \tag{12}$$

Each block model estimates a subset of the input signal and transmits the rest to downstream blocks. Each block model is a portion of the input signal. Hence, the final forecast is the sum of all prediction signals from all blocks. The N-Beats model is trained using a variation of the mean squared error loss function, which compares predicted and actual time series values. The optimization is performed with the help of Adam/Stochastic gradient descent optimization algorithms, by minimizing the loss.

3.5.2. Autoregressive integrated moving average model

The ARIMA based models are used in a variety of industries, primarily for forecasting time series. The standard notation for ARIMA models is ARIMA (p, d, q), where p is the order of the autoregressive

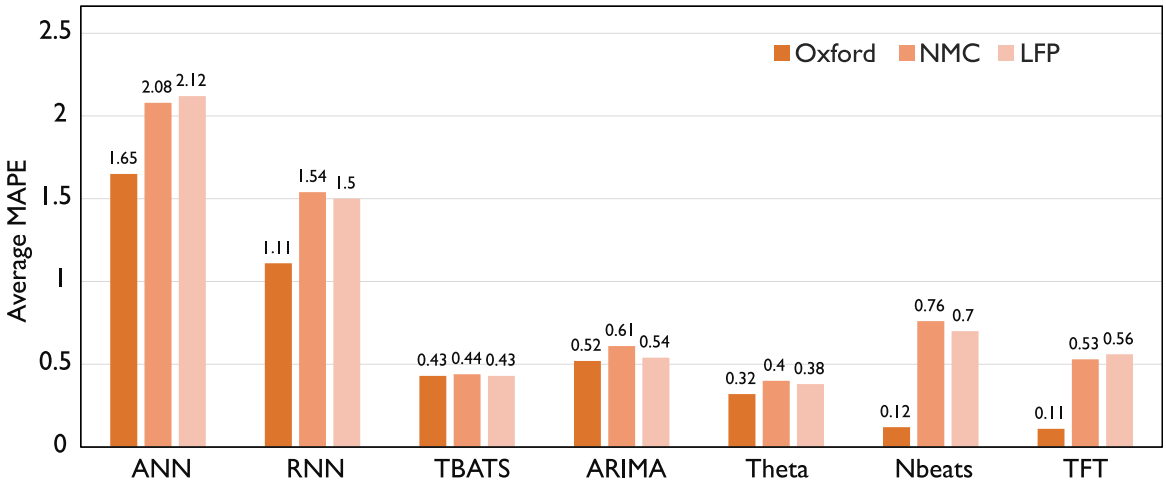

Fig. 10. A comparison of the average MAPE values of cells estimated through ANN, RNN, TBATS ARIMA, Theta, NBEATS and TFT models, using the data from Oxford, NMC and LFP batteries with respect to the results obtained from LSTM-based model.

model, d is the degree of differencing, and q is the order of the moving-average. ARIMA models use past data to predict the future by diffusing a non-stationary time series into a stationary series. These models forecast future values using `auto` correlations and moving averages over residual errors in the data. Some benefits of using ARIMA models are: (i) the forecast can be generalized using only the historical data of a time series, (ii) they provide accurate short-term forecasts and (iii) they can be employed to model non-stationary time series. In an ARIMA model, in order to obtain meaningful predictions, the stationarity is induced by differencing the time series. The components of ARIMA models, along with evaluating interventions and differencing operations are explained in [71].

3.5.3. The temporal fusion transformer model

The TFT model is a deep neural network model, which is based on attention. The major components of a TFT model are: (i) gated residual network involving ELU and GLU activation functions and mainly used to choose the most significant information for word prediction [72], (ii) variable selection network to identify useful information from noisy features [73,74] and (iii) the LSTM encoder–decoder module to produce context-aware embedding.

To summarize, the NBEATS, ARIMA and TFT models can be used for time series forecasting of battery life. TFT and NBEATS are deep learning models that can capture complex patterns in the data and provide accurate predictions. ARIMA is a traditional model that can be effective when the data follows a stationary pattern. The choice of model depends on the characteristics of the data and the requirements of the forecasting task. Time series forecasting of battery life can be achieved by feeding the historical data to the model to arrive at future predictions of the battery life. The predicted values can be used for battery health management and estimation of remaining useful life.

4. Results and discussion

Data normalization is a critical pre-processing step for preparing the input data for deep learning models. The scale and range of input data will be standardized through normalization, which helps to improve the efficiency and effectiveness of learning algorithm. In other words, without normalization, input features with high magnitudes can dominate the learning process, leading to a biased model that is less accurate in making predictions on new data. With the help of normalization, the input features can be scaled to have similar magnitudes and ranges, thereby preventing the domination of any particular feature in the learning process, resulting in a more stable and accurate model. The maximum normalization scheme is used is in this study, as mentioned below:

$$\mathbf{x}_{\text{normalized}} = \frac{\mathbf{x} - \mathbf{x}_{\text{min}}}{\mathbf{x}_{\text{max}} - \mathbf{x}_{\text{min}}} \tag{13}$$

where $\mathbf{x}$ indicates the available data and $\mathbf{x}_{\text{min}}$ and $\mathbf{x}_{\text{max}}$ represent the minimum and maximum values of the resource data.

Furthermore, during the training process the data is iterated through the network model until the error reaches the acceptable set limits. The training data is divided into batches of 32 size. The percentage of the training data that will be utilized for validation is specified by the validation split. In this study, training and testing data is split in the 20:80 ratio. The call backs: early stopping and model checkpoint to stop the training process if the loss does not improve for a specified number of epochs and to monitor the validation loss, respectively, are employed.

4.1. Mean average percentage error

The developed models are thoroughly trained and the data is tested, considering various parameters, like: temperature, driving profile, and change in impedance. This is achieved by comparing the forecasting results with the results predicted using the LSTM algorithm. A comparison of the MAPE values estimated from ANN, RNN, TBATS [75], ARIMA, Theta, NBEATS and TFT models, using the data from Oxford, NMC and LFP batteries with respect to the results obtained from LSTM-based model is provided in Fig. 10. The data-sets of real-time batteries used in the electric vehicle industry with more than 1500 cycles are considered for the analysis. The mathematical models like TBATS, ARIMA, and Theta are found to have lesser MAPE, with values less than 0.8% and an accuracy of 99%, as compared to the DML based models. Therefore, according to Fig. 10, models with mathematical input are observed to work better on data-sets with fewer cycles.

However, considering the high-cycle data-sets and the data-sets with better parametric results, like: Oxford and NMC, the mathematical models are observed to fail to interpret the complex physics associated with the batteries. The ANN is found to fail consistently as compared to other models in all kinds of cycles, and its inability to interpret the sequential data is evident, see Fig. 10. During the analysis of large Oxford degradation data-set, the DML models with capabilities to accommodate seasonality and trends to account for large number of cycles, i.e., TFT and NBeats models, are found to outperform every other model, with MAPE values less than 0.2%. Also, the above models are found to perform well with respect to experimental data-sets as well.

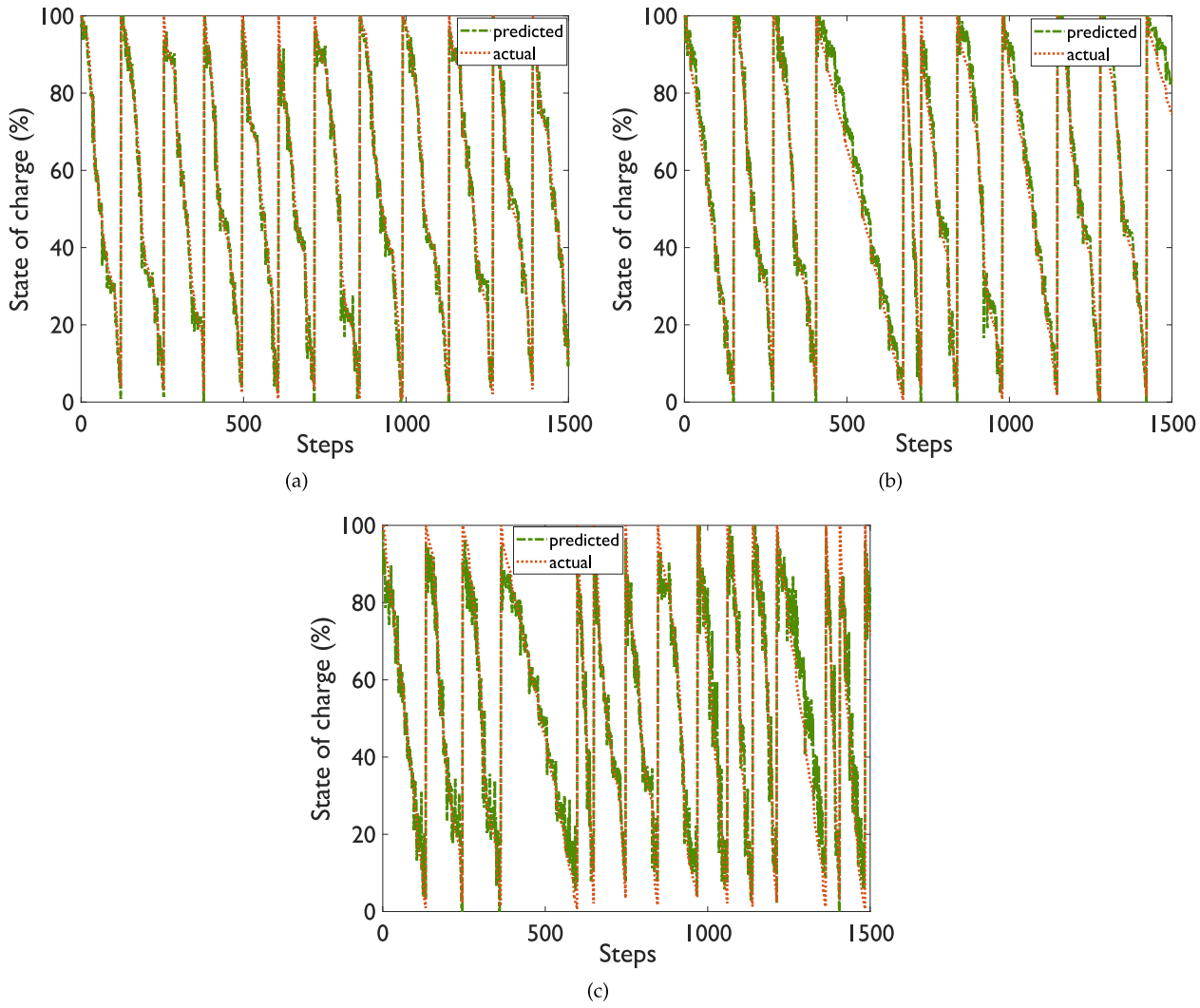

Fig. 11. A comparison of the actual and predicted state of charges of (a) cell 1, (b) cell 2 and (c) cell 3, considering the drive cycles during the first 1550 s.

4.2. State of charge

In order to predict the state of charge, the LG data-set is used to train the LSTM model. Furthermore, the state of charge is estimated with the help of the drive cycle test data from US06 and UDDS data-sets, as well as the mixed cycles data, where various methods of charging and discharging were combined to achieve the cumulative effect of a series. The model is provided with the processed sequential data for the first three different cells from the adopted data-set.

The developed network model is trained using the entire data-set, and the testing is performed in different steps. A comparison of the actual and predicted state of charges of cells 1–3 over a period of 1550 s is shown in Fig. 11, considering multiple charge–discharge cycles during the drive phase. In the similar lines, a comparison of the actual and predicted state of charges of cells 4, 5, 6, and 8 considering the mixed cycles, during the first 1550 s is provided in Fig. 12. According to Figs. 11 and 12, the predicted values are observed to be closely following the actual values. Therefore, the developed networks can be concluded to be properly trained.

The developed three different neural network models are tested considering four different driving test data-sets. The driving tests are a combination of US06, LA92, and UDDS, which were conducted in mixed form to test the overall functioning of the model. Moreover, additional mixed and drive cycles are considered at different temperatures: 10 °C and 25 °C. A comparison of the test results predicted using LSTM, CNN, and Transformer network models with respect to the actual data, corresponding to the mixed and drive cycles at 10 °C are provided in Figs. 13(a) and (b). According to Figs. 13(a) and (b), the transformer architecture is observed to outperform the CNN model. Whereas, the predictions from CNN model are in general better as compared to the traditional LSTM model. Also, the predictions are observed to deviate in the initial stages, and converge with increase in number of cycles.

Furthermore, the predictions from the LSTM model are found to be on the lower side as compared to the actual data. Therefore, due to large bias, application of a filter to reduce the fluctuations will not be helpful. On the other hand, the transformer architecture includes an attention mechanism, which allows the network to focus on relevant parts of the input sequence. This is particularly useful for sequential data with long-range dependencies. The attention mechanism also enables the model to capture complex patterns and relationships between input elements more effectively, as compared to the LSTM, which relies on a fixed hidden state. The superior performance of the transformer network and CNN models is due to the differences in hyper parameters, such as: the number of layers, the learning rate, and/or the batch size. Tuning hyper parameters is often necessary to achieve the best performance for a given application.

A CNN is particularly effective in capturing local patterns of sequential data. For instance, CNN can identify specific features in image classification of a set of images. In natural language processing, a CNN can capture local patterns in the text, like n-grams and specific word

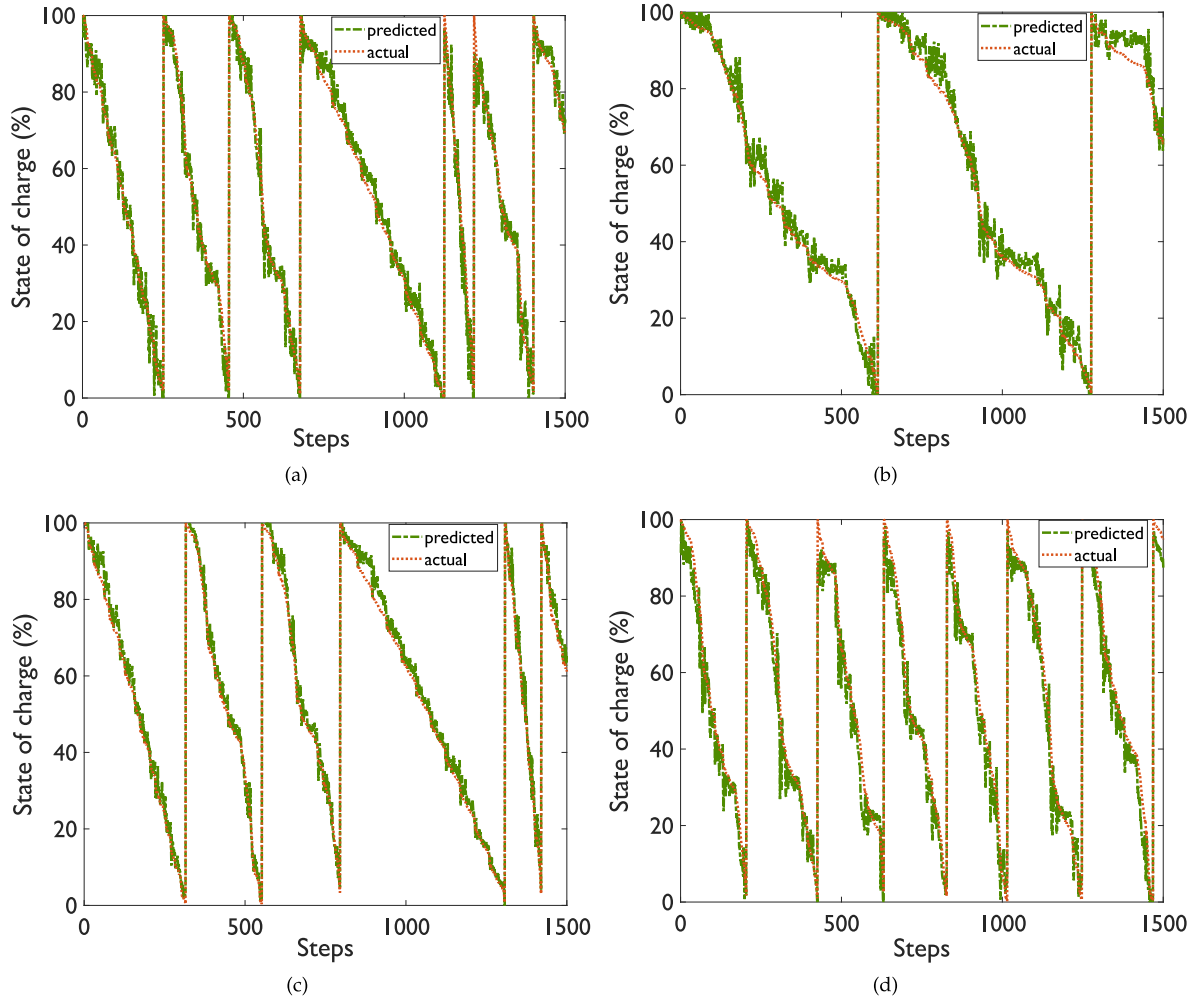

Fig. 12. A comparison of the actual and predicted state of charges of (a) cell 4, (b) cell 5, (c) cell 6 and (d) cell 8, considering the mixed cycles during the first 1550 s.

combinations. A LSTM network is designed to store and use information from previous time steps. However, its memory is limited to a fixed size. This means the LSTM provides poor results with long sequences of data, especially if the input data has many features. In contrast, the transformer architecture can handle longer sequences and larger input sizes.

A comparison of the test results predicted using LSTM, CNN, and Transformer network models with respect to the actual data, corresponding to the mixed and drive cycles at 25 °C is provided in Figs. 13(c) and (d). An observation of Figs. 13(c) and (d) indicate that the results predicted using transformer architecture are better than the results predicted using other two networks, due to the presence of multi-head attention module in the transformer architecture. Small fluctuations in the predicted data can be eliminated by employing an appropriate filter. Thus, the predictions can be extended to real-life problems.

An observation of the state of charge variation in Fig. 13 indicates that the operation of the battery is more stable at room temperature, i.e., at 25 °C. Also, the employed network model is able to predict the battery behavior more accurately at room temperature, without the need for a long history of data. However, the battery performance is found to be better at 10 °C and 0 °C, at 500 input steps. This suggests that increasing the number of input steps can help to improve the estimation of results when operating under temperatures lower than room temperature. As a result, the number of input steps has to be carefully selected, as inappropriate increments could result in performance degradation. Therefore, choosing the appropriate number of input steps is crucial when using a many-to-one approach to estimate the battery behavior at different temperatures.

The findings of this study provide insights to the optimal configurations for predicting battery behavior under different temperatures. This helps to develop efficient battery management systems, which can lead to optimize battery performance and prolong the lifespan. As a result, development of efficient and reliable energy storage systems can be realized.

4.3. Hyper-parameter tuning

The MAE and RMSE estimated based on the predictions from the developed three different networks: LSTM, CNN, and transformer models, and trained on sequential data at three different temperatures: 0 °C, 10 °C, and 25 °C are plotted in Figs. 14(a) and (b), respectively. The models are trained to predict the given target variable based on a sequence of input data. At 0 °C, both the MAE and RMSE values of all three networks is observed to decrease as the number of steps in the sequence increases from 300 to 700, see Fig. 14. This suggests that longer sequences are helpful for better predicting the target variable. This is because the target variable is influenced by a longer history of input data at 0 °C, as longer sequences can capture more of this history.

The transformer network model is observed to consistently provide better results with lesser MAE and RMSE values, as compared to the LSTM and CNN models at all three temperatures. The transformer model is found to perform better for all three sequence lengths, indicating that it is the most effective network in capturing the relevant

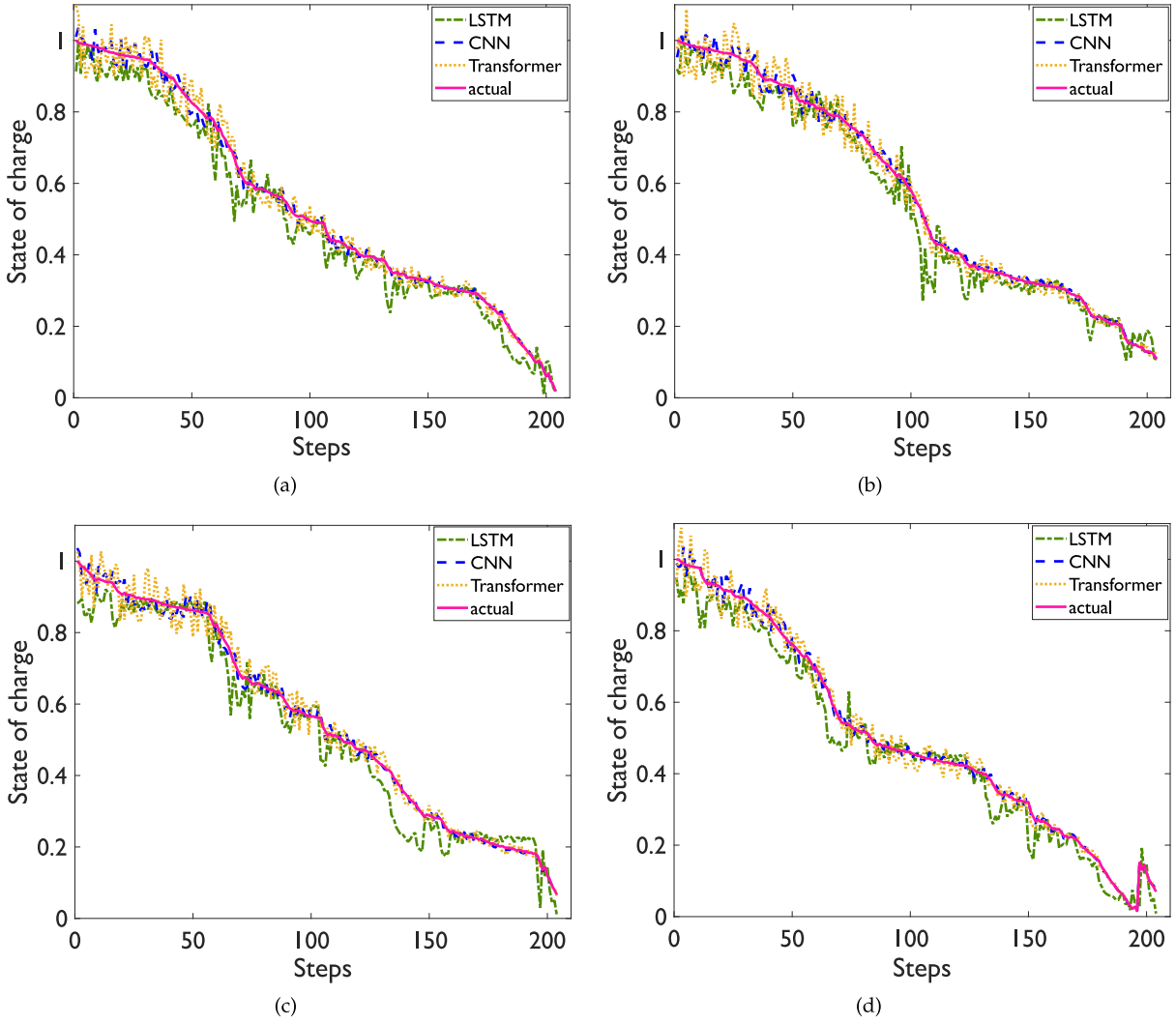

Fig. 13. A comparison of the test results of (a) cell 1, and (b) cell 2 at 10 °C, (c) cell 3 and (d) cell 4 at 25 °C, predicted using LSTM, CNN, and Transformer network models with the actual data, corresponding to the mixed and drive cycles at 10 °C.

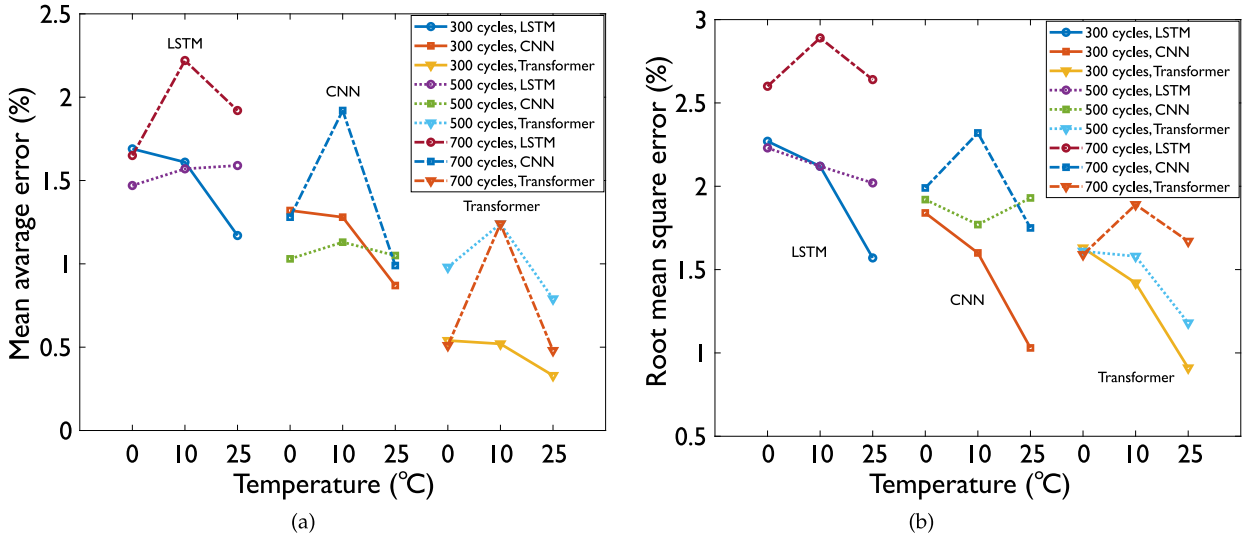

Fig. 14. Variation of percentage (a) MAE and (b) RMSE values predicted using LSTM, CNN and transform network models with 300, 500 and 700 number of steps in data sequences, at 0 °C, 10 °C and 25 °C temperature.

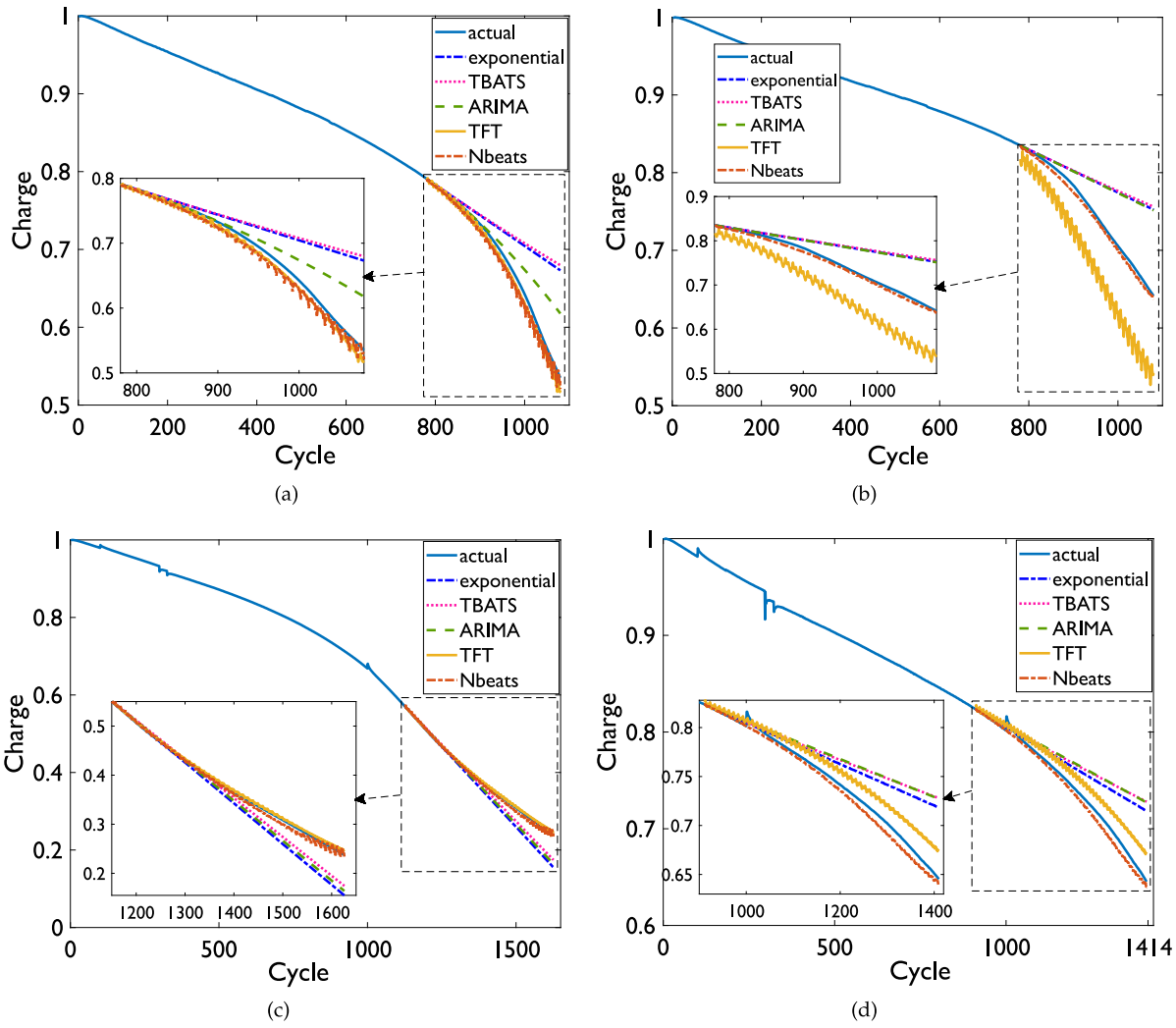

Fig. 15. Variation of charge with number of cycles estimated using the proposed forecasting models in architecture 2, considering the multivariate series of NMC cells for (a) cell 1, (b) cell 2, (C) cell 3, and (d) cell 4, during the last 500/300 cycles of the battery discharge.

information from short and long sequences. On the other hand, at 10 °C all the models are observed to exhibit a trend of increasing MAE and RMSE values as the number of steps in the sequence increases from 300 to 700, see Fig. 14. This indicates that longer sequences are not helpful for predicting the target variable at 10 °C. However, the predictions from the transformer model are better with lesser MAE and RMSE values, indicating that it is most effective at capturing the relevant information regardless of sequence length.

Furthermore, at 25 °C, the LSTM model exhibits a trend of increasing MAE and RMSE values as the number of steps in the sequence increases from 300 to 700, see Fig. 14. This indicates that longer sequences are not helpful for the LSTM model in predicting the target variable at 25 °C. On the other hand, results from the CNN model are observed to exhibit a decreasing trend in MAE and RMSE values as the number of steps increases from 500 to 700, suggesting that longer sequences are helpful for the transformer model in capturing the relevant information for predicting the target variable at 25 °C. However, the overall MAE and RMSE values are lowest for the predictions using transformer network. In general, it is possible to have long-term patterns/trends in the input data which are relevant for predicting the target variable at 25 °C. The transformer model is better in capturing these patterns. Therefore, the transformer model is better suited for sequential data tasks as compared to the LSTM and CNN models.

To summarize, the predicted results indicate that the transformer model is the most effective for the analysis of sequential data. However, the performance of LSTM and CNN models depends on the specific characteristics of the data and the task. Considering the relevant patterns and trends in the input data, the sequence length also influences the performance of different models with temperature.

4.4. Analysis of predicted results from the forecasting models

Variation of charge with number of cycles estimated using the proposed forecasting models in architecture 2, considering the multivariate series of NMC cells 1–4, during the last 500/300 cycles of the battery discharge are shown in Fig. 15. The forecasting models mainly consists of covariates which are passed through the encoder–decoder network to reduce the dimensionality of the features. Later on, these features of past covariates are used for forecasting the results at the end of the cycle. Four different NMC cells at room temperature are considered for forecasting. The mathematical models, like: exponential and TBATS, are observed to produce large errors when they are applied to analyze the cells containing the smallest number of cycles. This is due to the inability of the mathematical models to read the complex internal physics within the cell. The ARIMA model is observed to perform better as compared to other models. This is because ARIMA model is able to correctly read covariates, seasonality, and trends. The model is particularly designed to perform the time analysis. The results estimated from NBEATS and TFT networks are found to be better.

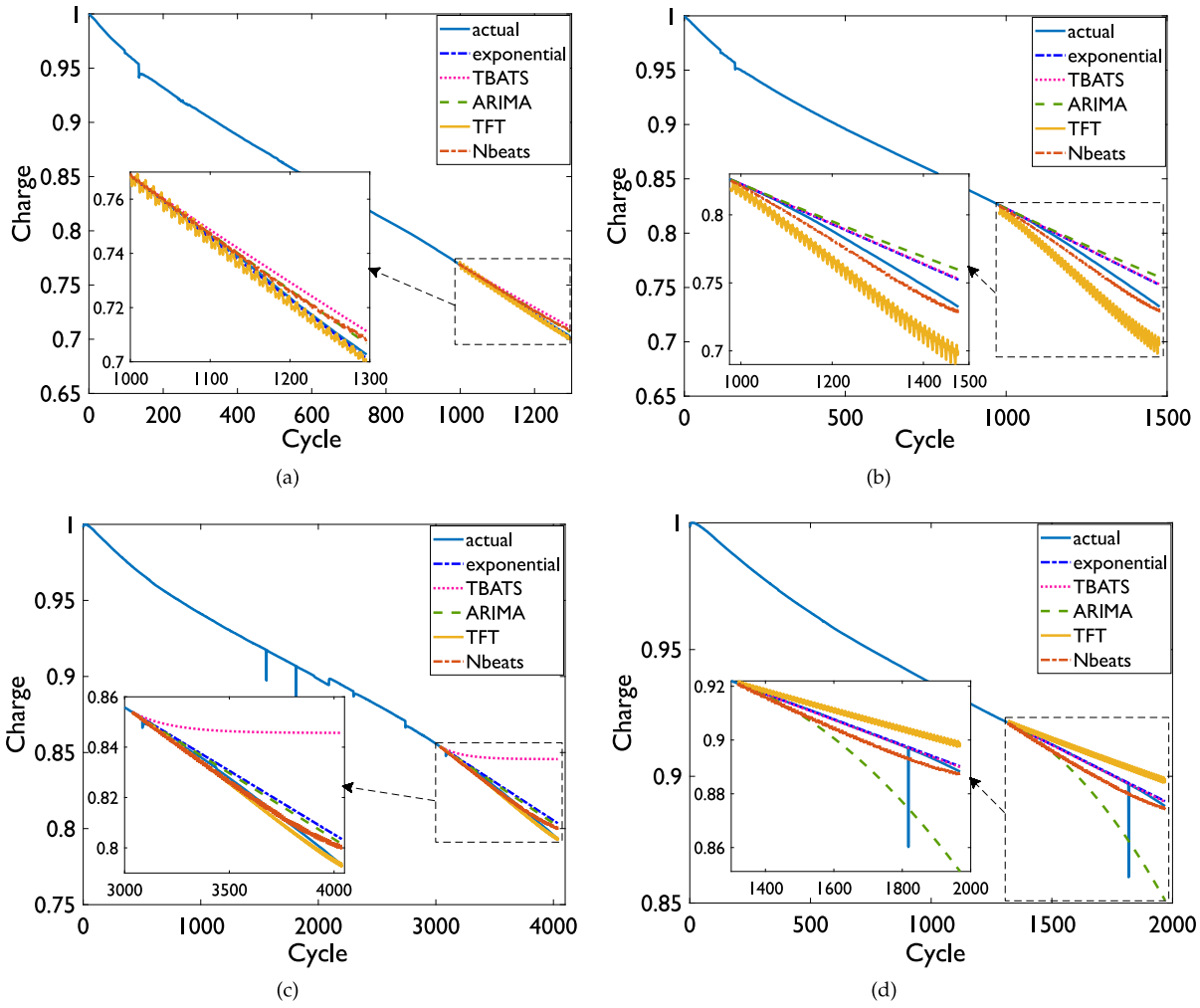

Fig. 16. Variation of charge with number of cycles estimated using the proposed forecasting models in architecture 2, considering the multivariate series of LFP cells for (a) cell 1, (b) cell 2, (C) cell 3, and (d) cell 4.

However, some bias in cell 2 along with the variations in results from the TFT network is noticed. The MAE value is estimated to be 0.78%.

Condition filters are employed to separate the variations in the forecast cab and hence, improve the performance of the forecasting models. However, the N-Beats model is found to be the best model for forecasting. This is mainly due to the fact that the model is lighter as compared to the transformer based TFT model, which require large amounts of data to train the network. With limited covariates, which are reduced in size by the dimensionality reduction using the encoder–decoder network, N-Beats network is found to effectively learn the health of the batteries. Similarly, variation of charge with number of cycles estimated using the proposed forecasting models in architecture 2, considering the multivariate series of LFP cells 1–4, during the last 500/300 cycles of the battery discharge are shown in Fig. 16. The cells are maintained at room temperature, i.e., at 25 °C, and the reduction in battery capacity with the number of cycles is estimated through forecasting. Similar observations are noticed in case of LFP cells as well. The TBATS and the exponential based mathematical models are unable to provide accurate forecasts, since they fail to capture the actual curve during the end cycles and possess a significant amount of bias. However, the bias can also be observed in the N-Beats and the TFT models as well. This is because the LFP data-set contains a significantly smaller number of covariates. The predictions from N-Beats network are significantly better than any other models, with mean absolute errors of 0.32%, 0.41%, 0.28%, and 0.38% for cells 1–4, respectively. Therefore, a comparison of the results from numerous models contained within the same architecture is depicted in Figs. 15 and 16. The findings indicate that N-Beats and TFT are the best models among the considered networks, where the results from these models are comparable to those from a fully trained network.

On the other hand, a comparison of the mean absolute error estimated using the architectures: normal variates mean traditional method, feature reduction with auto encoders and charge/discharge times of each cycle, is provided in Fig. 17. The performance of mathematical models can be concluded to be optimal for a limited number of cycles. This is true for both the LFP and the Oxford regeneration data-sets. According to Fig. 17, LFP3 is observed to exhibit best performance with the lowest MAE value of 0.3% among the LFP cell types, when feature reduction with auto encoders are used as covariates. This is because the encoder/decoder model can reduce the number of covariate dimensions. However, NMC cell4 achieves the best performance with the lowest MAE value of 0.3%, as compared to other NMC cell types when charge/discharge times of each cycle are used as covariates. While utilizing the encoder/decoder architecture for dimensionality reduction of covariates, this model is observed to outperform other models in terms of MAE to parameter ratio. Encoder/decoder is observed to perform particularly well for LFP cells 2–4, and NMC cell 4, with the lowest MAE values of 0.4%, 0.3%, 0.55%, and 0.33%, respectively, see Fig. 17.

A comparison of the results: MAPE, parameter size, ratio of MAPE to parameter size, from three different architectures is provided in Fig. 18. As discussed above, the MAPE is also observed to be lower for

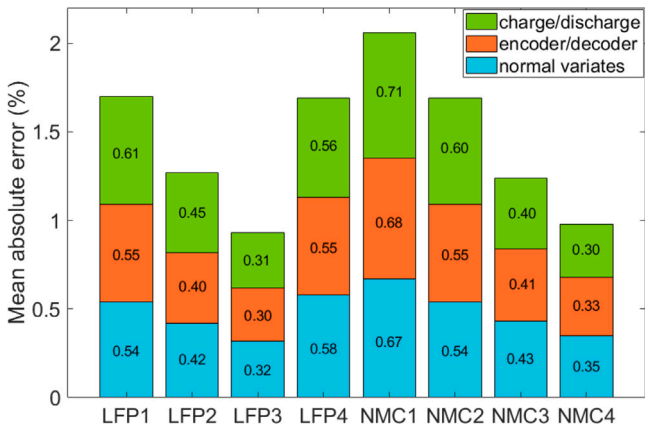

Fig. 17. A comparison of the mean absolute error estimated from the following architectures: normal variates mean traditional method, feature reduction with auto encoders and charge/discharge times of each cycle.

LFP cells 2–4, and NMC cell 4 in all the architectures, see Fig. 18(a). The corresponding parameter size is found to be more than 60% for LFP cells 2–4 and more than 30% NMC cell 4 considering normal variates, see Fig. 18(b). Whereas, the corresponding parameter sizes are found to be less than 20% and 10% considering encoder/decoder and charge/discharge cycle architectures, respectively. Similar trend is observed in the ratio of MAPE to parameter size plots shown in Fig. 18(c). Therefore, the parameter size and the ratio of MAPE to parameter size are sensitive to the type of selected architecture. The parameter size reduces and hence the ratio of MAPE to parameter size increases from normal variates to charge/discharge cycle architectures. In general, the performance of the above models differ depending on the particular cell type and the employed covariates in the analysis, see Fig. 18. As a result, it is essential to provide careful consideration to various model architectures and covariate options, as well as conduct experiments with the obtained results, in order to select the most suitable strategy for forecasting the capacity loss of batteries.

A summary of the neural network models used in this study for forecasting is provided in Table 2. As shown in Table 2, ANN, NBEATS and DFT models are DNN based models, whereas, TBATS, ARIMA and Theta are statistical models and LSTM is a recurrent network model. The performance of the statistical models is better when the data contains small number of cycles with requirements of less data and space. Whereas, DNN based models require large data and more time for training, whereas, they perform better when the data contains small and large cycles with less features.

4.5. Limitations

The predictive models are observed to produce larger MAE when trained with data from other sources. For instance, a MAE of 0.05% for cell 3 with the present data is better than a value of 4.8% with data from other sources. Although this problem can be avoided by training the model using a variety of data-sets, this is not feasible in real time. On the other hand, imposing appropriate conditions to a real time model combined with a variety of data-sets can improve the accuracy. However, this will lead to increased complexity, as these models consider multiple features and their predictions are based on the initial trained weights. As a result, the process is required to be repeated whenever the battery is changed.

To summarize, the transformer-based architecture is recommended for accurate prediction of SOC. The proposed innovative forecasting method is a promising avenue for achieving accurate predictions, which allows better resource planning and optimal utilization of energy storage resources. The challenges associated with predictive analysis along with some feasible effective solutions are also discussed. For instance, by employing Dijkstra algorithms, which is a powerful tool in graph theory, a wide range of predictive and forecasting models can be effectively applied to enhance the design of batteries, identify potential battery failures, and optimize charging station systems. These models offer valuable insights and enable comprehensive component analysis. However, the conventional approach of generating samples to derive results can be time-consuming. Thankfully, by leveraging various forecasting techniques, these models can swiftly produce accurate outcomes, eliminating the need for lengthy sampling processes.

5. Conclusions

An attempt is made here to predict the SOC of Li-ion batteries using three different network models: LSTM, NBEATS and transformer-based architectures. The proposed models are made robust by conducting various testing scenarios, such as: mixed and drive cycles, apart from different temperature conditions. The models are able to capture the dynamic behavior of the battery during discharge accurately. This is confirmed by the analysis of the recorded voltage, current, and temperature data, and the precise estimation of SOC. The results indicate that the transformer-based architecture is better compared to the other two architectures, achieving a significantly low mean error of 0.32%, as compared to a mean error of 1.47% from the LSTM architecture.

Furthermore, a novel forecasting method based on the time-series analysis is also introduced. The accuracy of the results from the time-series analysis is found to be significantly high as compared to the conventional approaches. Moreover, better results are achieved with the help of NBEATS architecture in conjunction with encoder–decoder-based feature extraction, where an accuracy rate is found to exceed

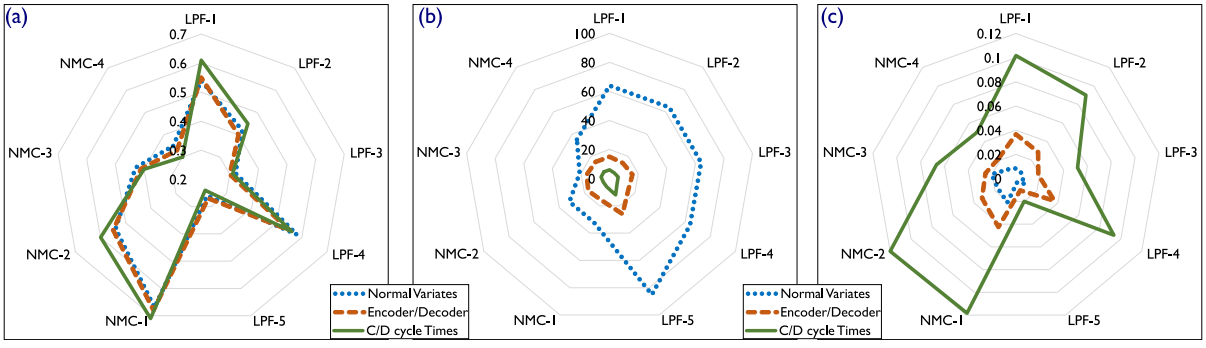

Fig. 18. Comparison of the results: (a) MAPE (b) parameter size (c) ratio of MAPE to parameter size, from three different architectures.

Table 2
Summary of models used in this study for forecasting.

Model	Type	Parameters	Performance (%)	Remarks
ANN	Deep neural network	Optimizer = Adam, Matrices = MAPE, MAE, No. of layers = 3, Layer width = 64.	2.08	Simple architecture, Easy to use, Requires large data for training.
LSTM	Recurrent network	Layer width = 64, No. of dense layers = 2, LSTM layers = 2, LSTM layers width = 16, Matrices = MAE, Activation: RELU.	1.54	Require covariates, Shown best results in past, Requires large data for Training, Short and Long distance , Dependencies are also captured.
TBATS	Statistical	Box–Cox, ARMA errors, Seasonal components.	0.44	Statistical method, Less space requirement, Less data required.
ARIMA	Statistical	Feature Implementation, Stranded darts.	0.61	Statistical method, Heavier than TBATS, Less data required, Past covariates needed.
Theta	Statistical	Feature Implementation, Stranded darts.	0.40	Statistical method, Past covariates needed, Works better for small cycles.
Nbeats	Deep neural network	Generic architecture, No. of stacks = 30, No. of blocks = 4, No. of layers = 4, Layer width = 128.	0.76	Past covariates needed, Best results among peers, for both small and large cycles, Heavy size, Less features required.
TFT	Deep neural network	Hidden layer width = 64, No. of LSTM layers = 1, No. of attention heads = 4, Dropout = 0.1, Relative indexing.	0.53	Past covariates needed, Best for large cycles, Captures sudden change in discharge perfectly, Less features required, More training time needed.

98% and MAE values are less than 0.2%. The findings of this study can significantly influence the design of battery management systems and energy storage applications.

5.1. Future scope

The integration of the developed models into IoT based devices opens up new possibilities. By implementing them within IoT frameworks, it is feasible to receive timely alerts/alarms when a battery requires maintenance/replacement. This proactive approach ensures that batteries are properly managed, maximizing their lifespan and minimizing the risk of sudden failures.

Furthermore, the models can be exported to web-based APIs, enabling seamless integration into battery-related applications. This integration empowers developers and engineers to create better and more durable batteries by utilizing the predictive capabilities of these models. With the help of APIs, the models can be harnessed to optimize battery designs, evaluate different materials and configurations, and ultimately contribute to advancements in battery technology.

Therefore, the utilization of Dijkstra algorithms in predictive and forecasting models offers immense benefits for battery-related applications. These models streamline battery design, enable early detection of potential failures, optimize charging station systems, and facilitate component analysis. By incorporating these models into IoT devices, battery maintenance can be significantly improved, and by exporting them to web-based APIs, the development of more resilient and efficient batteries will be feasible.

CRediT authorship contribution statement

S. Singh: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation. **P.R. Budarapu:** Writing – review & editing, Visualization, Supervision, Resources, Project administration, Methodology, Conceptualization.

Declaration of competing interest

We confirm that the manuscript is not published elsewhere and is not under consideration by another journal. All the authors declare no conflict of interest. They approved the submission of the manuscript to the special issue.

Data availability

Data will be made available on request.

Acknowledgments

The second author (PRB) is thankful to the Science and Engineering Research Board, Department of Science and Technology, India, for funding this research through grant number SRG/2019/001581.

References

[1] Siruvuri Varma, Verma H, Javvaji B, Budarapu PR. Fracture strength of Graphene at high temperatures: data driven investigations supported by MD and analytical approaches. Int J Mech Mater Des 2022;18(4):743–67.
[2] Sharma Shivam, Awasthi Rajneesh, Sastry Yedlabala Sudhir, Budarapu Pattabhi Ramaiah. Physics-informed neural networks for estimating stress transfer mechanics in single lap joints. J Zhejiang Univ-SCI A 2021;22(8):621–31.
[3] Budarapu PR, Kumar S, Khan MA, Rammohan B, Anitescu C. Engineered interphase mechanics in single lap joints: analytical and PINN formulations. Int J Comput Methods 2022;19(08):2143021.
[4] Verma H, Siruvuri Varma, Budarapu PR. A machine learning-based image classification of silicon solar cells. Int J Hydromechatron 2024;7(1):49–66.
[5] Ling Chen. A review of the recent progress in battery informatics. npj Comput Mater 2022;8(1):33.
[6] Zhang Liyuan, Shen Zijun, Sajadi S Mohammad, Prabuwono Anton Satria, Mahmoud Mustafa Z, Cheraghian G, El Din ElSayed M Tag. The machine learning in lithium-ion batteries: a review. Eng Anal Bound Elem 2022;141:1–16.
[7] Chandra K Parthiv, Budarapu PR. Design and analysis of Lithium-ion pouch cell with LMO-NMC blended cathode using coupled thermo-electro-chemical model. J Energy Storage 2024;78:109958.

[8] Poh Wesley Qi Tong, Xu Yan, Tan Robert Thiam Poh. A review of machine learning applications for li-ion battery state estimation in electric vehicles. In: 2022 IEEE PES innovative smart grid technologies-Asia. ISGT Asia, IEEE; 2022, p. 265–9.

[9] Ren Zhong, Du Changqing. A review of machine learning state-of-charge and state-of-health estimation algorithms for lithium-ion batteries. Energy Rep 2023;9:2993–3021.

[10] Jin Siyu, Sui Xin, Huang Xinrong, Wang Shunli, Teodorescu Remus, Stroe Daniel-Ioan. Overview of machine learning methods for lithium-ion battery remaining useful lifetime prediction. Electronics 2021;10(24):3126.

[11] So Magnus, Inoue Gen, Park Kayoung, Nunoshita Keita, Ishikawa Shota, Tsuge Yoshifumi. Contact model for DEM simulation of compaction and sintering of all-solid-state battery electrodes. MethodsX 2022;9:101857.

[12] How Dickson NT, Hannan MA, Lipu MS Hossain, Ker Pin Jern. State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review. IEEE Access 2019;7:136116–36.

[13] Ng Man-Fai, Sun Yongming, Seh Zhi Wei. Machine learning-inspired battery material innovation. Energy Adv 2023;2(4):449–64.

[14] Zhao Jingyuan, Han Xuebing, Ouyang Minggao, Burke Andrew F. Specialized deep neural networks for battery health prognostics: Opportunities and challenges. J Energy Chem 2023.

[15] von Bülow Friedrich, Meisen Tobias. A review on methods for state of health forecasting of lithium-ion batteries applicable in real-world operational conditions. J Energy Storage 2023;57:105978.

[16] Pepe Simona, Ciucci Francesco. Long-range battery state-of-health and end-of-life prediction with neural networks and feature engineering. Appl Energy 2023;350:121761.

[17] Medina Marie Chantelle Cruz, de Oliveira João Fausto L. A selective hybrid system for state-of-charge forecasting of lithium–ion batteries. J Supercomput 2023;1–20.

[18] Zhang Lijun, Ji Tuo, Yu Shihao, Liu Guanchen. Accurate prediction approach of SOH for lithium-ion batteries based on LSTM method. Batteries 2023;9(3):177.

[19] Hossain Lipu MS, Hannan MA, Hussain Aini, Ayob Afida, Saad Mohamad HM, Muttaqi Kashem M. State of charge estimation in lithium-ion batteries: A neural network optimization approach. Electronics 2020;9(9):1546.

[20] Charkhgard Mohammad, Farrokhi Mohammad. State-of-charge estimation for lithium-ion batteries using neural networks and EKF. IEEE Trans Ind Electron 2010;57(12):4178–87.

[21] Fan Yuanliang, Wu Jing, Chen Zitao, Wu Han, Huang Jianye, Liu Binqian. Data-driven state-of-charge estimation of lithium-ion batteries. In: 2020 8th international conference on power electronics systems and applications. PESA, IEEE; 2020, p. 1–5.

[22] Wang Baojin, Liu Zhiyuan, Li Shengbo Eben, Moura Scott Jason, Peng Huei. State-of-charge estimation for lithium-ion batteries based on a nonlinear fractional model. IEEE Trans Control Syst Technol 2016;25(1):3–11.

[23] Tian Jinpeng, Chen Cheng, Shen Weixiang, Sun Fengchun, Xiong Rui. Deep learning framework for lithium-ion battery state of charge estimation: recent advances and future perspectives. Energy Storage Mater 2023;102883.

[24] Venugopal Prakash, Reka S Sofana. State of charge estimation of lithium batteries in electric vehicles using IndRNN. IETE J Res 2023;69(5):2886–96.

[25] Zafar Muhammad Hamza, Mansoor Majad, Abou Houran Mohamad, Khan Noman Mujeeb, Khan Kamran, Moosavi Syed Kumayl Raza, Sanfilippo Filippo. Hybrid deep learning model for efficient state of charge estimation of Li-ion batteries in electric vehicles. Energy 2023;282:128317.

[26] Stroe Daniel-Ioan, Schaltz Erik. Lithium-ion battery state-of-health estimation using the incremental capacity analysis technique. IEEE Trans Ind Appl 2019;56(1):678–85.

[27] Xu Song, Zha Fang-Lin, Huang Bo-Wen, Yu Bing, Huang Hai-Bo, Zhou Ting, Mao Wen-Qi, Wu Jie-Jun, Wei Jia-Qiang, Gong Shang-Kun, et al. Research on the state of health estimation of lithium-ion batteries for energy storage based on XGB-AKF method. Front Energy Res 2023;10:999676.

[28] He Jiangtao, Wei Zhongbao, Bian Xiaolei, Yan Fengjun. State-of-health estimation of lithium-ion batteries using incremental capacity analysis based on voltage–capacity model. IEEE Trans Transp Electrif 2020;6(2):417–26.

[29] Su Kangze, Deng Biao, Tang Shengjin, Sun Xiaoyan, Fang Pengya, Si Xiaosheng, Han Xuebing. Remaining useful life prediction of lithium-ion batteries based on a cubic polynomial degradation model and envelope extraction. Batteries 2023;9(9):441.

[30] Meng Xianmeng, Cai Cuicui, Wang Yueqin, Wang Qijian, Tan Linglong. Remaining useful life prediction of lithium-ion batteries using CEEMDAN and WOA-SVR model. Front Energy Res 2022;10:984991.

[31] Shi Yuanhao, Yang Yanru, Wen Jie, Cui Fangshu, Wang Jingcheng. Remaining useful life prediction for lithium-ion battery based on CEEMDAN and SVR. In: 2020 IEEE 18th international conference on industrial informatics. INDIN, Vol. 1, IEEE; 2020, p. 888–93.

[32] Zhang Lijun, Mu Zhongqiang, Sun Changyan. Remaining useful life prediction for lithium-ion batteries based on exponential model and particle filter. IEEE Access 2018;6:17729–40.

[33] Kannan M, Sundareswaran Kinattingal, Nayak PSrinivas Rao, Simon Sishaj P. A combined DNN-NBEATS architecture for state of charge estimation of lithium-ion batteries in electric vehicles. IEEE Trans Veh Technol 2023.

[34] Zhang Zhaopu, Min Haitao, Guo Hangang, Yu Yuanbin, Sun Weiyi, Jiang Junyu, Zhao Hang. State of health estimation method for lithium-ion batteries using incremental capacity and long short-term memory network. J Energy Storage 2023;64:107063.

[35] von Bülow Friedrich, Wassermann Markus, Meisen Tobias. State of health forecasting of lithium-ion batteries operated in a battery electric vehicle fleet. J Energy Storage 2023;72:108271.

[36] Lee Gyumin, Kwon Daeil, Lee Changyong. A convolutional neural network model for SOH estimation of Li-ion batteries with physical interpretability. Mech Syst Signal Process 2023;188:110004.

[37] Luo Kai, Zheng Huiru, Shi Zhicong. A simple feature extraction method for estimating the whole life cycle state of health of lithium-ion batteries using transformer-based neural network. J Power Sources 2023;576:233139.

[38] Peng Simin, Sun Yunxiang, Liu Dandan, Yu Quanqing, Kan Jiarong, Pecht Michael. State of health estimation of lithium-ion batteries based on multi-health features extraction and improved long short-term memory neural network. Energy 2023;282:128956.

[39] Gu Xinyu, See KW, Li Penghua, Shan Kangheng, Wang Yunpeng, Zhao Liang, Lim Kai Chin, Zhang Neng. A novel state-of-health estimation for the lithium-ion battery using a convolutional neural network and transformer model. Energy 2023;262:125501.

[40] Jorge Inès, Mesbahi Tedjani, Samet Ahmed, Boné Romuald. Time series feature extraction for lithium-ion batteries state-of-health prediction. J Energy Storage 2023;59:106436.

[41] Wang Zhuqing, Liu Ning, Chen Chilian, Guo Yangming. Adaptive self-attention LSTM for RUL prediction of lithium-ion batteries. Inform Sci 2023;635:398–413.

[42] Siruvuri Varma, Budarapu PR, Paggi M. Influence of cracks on fracture strength and electric power losses in Silicon solar cells at high temperatures: deep machine learning and molecular dynamics approach. Appl Phys A 2023;129(6):408.

[43] Wang Zili, Liu Yonglu, Wang Fen, Wang Hui, Su Mei. Capacity and remaining useful life prediction for lithium-ion batteries based on sequence decomposition and a deep-learning network. J Energy Storage 2023;72:108085.

[44] Guo Fei, Wu Xiongwei, Liu Lili, Ye Jilei, Wang Tao, Fu Lijun, Wu Yuping. Prediction of remaining useful life and state of health of lithium batteries based on time series feature and Savitzky-Golay filter combined with gated recurrent unit neural network. Energy 2023;270:126880.

[45] Eleftheriadis Panagiotis, Giazitzis Spyridon, Leva Sonia, Ogliari Emanuele. Data-driven methods for the state of charge estimation of lithium-ion batteries: An overview. Forecasting 2023;5(3):576–99.

[46] Wang Shunli, Fan Yongcun, Jin Siyu, Takyi-Aninakwa Paul, Fernandez Carlos. Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries. Reliab Eng Syst Saf 2023;230:108920.

[47] Fan Guodong, Zhang Xi. Battery capacity estimation using 10-second relaxation voltage and a convolutional neural network. Appl Energy 2023;330:120308.

[48] Wan Zijing, Kang Yilin, Ou Renwei, Xue Song, Xu Dongwei, Luo Xiaobing. Multi-step time series forecasting on the temperature of lithium-ion batteries. J Energy Storage 2023;64:107092.

[49] Ma Bin, Yu Han-Qing, Wang Wen-Tao, Yang Xian-Bin, Zhang Li-Sheng, Xie Hai-Cheng, Zhang Cheng, Chen Si-Yan, Liu Xin-Hua. State of health and remaining useful life prediction for lithium-ion batteries based on differential thermal voltammetry and a long and short memory neural network. Rare Met 2023;42(3):885–901.

[50] Schmitt Jakob, Horstkötter Ivo, Bäker Bernard. Electrical lithium-ion battery models based on recurrent neural networks: A holistic approach. J Energy Storage 2023;58:106461.

[51] Larvaron Benjamin, Clausel Marianne, Bertoncello Antoine, Benjamin Sébastien, Oppenheim Georges. Chained Gaussian processes with derivative information to forecast battery health degradation. J Energy Storage 2023;65:107180.

[52] Huang Zhelin, Xu Fan, Yang Fangfang. State of health prediction of lithium-ion batteries based on autoregression with exogenous variables model. Energy 2023;262:125497.

[53] Abedi Sara, Kwon Soongeol. Rolling-horizon optimization integrated with recurrent neural network-driven forecasting for residential battery energy storage operations. Int J Electr Power Energy Syst 2023;145:108589.

[54] Cui Binghan, Wang Han, Li Renlong, Xiang Lizhi, Du Jiannan, Zhao Huaian, Li Sai, Zhao Xinyue, Yin Geping, Cheng Xinqun, et al. Long-sequence voltage series forecasting for internal short circuit early detection of lithium-ion batteries. Patterns 2023;4(6).

[55] Messing Marvin, Shoa Tina, Habibi Saeid. Estimating battery state of health using electrochemical impedance spectroscopy and the relaxation effect. J Energy Storage 2021;43:103210.

[56] Liu Jialong, Duan Qiangling, Qi Kaixuan, Liu Yujun, Sun Jinhua, Wang Zhirong, Wang Qingsong. Capacity fading mechanisms and state of health prediction of commercial lithium-ion battery in total lifespan. J Energy Storage 2022;46:103910.

[57] Wu Yi, Li Wei, Wang Youren, Zhang Kai. Remaining useful life prediction of lithium-ion batteries using neural network and bat-based particle filter. IEEE Access 2019;7:54843–54.

[58] Naguib Mina, Kollmeyer Phillip, Skells Michael. LG 18650HG2 li-ion battery data. 2020, Mendeley Data, V1.
[59] Birkl Christoph R. Oxford battery degradation dataset 1. 2017.
[60] Lin Chun-Pang, Cabrera Javier, Yang Fangfang, Ling Man-Ho, Tsui Kwok-Leung, Bae Suk-Joo. Battery state of health modeling and remaining useful life prediction through time series model. Appl Energy 2020;275:115338.
[61] Hochreiter Sepp, Schmidhuber Jürgen. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[62] Cho Kyunghyun, Van Merriënboer Bart, Gulcehre Caglar, Bahdanau Dzmitry, Bougares Fethi, Schwenk Holger, Bengio Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014, arXiv preprint arXiv:1406.1078.
[63] Chung Junyoung, Gulcehre Caglar, Cho KyungHyun, Bengio Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014, arXiv preprint arXiv:1412.3555.
[64] Graves Alex. Generating sequences with recurrent neural networks. 2013, arXiv preprint arXiv:1308.0850.
[65] Bahdanau Dzmitry, Cho Kyunghyun, Bengio Yoshua. Neural machine translation by jointly learning to align and translate. 2014, arXiv preprint arXiv:1409.0473.
[66] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, Kaiser Łukasz, Polosukhin Illia. Attention is all you need. Adv Neural Inf Process Syst 2017;30.
[67] Dumoulin Vincent, Visin Francesco. A guide to convolution arithmetic for deep learning. 2016, arXiv preprint arXiv:1603.07285.
[68] Mallat Stéphane. Understanding deep convolutional networks. Phil Trans R Soc A 2016;374(2065):20150203.
[69] Oreshkin Boris N, Carpov Dmitri, Chapados Nicolas, Bengio Yoshua. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. 2019, arXiv preprint arXiv:1905.10437.
[70] Lim Bryan, Arık Sercan Ö, Loeff Nicolas, Pfister Tomas. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast 2021;37(4):1748–64.
[71] Schaffer Andrea L, Dobbins Timothy A, Pearson Sallie-Anne. Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. BMC Med Res Methodol 2021;21(1):1–12.
[72] Dauphin Yann N, Fan Angela, Auli Michael, Grangier David. Language modeling with gated convolutional networks. In: International conference on machine learning. PMLR; 2017, p. 933–41.
[73] Kassani Peyman H, Lu Fred, Le Guen Yann, Belloy Michael E, He Zihuai. Deep neural networks with controlled variable selection for the identification of putative causal genetic variants. Nat Mach Intell 2022;4(9):761–71.
[74] Castellano Giovanna, Fanelli Anna Maria. Variable selection using neural-network models. Neurocomputing 2000;31(1–4):1–13.
[75] De Livera Alysha M, Hyndman Rob J, Snyder Ralph D. Forecasting time series with complex seasonal patterns using exponential smoothing. J Amer Statist Assoc 2011;106(496):1513–27.