

Machine learning-based state of charge estimation: A comparison between CatBoost model and C-BLSTM-AE model

Abderrahim Zilali^{a,*}, Mehdi Adda^a, Khaled Ziane^b, Maxime Berger^a

^a Département de mathématiques, informatique et génie, Université du Québec à Rimouski, Rimouski, QC G5G 3A1, Canada

^b Centre de Recherche et d'Innovation en Intelligence Énergétique CR2Ie, 175 Rue de la Vérendrye, Sept-Îles, QC G4R 5B7, Canada



ARTICLE INFO

Keywords:
 SOC estimation
 Machine learning
 Deep learning
 Auto-Encoder
 Gradient Boosting

ABSTRACT

The State of Charge (SOC) is a key metric within a Lithium-ion battery management system (BMS). Accurate SOC estimation is essential for enhancing battery longevity and ensuring user safety, making it a critical component of an effective BMS. Although SOC estimation has become an active research area for the machine learning (ML) community, only a handful of works have considered its estimation at negative temperatures. This paper proposes the application of two machine learning-based approaches for SOC estimation that perform well at wide range of temperatures (positive and negative) and varying dynamic loads. The first one is a hybrid deep learning approach based on the Convolutional BLSTM Auto-Encoder (C-BLSTM-AE) model that relies on extracting abstract features from input data. The second one is a CatBoost model that leverages the gradient boosting technique to enhance the prediction made by its constituent trees. The performance of the models is evaluated by comparing their regression accuracy and computational resource utilization. The C-BLSTM-AE model achieves a low Mean Absolute Error (MAE) of **0.52 %** under fixed ambient temperature conditions and maintains a MAE of **1.03 %** for variable ambient temperatures. The CatBoost model achieves a MAE of **0.69 %** with fixed temperature settings and a MAE of **1.09 %** under variable temperature conditions.

1. Introduction

Lithium-ion (Li-ion) batteries are becoming increasingly prevalent in the automotive industry because of their high energy density and minimal environmental impact. Particularly in electric vehicles (EVs), their demand increased by about 65 % between 2021 and 2022 (Busch et al., 2024). A battery management system (BMS) is required to ensure effective integration of the battery pack; it has several responsibilities, such as thermal management, ensuring cell balancing and battery equalization, and estimating battery states, namely State of Health (SOH) and State of Charge (SOC) (Cao & Emadi, 2011). SOC is essential for the reliable operation of the electric drive system since this quantity directly gauges a vehicle's remaining driving range, and is crucial for enhancing battery safety and preventing irregular charging and discharging (An et al., 2021). SOC is defined as the ratio between the remaining available capacity and the nominal capacity, which can be written as (Hu et al., 2022).

$$SOC_t = \frac{C_t}{C_n} \quad (1)$$

where SOC_t is the SOC value at time t , C_t is the instantaneous capacity and C_n is the nominal capacity. Dynamic SOC estimation is inherently challenging since SOC cannot be directly measured. Therefore, its estimation uses measurable quantities such as battery voltage and current, temperature, and other variables. For practical applications, several methods are used for SOC estimation (How et al., 2019). The Coulomb Counting method also referred to as the Ampere-hour counting method, is the most basic approach for estimating the SOC of batteries (Zhang et al., 2020). According to Fan et al. (2022), the Coulomb counting method is by far the most extensively used for SOC estimation due to its small computational requirements, which makes it easy to implement on a BMS chip with low processing power. It involves integrating the battery current to ascertain the stored charge, which is then divided by the battery's total charge capacity to determine the SOC (Ng et al., 2009) as described in 2.

$$SOC_t = SOC_0 - \int_0^t \eta I(t) dt \Big/ C_n \quad (2)$$

* Corresponding author.

E-mail addresses: abderrahim.zilali@uqr.ca (A. Zilali), mehdi.adda@uqr.ca (M. Adda), khaled.ziane@cegepsi.ca (K. Ziane), maxime.berger@uqr.ca (M. Berger).

where SOC_0 is the initial SOC value and I is the electric current at time t . The coulombic efficiency η represents the ratio of consumed electrons and available electrons during charging and discharging, which is assumed to be 0.9 and 1.0 during charging and discharging, respectively. While implementing Coulomb Counting for dynamic SOC estimation is relatively straightforward, it faces accuracy limitations due to errors in current integration (Sepasi et al., 2014). The accuracy of the Coulomb Counting integration method is also influenced by the initial SOC, and factors such as temperature drift and noise in the current sensor can result in cumulative errors throughout the integration process (Fan et al., 2022).

Another method commonly employed is the Open Circuit Voltage (OCV) method. The OCV method is based on a lookup table directly mapping the OCV of a battery with its SOC. The OCV uses the stable battery electromotive force in the open circuit state and SOC relationship to estimate the SOC value (Zheng et al., 2018). This estimation method relies on an approximate linear relationship between SOC and OCV. However, this correlation is inconsistent across different chemistries due to limitations such as variations in capacity and electrode material, and some chemistries, such as lithium-based, do not exhibit such a relationship (Hannan et al., 2017). Therefore, this approach is of limited use to estimate the SOC of Li-ion batteries.

Kalman Filter (KF)-based method is another widely used estimation technique. The main principle of the KF-based SOC estimation method is to relate the measured battery signals (voltage, current and temperature) with the battery model. The first step is to estimate a pre-determined SOC using the Coulomb Counting method. This calculation requires the initial SOC obtained from the SOC-OCV relationship. The second step involves estimating the model voltage using the chosen battery model. The parameters for this model can be determined based on the measured voltage, current, and temperature using various Parameter Identification Methods (PIMs) (Wang et al., 2016). The final step is to update the Kalman gain by comparing the model voltage with the measured voltage and using the voltage error to adjust the Kalman gain. The updated gain is then used to calculate the estimated SOC using a KF algorithm (Shrivastava et al., 2019). Due to its high accuracy and self-correcting features, KF-based SOC estimation is one of the preferred methods for online SOC estimation. However, the performance of KF-based SOC estimation heavily relies on the accuracy of the battery equivalent circuit model and the measurement covariance information (Xia et al., 2015).

The limitations in current SOC estimation methods have motivated the battery community to investigate more efficient and accurate techniques. With the rapid advancement of artificial intelligence, machine learning-based SOC estimation has become increasingly popular (Ng et al., 2020), leading to the development of various estimation techniques. However, only a few studies have addressed SOC estimation at negative temperatures, leaving an area of research that still needs further exploration. This paper proposes and validates the performance of two machine learning-based approaches for SOC estimation at both positive and negative temperatures, and under varying dynamic loads.

The remainder of the paper is organized as follows. Section II provides the background and literature review, offering an in-depth examination of existing research and foundational concepts related to machine learning-based SOC estimation for Li-ion batteries in EVs. Section III presents the methodology and details the proposed models and the rationale behind their selection. It also elaborates on the dataset, discussing its attributes, sources, and the preprocessing steps to prepare the data for model training. Section IV outlines the experimental setup and training process, including the configurations, tools, and procedures used to implement and evaluate the machine learning models. Section V presents the models' evaluation, showcasing the results, comparative analysis, and discussions on the effectiveness of the applied techniques regarding regression accuracy and computational resource utilization. Finally, section VI concludes the paper by summarizing the key findings

and their implications for SOC estimation.

2. Background and literature review

Machine learning (ML) is a paradigm that enables systems to learn from data without being explicitly programmed. This section presents the foundational concepts of the main machine learning algorithms used for SOC estimation. First, key definitions are presented, and then a review of existing machine learning-based methods is provided.

2.1. Dense neural network

A Dense Neural Network (DNN) is a machine learning model characterized by its multiple layers of interconnected neurons, designed to capture complex patterns in data. Each neuron in a DNN receives inputs from all neurons in the previous layer and performs nonlinear calculations (Bengio et al., 2017). By stacking multiple layers, DNNs can learn increasingly abstract representations of the input data, enabling them to identify high-level patterns that are not immediately apparent in the raw data (LeCun et al., 2015).

2.2. Convolutional neural network

A convolutional neural network (CNN) is a type of artificial neural network comprising, in most cases, of a convolutional layer and a pooling layer (Chollet, 2021; Li et al., 2021). The convolution layer contains many convolution filters, each acting as a template that determines whether a particular local feature is present in the input tensor (Pang et al., 2017). A convolution filter relies on a straightforward operation called a convolution, where matrix elements are repeatedly multiplied, transforming the input data into a feature map. The feature map is then passed to a pooling layer that provides a way to condense a large matrix into a smaller summary vector (James et al., 2013). CNN demonstrated outstanding results in computer vision and large-scale models due to their ability to extract complex features.

2.3. Long short-term memory

Long Short-Term Memory (LSTM) is a type of artificial neural network capable of capturing the spatiotemporal meaning within data (Jain et al., 2016). They have succeeded on many end-to-end learning tasks, spatially in Natural Language Processing (NLP). LSTMs address the vanishing and exploding gradient problems that hinder traditional recurrent neural networks (RNNs) in processing long sequences (Pascanu et al., 2013). One key advantage of LSTMs is their ability to store values over both short and long time intervals (Yu et al., 2019).

2.4. Bidirectional long short-term memory

Bidirectional Long Short-Term Memory (BLSTM) networks are an advanced type of RNN that enhances the standard LSTM architecture. Unlike traditional LSTMs, which only consider prior information in the sequence, BLSTMs process data in both forward and backward directions, which enables them to capture context from both past and future states (Schuster & Paliwal, 1997). BLSTMs have also succeeded in NLP tasks. For instance, they are widely used in named entity recognition, machine translation, and sentiment analysis, where understanding the surrounding context is crucial (Huang et al., 2015).

2.5. Auto-Encoder

An Auto-Encoder (AE) is a specific type of neural network which is mainly designed to encode the input into a compressed and meaningful representation and then decode it back such that the reconstructed input is as similar as possible to the original one (Bank et al., 2020). AE is a kind of signal compression model based on a neural network; therefore it

Table 1
Performance comparison with literature methods.

Method	Lower Error (MAE)	Temperatures (°C)
Transformer (Hannan et al., 2021) *	0.44 %	-20, -10, 0, 10, 25, 40
Proposed (C-BLSTM-AE)	0.52 %	-20, -10, 0, 10, 25, 40
LSTM-RNN (Chemali et al., 2017)	0.57 %	0, 10, 25
DNN (Chemali et al., 2018)	0.61 %	-20, -10, 0, 10, 25
TCN (Liu et al., 2021)	0.67 %	0, 10, 25
Proposed (CatBoost)	0.69 %	-20, -10, 0, 10, 25 40
LSTM (Wong et al., 2021)	1.17 %	25

* This paper used two distinct datasets from the LG *LiNiMnCoO₂* and Panasonic *LiNoCoAlO₂* cells.

is used in most cases for feature extraction (Yan & Han, 2018).

2.6. Temporal convolutional network

A Temporal Convolutional Network (TCN) is a neural network for sequence modeling tasks. It leverages causal and dilated convolutions to capture long-range dependencies in sequential data, ensuring that the model predictions at any time step depend only on past inputs, not future ones (Lea et al., 2017). TCN is known for their ability to handle long sequences efficiently, maintain a stable memory footprint, and avoid issues related to vanishing gradients, which are common in RNNs (Bai et al., 2018).

2.7. Transformers

A transformer network is a type of deep learning model primarily designed for handling sequential data, making it highly effective for tasks such as time series forecasting and signal processing (Benidis et al., 2022). Unlike RNNs that process data sequentially, transformers leverage a mechanism called self-attention (Vaswani et al., 2017), which allows them to simultaneously process and attend to all elements of a sequence. This parallel processing capability not only accelerates computation but also enables the model to capture long-range dependencies more effectively (Rae et al., 2019).

2.8. Gradient boosting

Gradient boosting is an ensemble learning technique that builds a predictive model in an iterative, stage-wise manner (Freund et al., 1999). The main idea is to construct the model incrementally by adding weak learners, typically decision trees, to correct errors made by previous models. The process starts with a simple model and iteratively adds new models that aim to minimize a specified loss function through gradient descent. This method has proven highly effective and is at the core of many successful algorithms known for their state-of-the-art performance on tabular data.

2.9. Machine learning-based SOC estimation

In the last decade, machine learning has grown rapidly, revolutionizing numerous fields and redefining the state of the art in many domains. SOC estimation has withdrawn the attention of the ML community, and since then, several techniques have been proposed.

In Chemali et al. (2018), a DNN is used to estimate a battery SOC based on the input vector $[V_t, T_t, Vavg_t, Iavg_t]$ encompassing instantaneous voltage and battery temperature as well as averaged voltage and averaged current. The model achieved a Mean Absolute Error (MAE) of 1.10 % for a 25 °C dataset and a MAE of 2.17 % for a -20 °C dataset. In Vidal et al. (2020), the authors instead proposed a Feedforward Neural Network (FNN) to estimate SOC and showed that this model is capable of estimating the battery SOC with accuracy under 1 % MAE. Because of

the limitation of a DNN to process sequence data, including $Vavg_t$ and $Iavg_t$ helps to capture the underlying trend in the averaged sequence, which translates into better estimation accuracy.

Other works focused on the premise of CNN to estimate SOC. (Bhattacharjee et al., 2021) proposed a 1D CNN-based estimator in conjunction with a transfer learning mechanism. The proposed algorithm generalized better and succeeded in estimating SOC for batteries with different chemical characteristics than the one used for training the model. Additionally, the presented framework enabled the CNN model to learn effectively with a significantly reduced amount of battery data. (Fan et al., 2022) proposed a U-Net architecture that only inputs voltage and current sequences to estimate SOC under dynamic temperature conditions. The proposed method can estimate the SOC at constant temperatures with a MAE within 1.1 % and RMSE of 1.4 %. For varying temperatures, the model achieved a MAE within 1.5 % and RMSE within 1.8 %. (Hannan et al., 2020) implemented a learning rate strategy to create a SOC estimation model using a Full Convolutional Network (FCN). The model achieved a RMSE of 0.85 % and a MAE of 0.7 % at 25 °C, and 2.0 % RMSE and 1.55 % MAE across ambient temperatures ranging from -20 °C to 25 °C.

Since the battery-measured signals are time series data, the use of LSTM has also received attention for SOC estimation. (Chemali et al., 2017) proposed a LSTM-RNN model that achieved a low MAE of 0.57 % at fixed ambient temperature and a MAE of 1.6 % on a dataset with ambient temperature ranging between 10 °C and 25 °C. The network model extracts high-level features from the input sequence comprising voltage, current, and battery surface temperature. In Wong et al. (2021) the authors estimated Li-ion battery SOC by using the measured voltage, current, and temperature values, with 1.17 % MAE and 1.57 % RMSE.

Many works have also focused on using TCN for SOC modeling tasks. (Yahia et al., 2023) used $[V_t, I_t, T_t, Vavg_t, Iavg_t]$ to train a TCN for SOC estimation. The proposed method achieved results comparable to those of RNN and CNN. (Liu et al., 2021) used TCN along with transfer learning; the method demonstrated effectiveness for SOC estimation and achieved an accuracy of 0.67 % MAE. The study promoted the application of transfer learning, which facilitates knowledge transfer between different batteries and requires only a small amount of battery data.

The latest contributions in the SOC estimation task focused on using transformer-based networks. Transformers have revolutionized the field of machine learning by providing state-of-the-art performance in various applications (Islam et al., 2023), owing to their ability to learn contextual relationships within data without relying on the sequential nature of RNN. In Hannan et al. (2021), the authors proposed a deep learning transformer model, using a Self-Supervised-Learning (SSL) framework, that achieves very low errors: RMSE of 0.90 % and MAE of 0.44 % at a constant ambient temperature, and RMSE of 1.19 % and MAE of 0.7 % at varying ambient temperatures. Moreover, the model can be effectively trained with only 5 epochs and 20 % of the total training data and enables transferability to a new Li-ion cell with different chemistry, maintaining performance comparable to models trained from scratch.

The literature review reveals that only a few studies have focused on SOC estimation at negative temperatures. This work addresses that gap by providing ML-based estimation models at -10 °C and -20 °C under various load conditions, offering new insights into SOC behavior in sub-zero conditions. Table 1 compares the proposed estimation performance and other methods from the literature that utilized the same dataset.

3. Methodology

This section outlines the framework adopted for SOC estimation, the mathematical formulation of the approach, and the methods employed. We propose estimation models at different temperature settings and varying loads.

To obtain accurate estimation results, two methods are presented: one that deals with the problem considering the sequential aspect of the input data and relies on deep learning to predict the SOC and another

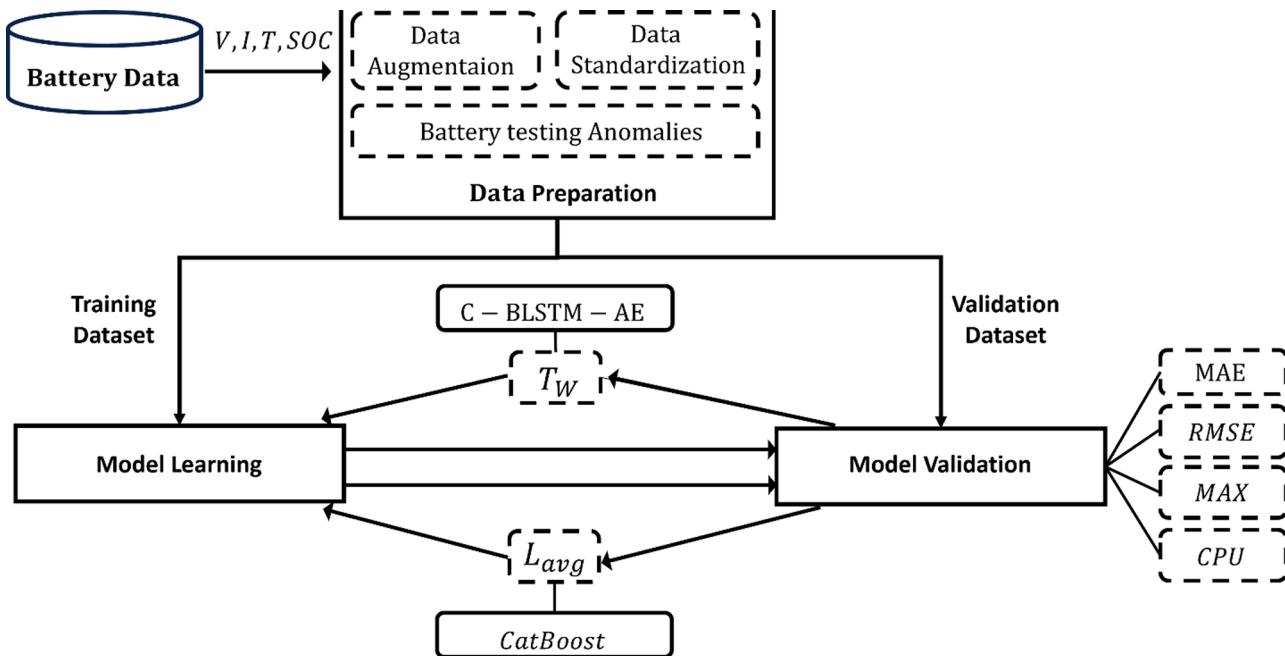


Fig. 1. The adopted workflow for SOC estimation.

Table 2
Cell specifications.

Type	LG 18650HG2
Chemistry	Li [NiMnCo]O2 (H—NMC) / Graphite + SiO
Nominal capacity	3 Ah
Nominal Voltage	3.6 V
Energy Density	240 (Wh/Kg)

that relies traditional machine learning and considers the tabular aspect of the data. Models are proposed for predicting the SOC at constant ambient temperatures and at varying temperatures. The models are presented for each of the ambient temperatures, i.e. 25 °C, 40 °C, 10 °C, 0 °C, -10 °C and -20 °C and models for 25 °C, 40 °C and 10 °C combined referred to as AllPositives model and finally AllNegatives model that combines 0 °C and -10 °C and 20 °C data.

The C-BLSTM-AE method estimates the SOC by leveraging historical battery measurements, i.e. voltage, current, and temperature. This method uses a segment of length m of history battery signals to estimate a point SOC. This estimation method can be described as:

$$\widehat{SOC}_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-(m-1)}) \quad (3)$$

where $x_t = [V_t, I_t, T_t]$ represents the battery measurement, i.e. current, voltage and temperature.

The results of this deep learning method are compared with a traditional machine learning method based on CatBoost learning algorithm for estimating the SOC of the battery. The model can be formulated as a direct mapping between the measured quantities and the corresponding SOC:

$$\widehat{SOC}_t = f([V_t, I_t, T_t, Vavg_t, Iavg_t]) \quad (4)$$

where $Vavg_t$, $Iavg_t$, and T_t represent the measured voltage, current, and temperature of the battery at time step t , respectively.

The procedure adopted for this work can be summarized into six main steps, as depicted in Fig. 1. The workflow includes the following processes: data description and exploration, data preprocessing which includes cleaning the dataset from missingness and testing anomalies, data augmentation, data standardization and finally model building and

evaluation.

3.1. Dataset description and exploration

To validate our models, raw data from a brand-new cylindrical 18,650 LiNiMnCoO2 cell by LG provided by McMaster University in Hamilton, Ontario, Canada (Vidal et al., 2020) are used. The drive cycle power profiles are for an electric Fiat 500e Hatchback vehicle with an energy capacity of 42.0 kWh. The battery consists of 192 cells of type LG 18650HG2. The battery is scaled for a smaller pack of 8 kWh to conduct the tests. To simulate the behavior changes of the cell under actual driving conditions, data was gathered by exposing the battery cell to various load profiles from velocity profiles of driving cycles used for fuel economy measurements, namely HWFET, UDDS, US06, and LA92, at different ambient temperatures (25 °C, 40 °C, 10 °C, 0 °C, -10 °C, and -20 °C). To mimic real-world driving dynamics, the cell was also subjected to a random combination of the standard drive cycles (Geetha & Subramani, 2019). The specification of the cell is given in Table 2.

The dataset comprises various measured variables for analyzing battery behavior over time. Key columns include Time, and Voltage (measured directly at the battery terminal), and Current (in Amps). It also records Ah (instant capacity in amp-hours, reset after each cycle), Wh (energy in watt-hours), and Power (in watts). Additionally, to keep track of the thermal conditions of the battery, the Temperature column tracks the battery's surface temperature in degrees Celsius. Fig. 2 depicts the dynamic aspects of discharge phase during UDDS drive cycle at ambient temperatures of 25 °C and 0 °C.

3.2. Data preprocessing

3.2.1. Handling outliers and missingness

To our knowledge, no documented work regarding the preprocessing of this dataset exists. This section includes cleaning anomalies such as missing values, outliers, and non-informative segments. Business filters were used for outlier detection for each sensor. The outlier variables were subsequently replaced with missing values. Missing value imputation was done using the nearest neighbor method. The dataset does not suffer severe missingness; only a handful of cases have been found.

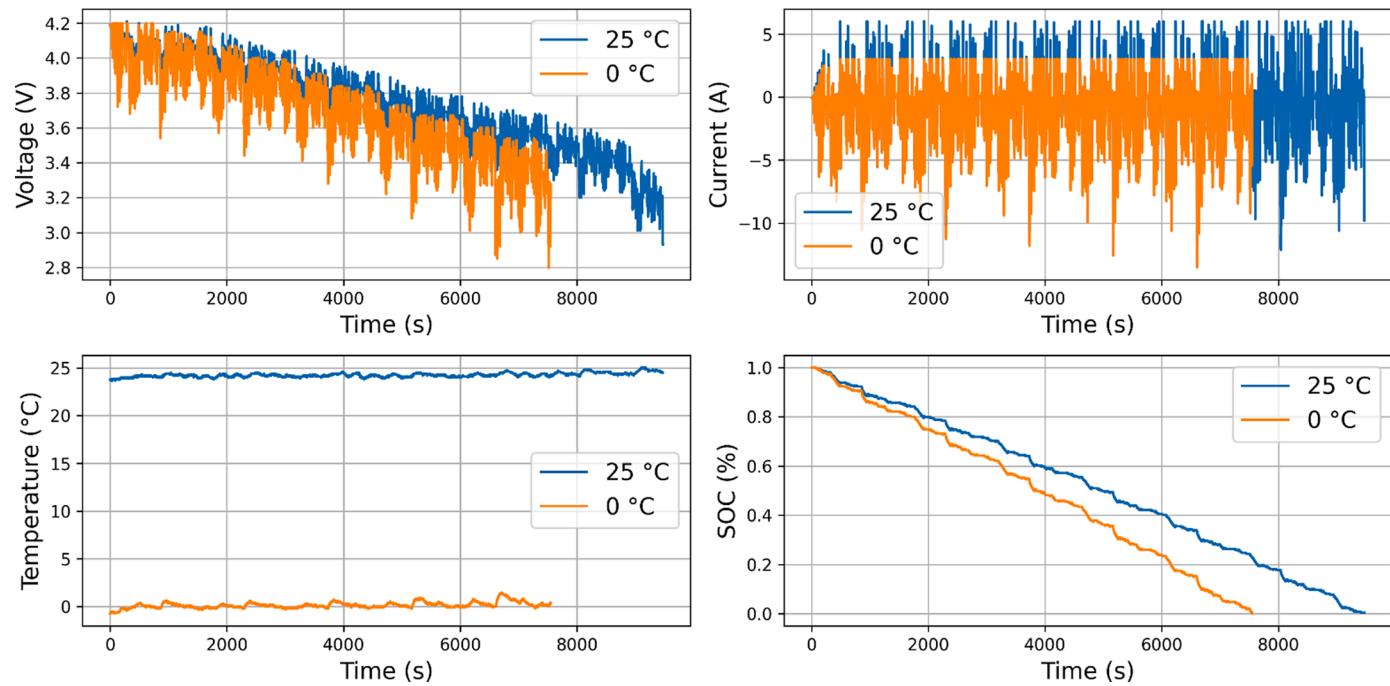


Fig. 2. Battery measurements during UDDS cycle at ambient temperatures of 25 °C and 0 °C.

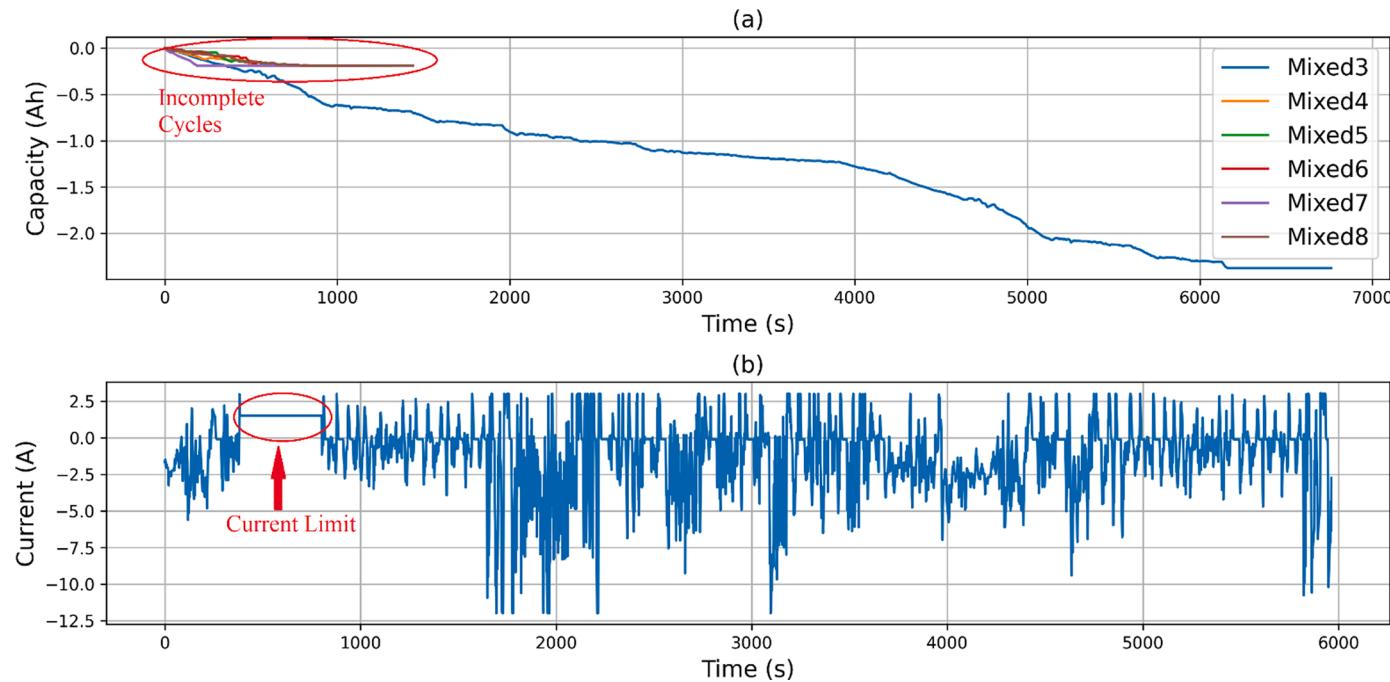


Fig. 3. Testing Anomalies, (a): Incomplete test cycles. (b): Current limit.

3.2.2. Handling testing anomalies

During the experiments, certain discharge records remained incomplete, yet they were included in the dataset accessible to the public. Specifically, at -40°C , during the mixed4, mixed5, mixed6, mixed7, and mixed8 cycles, the battery was not fully charged at the beginning of each cycle leading to incomplete cycles as illustrated in Fig. 3.a. Consequently, these files were excluded and were not utilized in the subsequent phases. Unlike in laboratory test conditions where a constant current (CC) discharging is performed, in EVs, the load for the traction battery is usually very unsteady during driving (Sarasketa-Zabala et al., 2015) because of the acceleration and deceleration of the vehicle (Keil &

Jossen, 2017). Moreover, due to regenerative braking, driving does not entail a straightforward discharging process, which results in repeated charging periods. For -10°C experiments, a current limit was set to prevent premature aging caused by the high charging currents (Vidal et al., 2020), which introduced sequences of constant current measurements, as highlighted in Fig. 3.b, which alters the underlying trend in the data. These sequences were deleted to keep the consistency of the trends in the measurements.

3.2.3. Data augmentation

To ensure that the learning algorithms are not impacted by a lack of

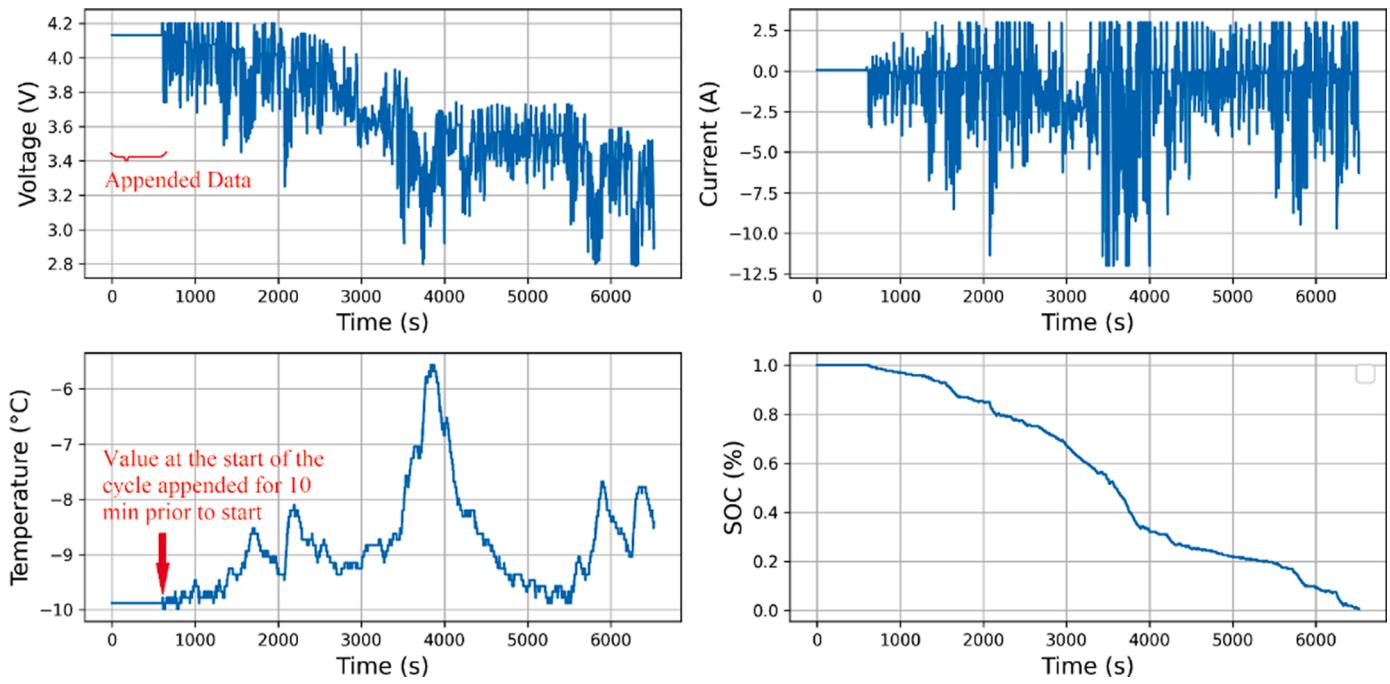


Fig. 4. Features augmentation.

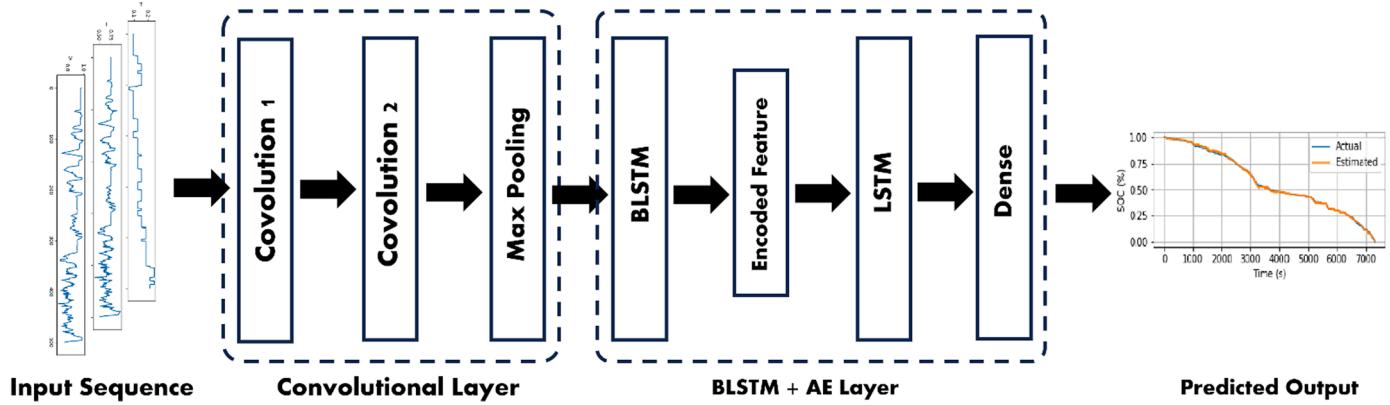


Fig. 5. C-BLSTM-AE architecture.

initial data, the dataset is augmented by appending the first readings of the vector $[V_0, I_0, T_0, SOC_0]$ at the beginning of each cycle as described in Fig. 4. This augmentation serves two key purposes: 1) Preservation of data, by preventing the exclusion of valuable initial data points ensuring that the entire cycle of data is utilized effectively, and that no potentially useful information is discarded, and 2) improved estimation by enabling the estimation of the earliest SOC values that fall below the upper limit of the time window. This is crucial for accurately capturing the initial behavior and trends of the cycle at high SOC values, providing a more comprehensive and precise understanding of the SOC behavior from the very beginning of the cycle. In essence, this data augmentation technique maximizes the use of available data. It also enhances the accuracy of early SOC estimations, thereby improving the overall performance and reliability of the learning algorithms.

3.2.4. Standardization

Since the input features vary across different ranges, standardization enhances stability in the optimization process, facilitating convergence during gradient-based training. It alleviates problems associated with vanishing and exploding gradients, enabling models to reach optimal

solutions efficiently. The minimum-maximum normalization method is employed to standardize all input features to a standard scale using Eq. (5).

$$X_{\text{normalized}} = (X - X_{\min}) / (X_{\max} - X_{\min}) \quad (5)$$

where $X_{\text{normalized}}$ is the value of the variable after normalization, X_{\min} and X_{\max} are, respectively the minimum and the maximum values of the variable.

3.2.5. Deep learning approach: C-BLSTM-AE

In this section, we introduce the proposed hybrid deep learning approach C-BLSTM-AE. As depicted in Fig. 5, the CNN block scans through the input sequence and extracts significant features from the data, and the resulting feature map is fed to a BLSTM block that learns the temporal dependencies among the sequence. The result of the two blocks is an encoded representation that preserves the spatial-temporal characteristics of the input sequence. An LSTM block then decodes the latent representation using an LSTM unit at the decoder, which renders the architecture less complex compared to a BLSTM used at the decoder

Algorithm 1: Pseudocode for Building a tree in CatBoost Algorithm

```

Input:  $M, \{(x_i, y_i)\}_{i=1}^n, \alpha, L, \{\sigma_i\}_{i=1}^n, Mode$ 
Output:  $T, M$ 
 $grad \leftarrow CalcGradient(L, M, y)$ 
 $r \leftarrow random(1, s)$ 
if  $Mode = Plain$  then
  |  $G \leftarrow (grad_i)_{i=1..n}$ 
else
  |  $G \leftarrow (grad_{\sigma_r(i)})_{i=1..n}$ 
end
 $T \leftarrow emptytree$ 
foreach step of top-down procedure do
  foreach candidate split  $c$  do
    |  $T_c \leftarrow addsplit(c, T)$ 
    | if  $Mode = Plain$  then
      | |  $\Delta(i) \leftarrow avg(grad_p(i))$  for  $p : leaf_f(p) = leaf_f(i)$ , for  $i = 1..n$ 
    | | else
    | | |  $\Delta(i) \leftarrow avg(grad_{\sigma_r(i)})$  for  $p : leaf_f'(p) = leaf_f(i), \sigma_r(p) < \sigma_r(i)$ , for  $i = 1..n$ 
    | | end
    | |  $loss(T_c) \leftarrow cos(\Delta, G)$ 
  | end
  |  $T \leftarrow arg\ min_{T_c}(loss(T_c))$ 
end
if  $Mode = Plain$  then
  | |  $M_{f'}(j) \leftarrow M_{f'}(j) - \alpha \cdot avg(grad_p(j))$  for  $p : leaf_f(p) = leaf_f(j)$  for  $j' = 1..s, j = 1..n$ 
else
  | | |  $M_{f', \sigma_j(i)} \leftarrow M_{f', \sigma_j(i)} - \alpha \cdot avg(grad_{f', j}(p))$  for  $p : leaf_f'(p) = leaf_f(i), \sigma_r(p) \leq j$  for
  | | |  $j' = 1..s, i = 1..n, j \geq \sigma_r(i) - 1$ 
end
return  $T, M$ 
  
```

Fig. 6. Pseudocode for CatBoost Algorithm.

Table 3
Optimal hyperparameters for DL algorithm by temperature condition.

Temperature (° C)	Learning Rate	Batch Size
25	1e-04	128
40	1e-04	256
10	1e-05	128
0	1e-04	128
-10	1e-05	128
-20	1e-05	128
AllPositives	1e-05	256
AllNegatives	1e-05	256

Table 4
Optimal hyperparameters for CatBoost algorithm by temperature condition.

Temperature (° C)	No. of Trees	Learning Rate
25	1000	0.05
40	1000	0.05
10	1000	0.05
0	1000	0.1
-10	1000	0.1
-20	1000	0.1
AllPositives	1500	0.1
AllNegatives	1500	0.1

stage of the network (Jogunola et al., 2022). Consequently, the encoded information from the BLSTM-AE output is decoded by the single layer of LSTM-AE before being fed into two fully connected layers for the final prediction.

3.2.6. Machine learning approach: CatBoost algorithm

Along with the proposed deep learning approach, a machine learning approach that utilizes CatBoost algorithm for making SOC predictions is proposed. CatBoost is known for its efficiency and high predictive power on tabular data. The algorithm starts by using a simple decision tree that makes predictions on the dataset and leverages the gradient boosting

technique to correct the prediction mistakes. It iteratively increases the number of trees used, each newly created tree focusing on correcting the errors made by the previous ones. This ensemble of trees works together to refine the prediction accuracy, leading to a robust model that can effectively handle the complex relationships between the input vector and the corresponding SOC. The final model aggregates all the trees, where the combined predictions yield a more accurate SOC estimation. The process of tree building is presented in Fig. 6.

4. Experimental setup and training

For training, mean absolute error (MSE) defined by 6, was chosen as the overall loss function evaluated during the training phase of the models.

$$MSE = \frac{1}{N} \sum_{n=1}^N (\widehat{SOC}_n - SOC_n)^2 \quad (6)$$

where N is the length of the data and SOC is the true value while and \widehat{SOC} is the output of the proposed network at time n.

To assess the goodness of fit of the models, the performance of the proposed models on testing cycles is evaluated using the root mean square error (RMSE) defined in Eq. (7), and the MAE, as defined in Eq. (8) and mean absolute maximal error defined as the highest error observed in the predictions as depicted in Eq. (9).

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (\widehat{SOC}_n - SOC_n)^2} \quad (7)$$

$$MAE = \frac{1}{N} \sum_{n=1}^N (|\widehat{SOC}_n - SOC_n|) \quad (8)$$

$$MAX = \max_N (|\widehat{SOC}_n + SOC_n|) \quad (9)$$

TensorFlow is used for DL model implementation. TensorFlow is an open-source deep learning software library for defining, training and

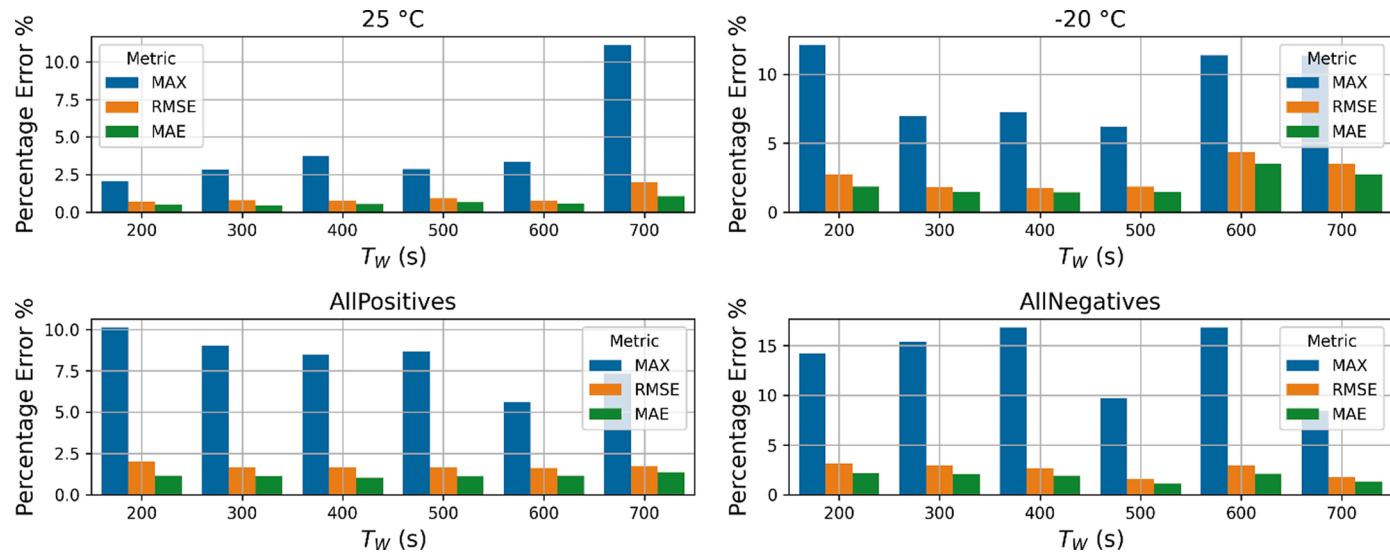


Fig. 7. C-BLSTM-AE estimations errors at different T_W for each different temperature condition.

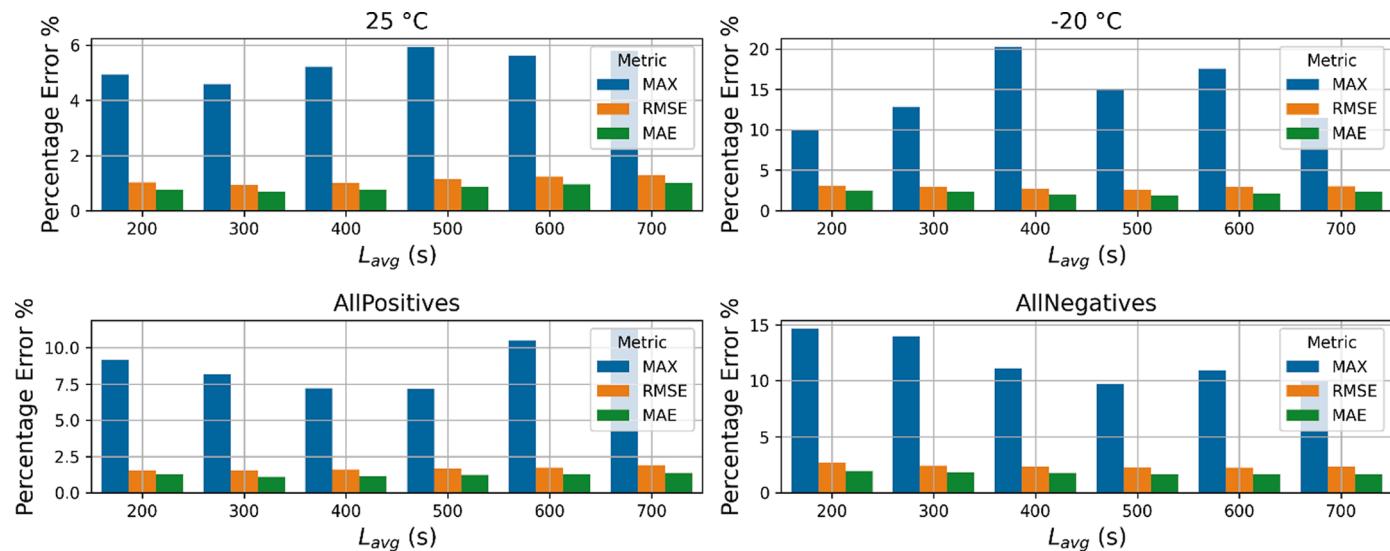


Fig. 8. CatBoost estimations errors at different L_{avg} for each temperature condition.

Table 5
Optimal time window lengths for each method broken by temperature condition.

Temperature ($^{\circ}\text{C}$)	Time Window Length	
	C-BLSTM-AE (T_W)	CatBoost (L_{avg})
25	300	300
40	300	300
10	300	500
0	400	600
-10	500	600
-20	500	500
AllPositives	400	300
AllNegatives	500	600

deploying machine learning models (Abadi et al., 2016). For training C-BLSTM-AE model, ReLU activation function is employed to activate the final neuron responsible for the prediction, primarily due to its positivity, which accommodates our use case where the predicted quantity is constrained between 0.0 and 1.0. Additionally, the ReLU

Table 6
C-BLSTM-AE Estimation Results.

Temperature ($^{\circ}\text{C}$)	MAE	RMSE	MAX	CPU time (s)
25	0.55	0.79	2.93	35,415
40	0.52	0.69	2.14	17,324
10	1.44	1.88	7.00	32,718
0	0.78	1.05	3.50	32,507
-10	0.80	1.11	5.85	63,856
-20	1.31	1.79	7.54	28,615
AllPositives	1.03	1.64	8.48	102,733
AllNegatives	1.13	1.58	9.69	141,274

activation function exhibits reduced sensitivity to random initialization, ensuring stability (Schmidt-Hieber, 2020). The model is trained for 400 epochs and, an early stopping mechanism is implemented, this helps prevent overfitting by halting training if there is no improvement in the loss function. To optimize the loss during training, we utilize the Adam optimizer (Zhang, 2018), which iteratively adjusts the network's parameters (weights and biases) based on the gradient of the loss function. The learning rate is set between 10e-05 and 10e-04 and batch size is

Table 7
CatBoost Estimation Results.

Temperature (° C)	MAE	RMSE	MAX	CPU time (s)
25	0.69	0.92	4.59	20
40	1.00	1.45	6.77	14
10	1.54	1.97	6.98	19
0	1.11	1.55	8.17	13
-10	1.33	1.81	8.41	14
-20	1.90	2.56	15.06	13
AllPositives	1.09	1.54	8.17	31
AllNegatives	1.64	2.23	10.89	24

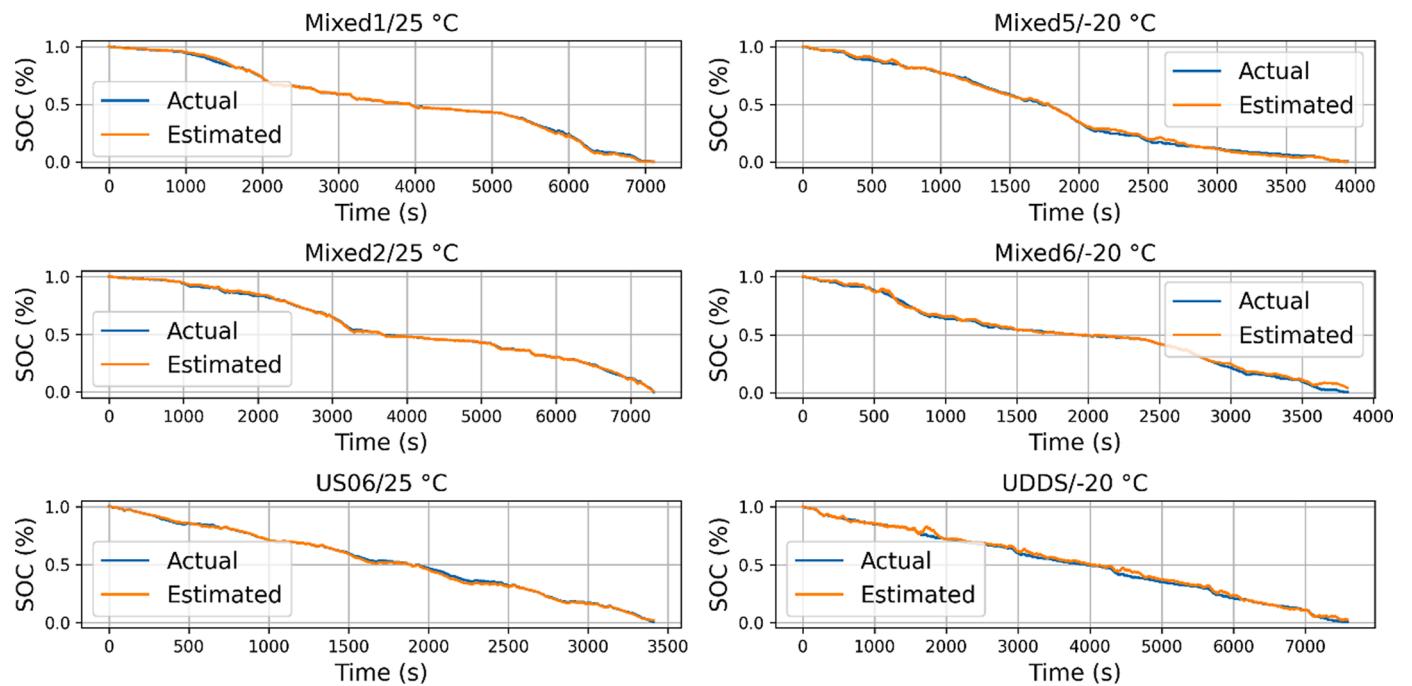


Fig. 9. C-BLSTM-AE estimation results for single ambient temperature.

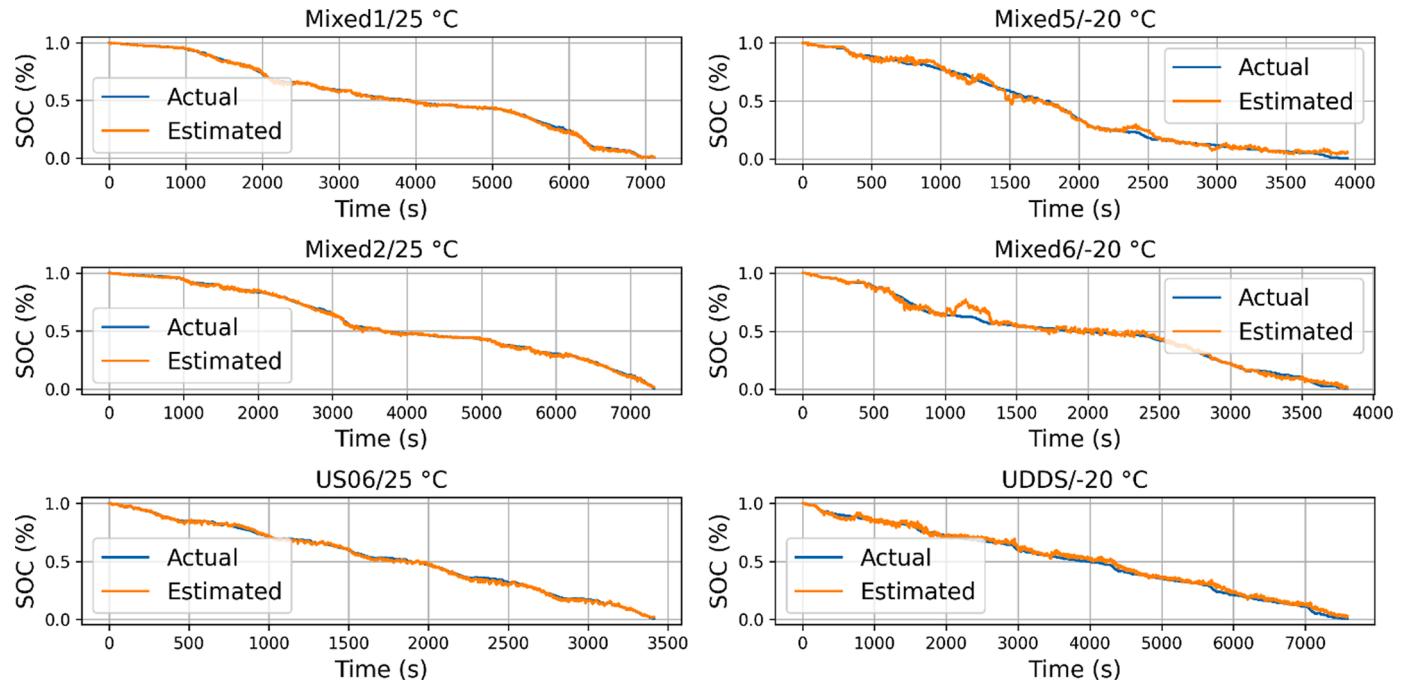


Fig. 10. CatBoost estimation results for single ambient temperature.

tested on batches of 64, 128, 256 and 512 instances.

For training the CatBoost model Scikit-learn framework is utilized. Scikit-learn is another open-source package designed for implementing and designing machine learning algorithms. A 5-fold cross-validation is used to ensure that the model's performance is assessed across different subsets of the data, enhancing the reliability and the robustness of the model. The number of trees is explored across values of 700, 1000, 1500, and 2000, with the learning rate ranging from 0.02, 0.05, 0.1, to 0.2, which allows for a comprehensive evaluation of model performance across different configurations. Grid Search scheme is followed to determine the optimal hyperparameter settings for each model. This

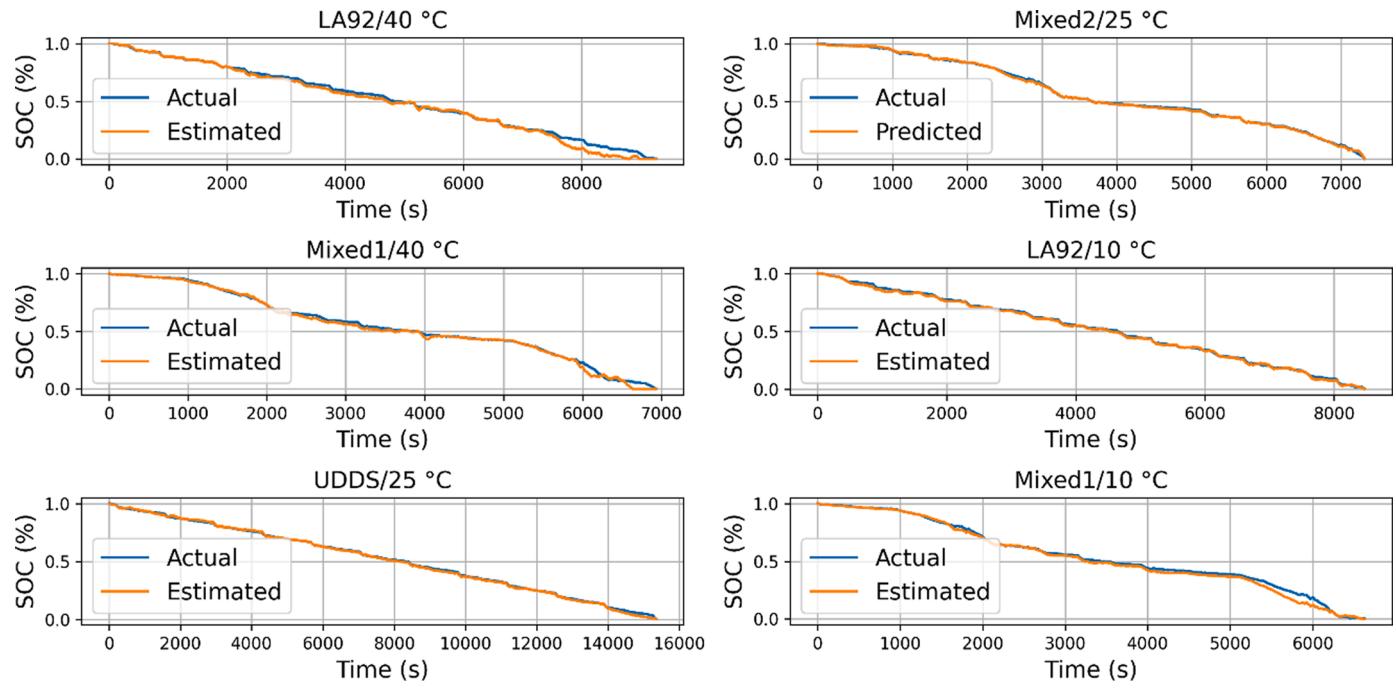


Fig. 11. C-BLSTM-AE estimation results for multiple ambient positive temperatures.

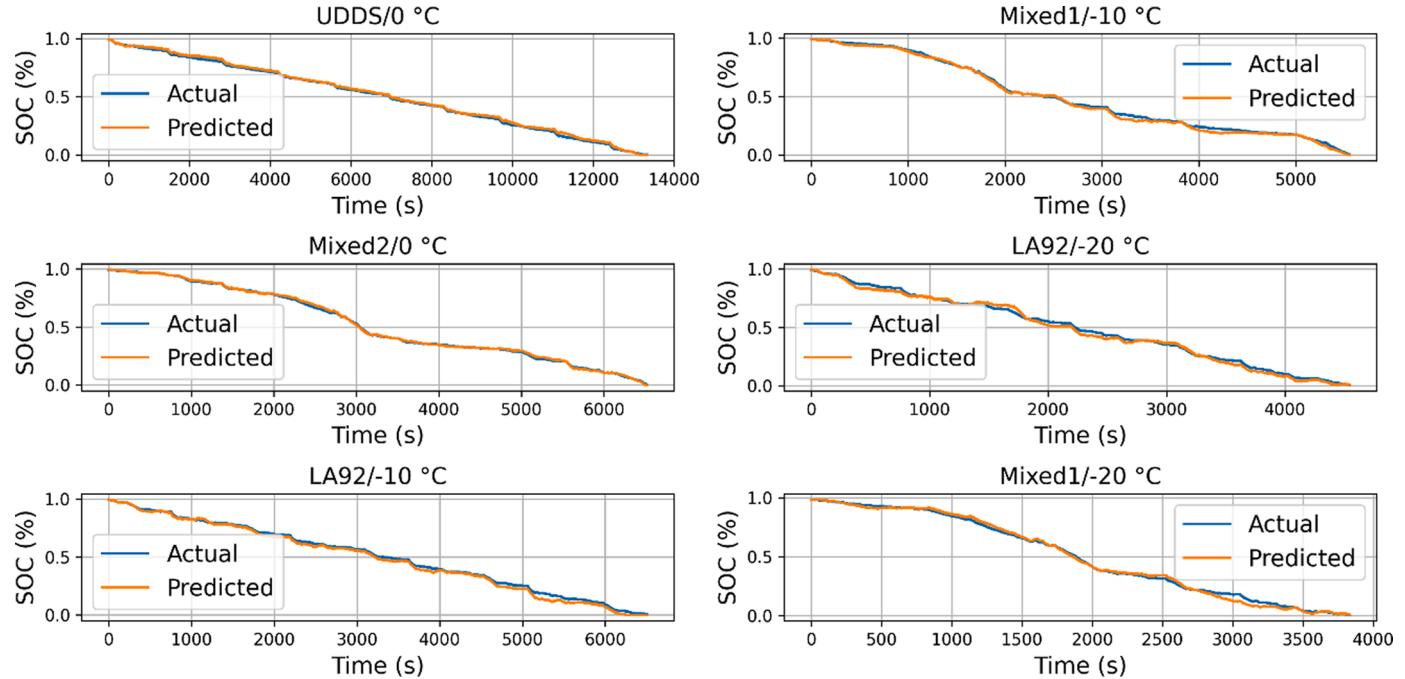


Fig. 12. C-BLSTM-AE estimation results for multiple ambient negative temperatures.

method evaluates exhaustively the hyperparameters space, systematically identifying the configuration that minimizes the loss function. The optimal settings are reported in Table 3 and Table 4.

All models studied were trained in a virtualized environment on a Windows 10, version 22H2 operating system with Intel(R) Xeon(R) Gold 6248R CPU at 3.00 GHz clock frequency and 32 GB of RAM.

5. Model evaluation

5.1. Time window length experimentation

The proposed models use segments of historical measurements as input to generate predictions. This section examines the impact of varying the time window lengths on performance metrics. Both models, C-BLSTM-AE and CatBoost, are evaluated using a range of time windows for input data. The C-BLSTM-AE model is tested with input sequence lengths T_W of 200, 300, 400, 500, 600, and 700 ss, while the CatBoost model examines averaging windows L_{avg} of the same lengths for voltage

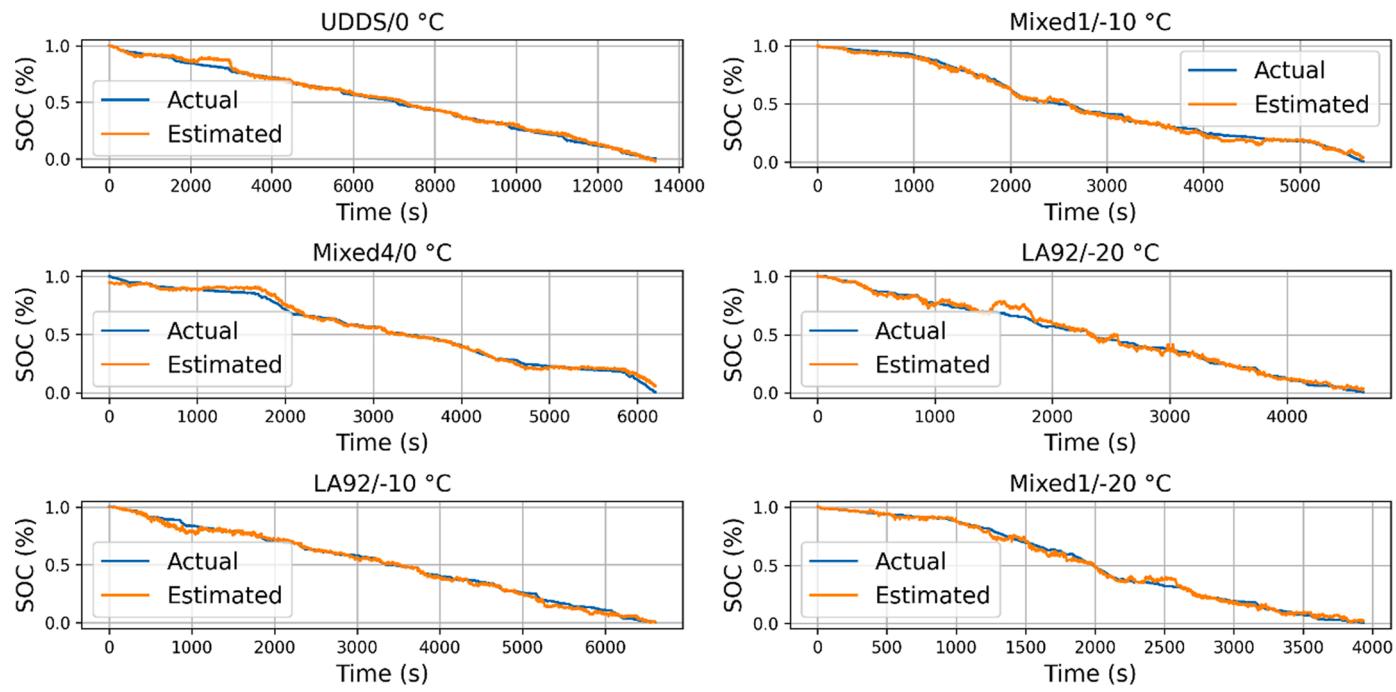


Fig. 13. CatBoost estimation results for multiple ambient negative temperatures.

and current data. Fig. 7 and Fig. 8 illustrate that the time window length influences performance, but the effect varies depending on the ambient temperature. Except at 10 °C, models tend to perform optimally with shorter windows (300–400 ss) at positive temperatures. Conversely models at negative temperatures require longer segments of historical data (longer than 500 s).

This assessment helps identify the most effective window durations for both models in terms of performance. This analysis can help in optimizing the conditions under which the models are implemented to achieve better accuracy and reliability. In Table 5 we summarize lengths of time windows that yielded optimal results for both models.

6. Results

To allow for comparison, the C-BLSTM-AE model and the CatBoost model are validated on the same validation cycles. Along with the aforementioned regression metrics, CPU time corresponding to model training under each temperature condition is presented to ensure a comprehensive evaluation of the models' performance, balancing accuracy with computational efficiency.

In terms of accuracy C-BLSTM-AE model achieved below 1.31 % MAE for all temperature conditions. Except at 10 °C which seems to be challenging for both models, the model recorded the best performance at positive ambient temperatures with a MAE of below 0.78 % with the best MAE results at 40 °C and 25 °C of 0.52 % and 0.55 % respectively. The performance slightly degraded at negative ambient temperatures especially at -20 °C. Aggregated temperatures models recorded a MAE of around 1.03 % for positive temperatures and 1.13 % for negative temperatures (Table 6). The CatBoost model maintained below 1.90 % MAE for all temperature conditions. The model recorded the best performance at positive ambient temperatures with a MAE of below 0.69 % with the best results at 25 °C and jumped to above 1 % for the rest of the conditions. The performance further degraded at negative temperatures and attained 1.33 % at -10 °C and 1.90 % at -20 °C. Aggregated temperatures models recorded a MAE of around 1.09 % for positive temperatures and 1.64 % for negative temperatures. RMSE results show high variability in predictions compared to C-BLSTM-AE results, this is further supported by high values of MAX reaching 15 % at -20 °C as

depicted in Table 6 and Table 7. Another considered aspect is the computation speed. Deep learning model requires extensive back-propagation time to calculate gradient updates, leading to longer CPU computation times. As demonstrated in Table 6, training the C-BLSTM-AE model took several thousand CPU seconds and the CPU time positively correlates with the length of the historical input sequence. In contrast, training the CatBoost model was much faster, requiring significantly less time, measured in mere units of seconds as shown in Table 7.

6.1. Results interpretation

The estimation performance is illustrated in Figs. 9–13, where the estimated SOC profiles are compared with the ground truth SOC. Notably, in Figs. 9 and Fig. 11, the C-BLSTM-AE model shows a strong alignment with the actual SOC curve, with only minor divergences observed, particularly towards the end of the cycle. Fig. 12 displays a fluctuating behavior in the estimations, which supports our observation regarding the challenges faced in estimation under cold conditions, particularly at -10 °C and -20 °C. Fig. 10 shows the performance of the CatBoost model, where the errors are more pronounced compared to those of the DL model. There are significant divergences during mixed load cycles, and the deviations become even more noticeable under cold conditions, as highlighted in Fig. 13.

This paper makes a significant contribution by addressing gaps in the current research landscape, particularly in the aspect of SOC estimation at extremely cold temperatures. While only a handful of studies have considered negative temperatures, this work tackles this challenge by providing estimations at -10 °C and -20 °C under various loads. Additionally, we explored the potential of simple yet powerful machine learning algorithms, such as CatBoost, which offers lower calculation times compared to more complex deep learning algorithms. By focusing on these underexplored areas, our research not only enhances the understanding of battery SOC estimation in warm and cold climates but also introduces efficient and practical solutions for industrial applications.

7. Conclusion

In this paper, two machine learning methods are proposed for SOC estimation task. The proposed methods establish a nonlinear mapping relation between the measurable quantities and the corresponding SOC. The proposed DL model, C-BLSTM-AE, leverages a hybrid architecture to estimate the SOC, demonstrating accurate estimation across different drive cycles and ambient temperatures, with a MAE of 0.52 % for a single ambient temperature and a MAE of 1.03 % for multiple ambient temperatures. Even though traditional machine learning methods did not receive much attention for SOC estimation task, the proposed CatBoost model shows potential for SOC estimation with a MAE of <1.90 % within a short computation time. Since ambient temperature significantly affects the prediction, both models show a trend of decreasing accuracy as the ambient temperature decreases with the higher error observed at -20 °C. CatBoost model is more sensitive to changes in temperature compared to C-BLSTM-AE model. We note that estimations at 10 °C present a challenging case for the models. Overall C-BLSTM-AE model consistently outperforms CatBoost model in terms of MAE, RMSE, and MAX across all temperature conditions. Furthermore, the C-BLSTM-AE estimated SOC is more stable and comply with the physical property of being between 0.0 and 1.0, in contrast to CatBoost negative estimations near the value 0 %. These shortcomings can be overlooked given computation efficiency of the model, which completes in just a few seconds, whereas the regression results of C-BLSTM-AE require several hours of CPU time, with processing time increasing in proportion to the input sequence length.

This work addresses the gap in ML-based SOC estimation by including the commonly neglected scenario of negative temperatures. While only a handful of studies have considered such conditions, this research provides ML-based estimation models at 40 °C, 25 °C, 10 °C, 0 °C, -10 °C, and -20 °C under various load conditions. By tackling this challenge, the paper enhances our understanding of SOC behavior in cold environments, thus broadening the applicability of ML-based SOC estimation.

CRediT authorship contribution statement

Abderrahim Zilali: Conceptualization, Methodology, Software, Validation, Visualization, Writing – review & editing, Writing – original draft. **Mehdi Adda:** Conceptualization, Methodology, Supervision, Validation, Writing – review & editing. **Khaled Ziane:** Resources, Supervision, Validation, Writing – review & editing. **Maxime Berger:** Visualization, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data used for this research paper is open to the public.

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C. et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.
- An, F., Jiang, J., Zhang, W., Zhang, C., & Fan, X. (2021). State of energy estimation for lithium-ion battery pack via prediction in electric vehicle applications. *IEEE Transactions on Vehicular Technology*, 71(1), 184–195.
- Bai, S., Kolter, J.Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv: 1803.01271*.
- Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv: 2003.05991*.
- Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning*. Cambridge, MA, USA: MIT press. Vol. 1.
- Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., et al. (2022). Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, 55(6), 1–36.
- Bhattacharjee, A., Verma, A., Mishra, S., & Saha, T. K. (2021). Estimating state of charge for xEV batteries using 1D convolutional neural networks and transfer learning. *IEEE Transactions on Vehicular Technology*, 70(4), 3123–3135.
- Busch, P., Pares, F., Chandra, M., Kendall, A., & Tal, G. (2024). Future of global electric vehicle supply chain: Exploring the impact of global trade on electric vehicle production and battery requirements. *Transportation Research Record*, Article 03611981241244797.
- Cao, J., & Emadi, A. (2011). Batteries need electronics. *IEEE Industrial Electronics Magazine*, 5(1), 27–35.
- Chemali, E., Kollmeyer, P. J., Preindl, M., Ahmed, R., & Emadi, A. (2017). Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries. *IEEE Transactions on Industrial Electronics*, 65(8), 6730–6739.
- Chemali, E., Kollmeyer, P. J., Preindl, M., & Emadi, A. (2018). State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. *Journal of Power Sources*, 400, 242–255.
- Chollet, F. (2021). *Deep learning with python*. Simon and Schuster.
- Fan, X., Zhang, W., Zhang, C., Chen, A., & An, F. (2022). SOC estimation of Li-ion battery using convolutional neural network with U-net architecture. *Energy*, 256, Article 124612.
- Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771–780), 1612.
- Geetha, A., & Subramani, C. (2019). Development of driving cycle under real world traffic conditions: A case study. *International Journal of Electrical and Computer Engineering*, 9(6), 4798–4803.
- Hannan, M., How, D., Lipu, M.H., Mansor, M., Ker, P., Dong, Z. et al. (2021). Towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model: A deep learning approach.
- Hannan, M. A., How, D. N., Lipu, M. H., Ker, P. J., Dong, Z. Y., Mansur, M., et al. (2020). SOC estimation of Li-ion batteries with learning rate-optimized deep fully convolutional network. *IEEE Transactions on Power Electronics*, 36(7), 7349–7353.
- Hannan, M. A., Lipu, M. H., Hussain, A., & Mohamed, A. (2017). A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renewable and Sustainable Energy Reviews*, 78, 834–854.
- How, D. N., Hannan, M., Lipu, M. H., & Ker, P. J. (2019). State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review. *IEEE access : practical innovations, open solutions*, 7, 136116–136136.
- Hu, C., Cheng, F., Ma, L., & Li, B. (2022). State of charge estimation for lithium-ion batteries based on TCN-LSTM neural networks. *Journal of the Electrochemical Society*, 169(3), Article 030544.
- Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., et al. (2023). A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications*, Article 122666.
- Jain, A., Zamir, A. R., Savarese, S., & Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. New York: Springer.
- Jogunola, O., Adebisi, B., Hoang, K. V., Tsado, Y., Popoola, S. I., Hammoudeh, M., et al. (2022). CBLSTM-AE: A hybrid deep learning framework for predicting energy consumption. *Energyries*, 15(3), 810.
- Keil, P., & Jossen, A. (2017). Impact of dynamic driving loads and regenerative braking on the aging of lithium-ion batteries in electric vehicles. *Journal of the Electrochemical Society*, 164(13), A3081.
- Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12), 6999–7019.
- Liu, Y., Li, J., Zhang, G., Hua, B., & Xiong, N. (2021). State of charge estimation of lithium-ion batteries based on temporal convolutional network and transfer learning. *IEEE access : practical innovations, open solutions*, 9, 34177–34187.
- Ng, K. S., Moo, C.-S., Chen, Y.-P., & Hsieh, Y.-C. (2009). Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied energy*, 86(9), 1506–1511.
- Ng, M.-F., Zhao, J., Yan, Q., Conduit, G. J., & Seh, Z. W. (2020). Predicting the state of charge and health of batteries using data-driven machine learning. *Nature Machine Intelligence*, 2(3), 161–170.
- Pang, Y., Sun, M., Jiang, X., & Li, X. (2017). Convolution in convolution for network in network. *IEEE transactions on neural networks and learning systems*, 29(5), 1587–1597.
- Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*.
- Rae, J.W., Potapenko, A., Jayakumar, S.M., & Lillicrap, T.P. (2019). Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.
- Sarasketa-Zabala, E., Gandiaga, I., Martinez-Laserna, E., Rodriguez-Martinez, L., & Villarreal, I. (2015). Cycle ageing analysis of a LiFePO4/graphite cell with dynamic

- model validations: Towards realistic lifetime predictions. *Journal of Power Sources*, 275, 573–587.
- Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function.
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11), 2673–2681.
- Sepasi, S., Ghorbani, R., & Liaw, B. Y. (2014). A novel on-board state-of-charge estimation method for aged Li-ion batteries based on model adaptive extended Kalman filter. *Journal of Power Sources*, 245, 337–344.
- Shrivastava, P., Soon, T. K., Idris, M. Y. I. B., & Mekhilef, S. (2019). Overview of model-based online state-of-charge estimation using Kalman filter family for lithium-ion batteries. *Renewable and Sustainable Energy Reviews*, 113, Article 109233.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Vidal, C., Kollmeyer, P., Naguib, M., Malysz, P., Gross, O., & Emadi, A. (2020). Robust xev battery state-of-charge estimator design using a feedforward deep neural network. *SAE International Journal of Advances and Current Practices in Mobility*, 2 (2020-01-1181), 2872–2880.
- Wang, B., Liu, Z., Li, S. E., Moura, S. J., & Peng, H. (2016). State-of-charge estimation for lithium-ion batteries based on a nonlinear fractional model. *IEEE Transactions on Control Systems Technology*, 25(1), 3–11.
- Wong, K. L., Bosello, M., Tse, R., Falcomer, C., Rossi, C., & Pau, G. (2021). Li-ion batteries state-of-charge estimation using deep lstm at various battery specifications and discharge cycles. In *Proceedings of the conference on information technology for social good*.
- Xia, B., Wang, H., Tian, Y., Wang, M., Sun, W., & Xu, Z. (2015). State of charge estimation of lithium-ion batteries using an adaptive cubature Kalman filter. *Energies*, 8(6), 5916–5936.
- Yahia, D., Degaa, L., Sara, D., Allah, N. S., Mourad, L. M., & Rizoug, N. (2023). A temporal convolution network to electric vehicle battery State-of-charge estimation. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*.
- Yan, B., & Han, G. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE access : practical innovations, open solutions*, 6, 41238–41248.
- Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7), 1235–1270.
- Zhang, S., Guo, X., Dou, X., & Zhang, X. (2020). A data-driven coulomb counting method for state of charge calibration and estimation of lithium-ion battery. *Sustainable Energy Technologies and Assessments*, 40, Article 100752.
- Zhang, Z. (2018). Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*.
- Zheng, Y., Ouyang, M., Han, X., Lu, L., & Li, J. (2018). Investigating the error sources of the online state of charge estimation methods for lithium-ion batteries in electric vehicles. *Journal of Power Sources*, 377, 161–188.