mathematics

MDPI

Article

DLPformer: A Hybrid Mathematical Model for State of Charge Prediction in Electric Vehicles Using Machine Learning Approaches

Yaoyidi Wang [1], Niansheng Chen [1], Guangyu Fan [1,*], Dingyu Yang [2], Lei Rao [1] and Songlin Cheng [1] and Xiaoyong Song [1]

[1] School of Electronic Information Engineering, Shanghai Dianji University, Shanghai 201306, China; 226003010108@st.sdju.edu.cn (Y.W.); chenns@sdju.edu.cn (N.C.); raol@sdju.edu.cn (L.R.); chengsl@sdju.edu.cn (S.C.); songxy@sdju.edu.cn (X.S.)

[2] Alibaba Group, Hangzhou 310056, China; dingyu.ydy@alibaba-inc.com

* Correspondence: fangy@sdju.edu.cn

Abstract: Accurate mathematical modeling of state of charge (SOC) prediction is essential for battery management systems (BMSs) to improve battery utilization efficiency and ensure a good safety performance. The current SOC prediction framework only considers battery-related features but ignores vehicle information. Additionally, in light of the emergence of time-series Transformers (TSTs) that harness the power of multi-head attention, developing a SOC prediction model remains a significant challenge. Therefore, we introduce a new framework that integrates laboratory battery data with mathematical vehicle model features to improve the accuracy of the SOC and propose a prediction model named DLPformer, which can effectively capture variations in the SOC attributed to both trend and seasonal patterns. First, we apply Matlab/Simulink to simulate a mathematical model of electric vehicles and process the generated vehicle data with Spearman correlation analysis to identify the most relevant features, such as the mechanical losses of the electric motor, differential, and aerodynamic drag. Then, we employ a data fusion method to synchronize the heterogeneous datasets with different frequencies to capture the sudden changes in electric vehicles. Subsequently, the fused features are input into our prediction model, DLPformer, which incorporates a linear model for trend prediction and patch-input attention for seasonal component prediction. Finally, in order to effectively evaluate the extrapolation and adaptability of our model, we utilize different driving cycles and heterogeneous battery datasets for training and testing. The experimental results show that our prediction model significantly improves the accuracy and robustness of SOC prediction under the proposed framework, achieving MAE values of 0.18% and 0.10% across distinct driving cycles and battery types.

Keywords: mathematical model; state of charge; electric vehicle; machine learning

MSC: 68T01

check for updates

Citation: Wang, Y.; Chen, N.; Fan, G.; Yang, D.; Rao, L.; Cheng, S.; Song, X DLPformer: A Hybrid Mathematical Model for State of Charge Prediction in Electric Vehicles Using Machine Learning Approaches. *Mathematics* **2023**, *11*, 4635. https://doi.org/10.3390/math11224635

Academic Editor: Anatoliy Swishchuk

Received: 19 October 2023
Revised: 7 November 2023
Accepted: 10 November 2023
Published: 13 November 2023

1. Introduction

Global emissions and pollution from transportation are increasing, and electric vehicles (EVs) have been proven to be one excellent solution to replace conventional internal combustion engine vehicles [1,2]. EVs use lithium-ion batteries (LIBs) for their high voltage and power density and are equipped with a battery management system (BMS) to regulate the battery usage and scheduling, as well as monitor the battery status and perform maintenance [3,4].

The state of charge (SOC) is defined as the ratio of the remaining capacity to the current maximum available capacity [5]. The accurate prediction of the SOC, representing the ratio of remaining to maximum capacity, plays a pivotal role in the BMS [6]. By ensuring precise

SOC forecasts, the BMS can safeguard against overcharging or undercharging, which can compromise battery longevity and performance [7,8]. Reliable SOC prediction also acts as a cornerstone for balancing control algorithms and optimizing the battery performance, safety, and reliability.

SOC prediction for lithium-ion batteries typically falls into three categories: traditional methods [9], model-based methods [10], and data-driven methods [11,12]. Traditional methods often rely on lookup tables, commonly used in BMSs of major automakers. While these methods are simple and easily deployable, they lack precision and are unsuitable for real-time operations. Model-based methods require the precise construction of lithium-ion battery models. Therefore, designing a sophisticated SOC model is challenging due to the variations in battery types and lifespans observed in real-world usage. Alternatively, data-driven methods are gaining popularity among researchers globally. They involve building nonlinear input–output relationship models directly from extensive historical internal battery data, such as currents, voltages, and temperatures. Various studies have explored the utility of models like long short-term memory (LSTM), the gated recurrent unit (GRU), the recurrent neural network (RNN), and more [6].

However, their prediction accuracy heavily relies on the quality and quantity of sample data [13]. For instance, a low sampling frequency of data makes it difficult for the model to capture abrupt changes in electric vehicles [14]. During vehicle acceleration, the battery may be required to provide a higher current output, while during braking it needs to absorb a larger current input [15]. These momentary changes result in rapid fluctuations in the battery voltage and current, thereby affecting the prediction of the battery SOC. Furthermore, all the above models, while proficient in various aspects, have been hindered by their inability to effectively handle long-term patterns. Since the battery charge and discharge is a lengthy process, the input for the prediction model is usually a long time-series signal. This limitation is primarily attributed to the gradient vanishing problem, which has necessitated the continued reliance on the widely used stochastic gradient descent method for training [16]. In addition, since the internal battery chemistry is a complex process, the SOC prediction is susceptible to the battery discharge rate, ambient temperature, and battery degradation [17]. Meanwhile, there exists a significant nonlinear correlation between the external variables of an electric vehicle and its battery, which can make accurate SOC prediction challenging [18]. For example, the aerodynamic drag coefficient of the electric vehicle varies with its speed, and the mechanical power loss of the motor can result in faster battery discharge, further contributing to inaccurate SOC prediction.

We summarize some limitations of existing SOC prediction methods, including the following:

(I) Current battery input data cannot reflect the sudden changes in the operating conditions of electric vehicles. The battery SOC fluctuates significantly during the acceleration and deceleration of vehicles, making it difficult for models trained on single-time-point or longer-duration data to capture sufficient temporal information.

(II) Most of the existing prediction models were generally considered incapable of handling long-term patterns because of the gradient vanishing problem.

(III) External vehicle conditions are not considered in predicting the SOC. Previous studies on SOC prediction mainly used laboratory battery data as input variables. Furthermore, in actual EV operations it is hard to obtain external conditions that affect battery discharge due to cost and feasibility constraints.

Our research introduces the DLPformer framework, which integrates laboratory battery data with vehicle features to enhance SOC prediction accuracy. We employ a versatile simulation model that adapts to specific operating conditions by incorporating external vehicle data. To capture abrupt changes in electric vehicle behavior, we utilize a high 10 Hz sampling frequency. Additionally, DLPformer, our novel Transformer model, combines a linear trend predictor with patch-input attention for seasonal components, drawing inspiration from previous work [19]. This combination enables DLPformer to effectively capture variations in the SOC attributed to both trend and seasonal patterns. The main contributions of this research are as follows:

- A novel SOC prediction framework: A simulation model of an electric vehicle is built using Matlab/Simulink, which incorporates external vehicle data and laboratory battery data to establish the relationship between external vehicle conditions and battery consumption. To the best of our knowledge, we are the first to formulate this framework that can be customized for specific vehicle operating conditions by inputting corresponding driving cycles and electric vehicle parameters.
- A data fusion method for datasets with different frequencies: We resample the data to capture the sudden change in electric vehicles with features in our simulation and enhance the robustness of the trained model for complex battery conditions.
- We introduce a refined version of the latest time-series Transformer variants, named DLPformer. DLPformer incorporates an empirical rule to select the attention mechanism that is most easily learned in the given domain, which combines a linear model for trend component and patch-input attention for seasonal component prediction. Also, DLPformer acquires information from the entire input and has a superior performance when dealing with long-sequence data.
- We applied our framework to different driving cycles and heterogeneous batteries for training and testing. Through comparison and ablation experiments, we show that our proposed framework and prediction model can achieve more accurate SOC predictions and improve generalization and applicability.

2. Related Work

Recent research has spotlighted deep learning approaches for battery state prediction, focusing especially on the SOC. Standard SOC prediction frameworks typically involve five key steps: battery data collection, data preprocessing, feature engineering, model training, and model prediction [20,21]. In this domain, there are several notable approaches: Caliwag et al. [22] utilized LSTM with a vector autoregressive moving average as input for LSTM layers. Chaoui et al. [23] suggested employing a deep RNN for SOC and battery parameter estimation. Hannan et al. [24] proposed the utilization of a two-hidden-layer stacked GRU model with a one-cycle policy learning rate scheduler for accurate state of charge estimation in Li-ion batteries. Tian et al. [25] proposed a deep learning neural network (DNN) approach for SOC prediction in EVs using data from a 10-minute charging session, which consists of convolutional layers, GRU layers, and a dense layer. Huang et al. [26] developed a hybrid convolutional neural network with a gated recurrent unit (CNN-GRU) model using single-time-point measurements of the voltage, current, and temperature.

However, these studies mentioned above did not consider the influence of vehicle-related data on SOC prediction. Huang et al. [27] used real-world vehicle data from the National Big Data Alliance of New Energy Vehicles (NDANEV) but equalized battery degradation effects with the vehicle driving mileage. Li et al. [28] integrated weather and vehicle data and used a novel dual-dropout-based neural network to predict the SOC. Despite these efforts to include vehicle-related data, both studies suffered from significant data loss and sampling limitations.

With the introduction of the Cloud-based Battery Management System (Cloud-based BMS) concept [29–31], our ability to handle large-scale battery operational data in real time has been elevated. Consequently, the adoption of approaches based on models such as Transformers [32] is starting to make notable strides in the realm of SOC prediction. Hannan et al. [33] employed a Transformer neural network to predict the SOC based on current, voltage, and temperature data and further enhanced the prediction stability with an immersion and invariance adaptive observer. Shen et al. [34] introduced a Transformer-based deep learning approach with self-supervised learning for precise state of charge estimation in lithium-ion batteries. Sitapure et al. [35] addressed the challenge of accurately predicting EV battery parameters using novel time-series Transformers (TSTs), comparing their performance against traditional models and emphasizing the utilization of diverse data sources for improved predictions.

These developments highlighted the promising potential of Transformer-based models for SOC prediction and underscored the importance of considering a broader spectrum of data sources.

3. Prediction Framework for Electric Vehicle

In this section, we describe the details of our prediction framework for the SOC. It includes three modules: laboratory battery data and simulated external vehicle data; data fusion utilizing a high-frequency sampling module; and a deep-learning-based SOC prediction model called DLPformer. The proposed framework is illustrated in Figure 1. We first outline the process of constructing an electric vehicle simulation and analyze the generated features to identify the most related information. Next, we will introduce the fusion of laboratory battery and simulated external vehicle data, which will finally be performed in our proposed prediction model DLPformer.

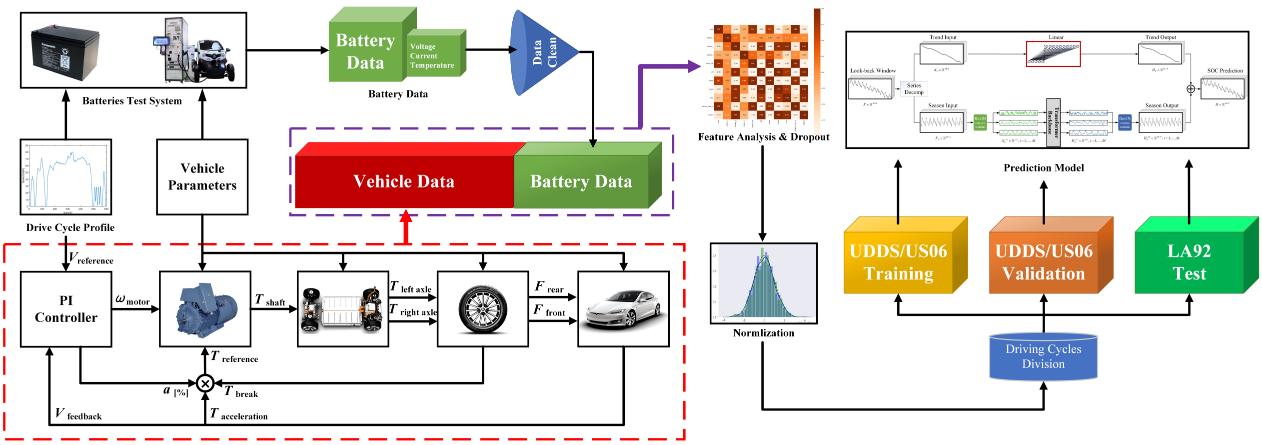


Figure 1. An overview of the proposed SOC prediction framework.

For ease of reading, the abbreviations used in this paper and their meanings are summarized in Table 1.

Table 1. Nomenclature table.

Abbreviation	Description
EV	Electric vehicle
BMS	Battery management system
SOC	State of charge
LIB	Lithium–ion batteries
LSTM	Long short–term memory
GRU	Gated recurrent unit
RNN	Recurrent neural network
DNN	Deep learning neural network
CNN–GRU	Convolutional neural network gated recurrent unit
TST	Time–series Transformer
NDANEV	National Big Data Alliance of New Energy Vehicles
UDDS	Urban Dynamometer Driving Schedule
US06	U.S.06–a set of driving cycles used for emissions testing
LA92	Los Angeles 92
MAE	Mean absolute error
MSE	Mean squared error
RMSE	Root mean squared error
T_{shaft}	Output shaft torque
$T_{\text{reference}}$	Reference torque

Table 1. *Cont.*

Abbreviation	Description
$T_{\text{acceleration}}$	Acceleration torque
T_{brake}	Braking torque
$T_{\text{left axle}}$	Torque on the left axle
$T_{\text{right axle}}$	Torque on the right axle
F_{rear}	Longitudinal force on the rear wheels
F_{front}	Longitudinal force on the front wheels
V_{feedback}	Feedback speed
W_{differ}	Mechanical power transferred through the differential
W_{motor}	Mechanical power of electric motor
W_{wheel}	Mechanical power transferred from axle to wheel
W_{roll}	Power loss due to rolling resistance
W_{break}	Mechanical power required for braking
W_{wind}	Power loss due to wind resistance

3.1. Electric Vehicle Simulation

Since it is difficult to measure or obtain vehicle data (such as losses in the electric motor, power system, and air resistance) in real systems, we conduct a simulation of an electric vehicle model using Matlab/Simulink [36]. Simulink's Powertrain Blockset [37] is a powerful computational tool that has been widely used in simulations and analysis of vehicle dynamics. The Powertrain Blockset consists of several blocks that are designed to calculate various aspects of the motor system.

In order to generate the electric motor's mechanical power data W_{motor}, we utilize the Mapped Motor block to control the output shaft torque T_{shaft}, based on the reference torque $T_{\text{reference}}$. This reference torque is determined by the acceleration torque $T_{\text{acceleration}}$ and braking torque T_{brake}.

Next, the Limited Slip Differential and Longitudinal Wheel blocks are used to calculate the corresponding torques $T_{\text{left axle}}$ and $T_{\text{right axle}}$ and the net longitudinal forces F_{rear} and F_{front} acting on the rear and front wheels, respectively, which allow us to determine the mechanical power transferred through the differential W_{differ}.

By utilizing the Vehicle Body 1DOF Longitudinal block, which considers the feedback speed V_{feedback} and acceleration torque $T_{\text{acceleration}}$, we are able to obtain the mechanical power transferred from the axle to the wheel W_{wheel}, as well as the power loss due to rolling resistance W_{roll}, the mechanical power required for braking W_{break}, and the power loss due to wind resistance W_{wind}.

To summarize, these blocks are used to simulate the powertrain system and vehicle dynamics of an electric vehicle. A schematic of the EV model is designed and presented in Figure 2. The mechanical power of the motor W_{motor}, power transferred through the differential W_{differ}, power transferred from axle to wheel W_{wheel}, power loss due to rolling resistanceW_{roll}, power required for braking W_{break}, and power loss due to wind resistance W_{wind} will be output as a part of the vehicle data. These components are widely recognized and utilized as essential features within real-world automotive datasets.

3.2. Feature Analysis

In the previous section, we obtained various features of the electric vehicle model, including the W_{motor}, W_{differ}, W_{wheel}, W_{roll}, W_{break}, and W_{wind}. However, it is important to note that not all vehicle features are equally relevant to the battery SOC, and the inclusion of irrelevant features in the dataset may result in overfitting. Therefore, a feature analysis is necessary to identify the most relevant characteristics for accurate SOC prediction.

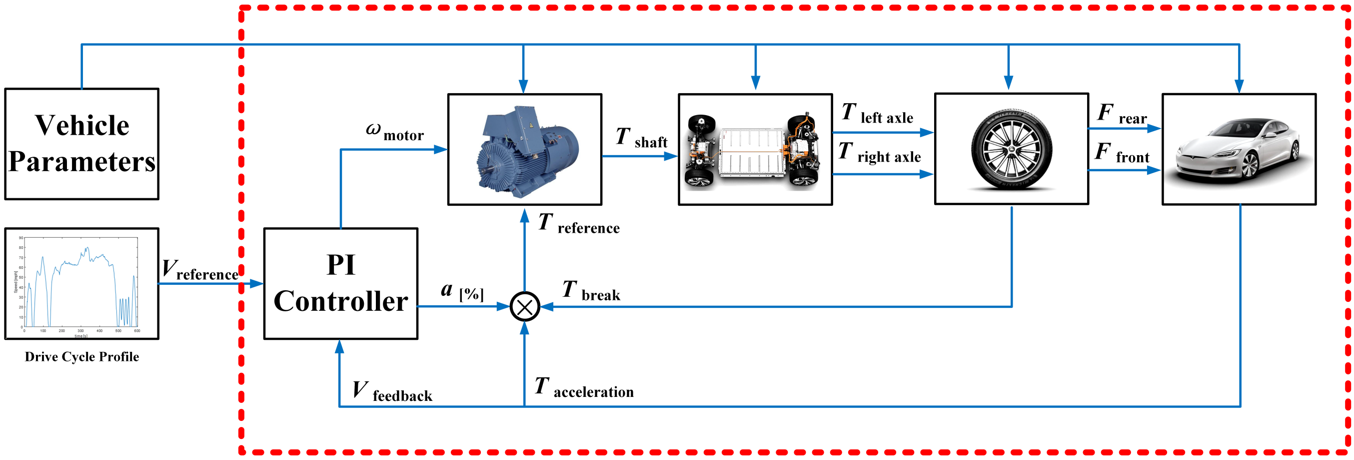


Figure 2. Schematic of the EV model using Matlab/Simulink.

To analyze the features, we utilize the Spearman correlation coefficient [38,39], which is ideal for SOC data that display a monotonic decay. The coefficient measures the strength and direction of the monotonic relationship between two variables, as defined by the following formula:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \quad (1)$$

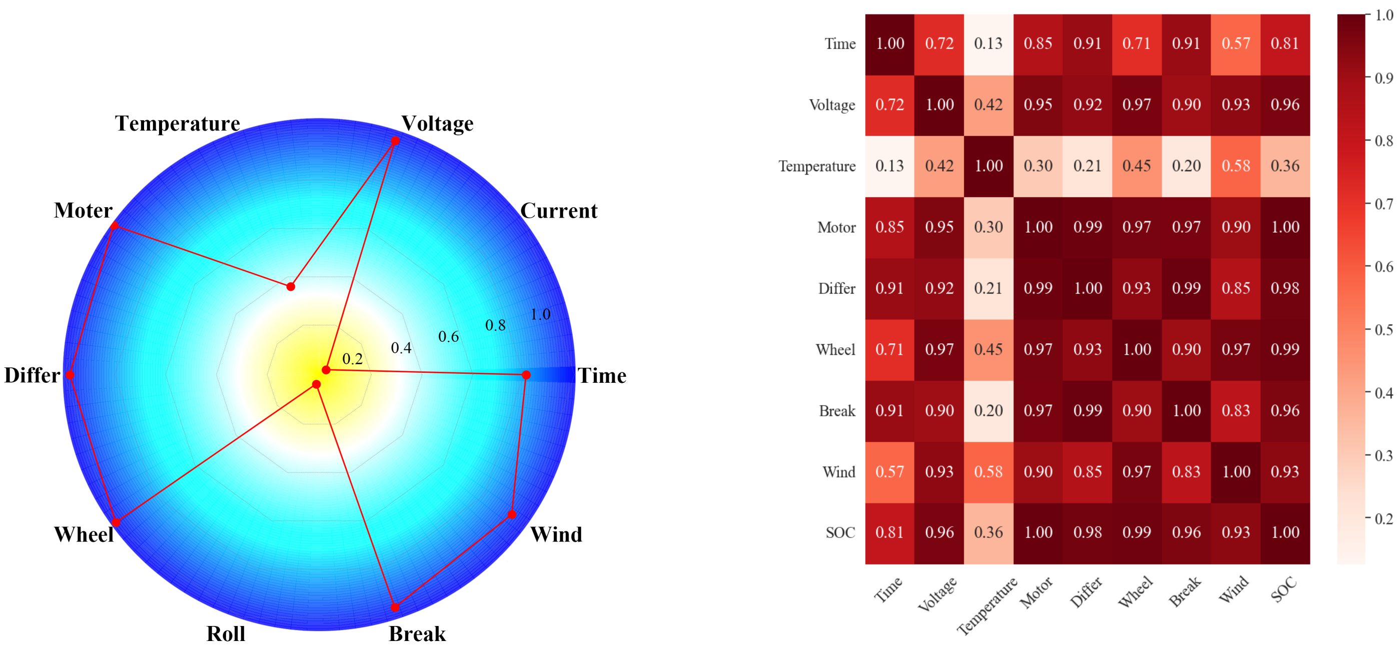
where d is the difference in ranks between the two variables and n is the number of data points. This method does not assume a linear relationship or a specific distribution form, making it more suitable for evaluating the correlation between features and the battery SOC, especially when dealing with monotonic data. The typical range for correlation coefficients is from 0 to 1.0. A value of 1.0 indicates a perfect positive correlation, while a value of 0 suggests no relationship between the variables.

First, we conduct a Spearman correlation analysis between the target variable SOC and various feature values, as illustrated in Figure 3. It is shown that the current in the battery data and the power loss W_{roll} due to rolling resistance in vehicle data both have low correlation coefficients of 0.02 and 0.03. Hence, we drop these two irrelevant features from our dataset. Then, we conduct a secondary correlation analysis on the remaining features, and, as shown in Figure 3, select eight relevant features for SOC prediction. This approach helps to eliminate irrelevant features and reduces the data size, thereby avoiding the overfitting of the model.

3.3. Data Fusion

To integrate the battery data and simulated external vehicle condition data, we employ a data fusion technique to match the spatial and temporal dimensions of the datasets.

For spatial matching, we use the vehicle parameters and driving cycle as the connection link. Specifically, we input the driving cycles, such as the Urban Dynamometer Driving Schedule (UDDS), the U.S.06—a set of driving cycles used for emissions testing (US06), and the Los Angeles 92 cycle (LA92), and the vehicle parameters used in the battery discharge tests into the simulation model to ensure the consistency of the data in the test environment.



(**a**) (**b**)

Figure 3. Spearman correlation coefficient (**a**) between different features and SOC; (**b**) of selected features.

For temporal matching, we first interpolate the missing battery data as it is collected in the laboratory using the sensor. We utilize the Lagrange interpolation method [40]. Assuming that the battery dataset consists of n-time data points, the Lagrange interpolation formula is as follows:

$$l_i(x_{missing}) = \prod_{j=1, j \neq i}^{n} \frac{x_{missing} - x_j}{x_i - x_j} \tag{2}$$

where $l_i(x_{missing})$ is the Lagrange interpolation basis function:

$$y_{missing} = \sum_{i=1}^{n} y_i \cdot l_i(x_{missing}) \tag{3}$$

By interpolating the missing data, we ensure that the battery data are a complete sample with a fixed sampling frequency of 10 Hz. We then set the simulation model's vehicle data sampling frequency to match the same rate of 10 Hz. This allows us to fuse the battery data and simulated vehicle data for SOC prediction.

3.4. Prediction Model

The DLPformer model incorporates both linear and enhanced Transformer modules to capture different aspects of the battery discharge data. Figure 4 illustrates the architecture of the DLPformer model, which begins by decomposing the battery discharge sequence into trend and seasonal components using a moving average approach. To predict the trend component, a linear model is employed, enabling the capture of trend information. The seasonal component undergoes smoothing using the RevIN module, ensuring a more refined input sequence. Subsequently, a Transformer-based model with a patching structure is utilized to capture nonlinear patterns and variable dependencies, facilitating the prediction of the periodic component. In the final step, the predicted trend and seasonal components are aggregated to obtain an overall SOC prediction.

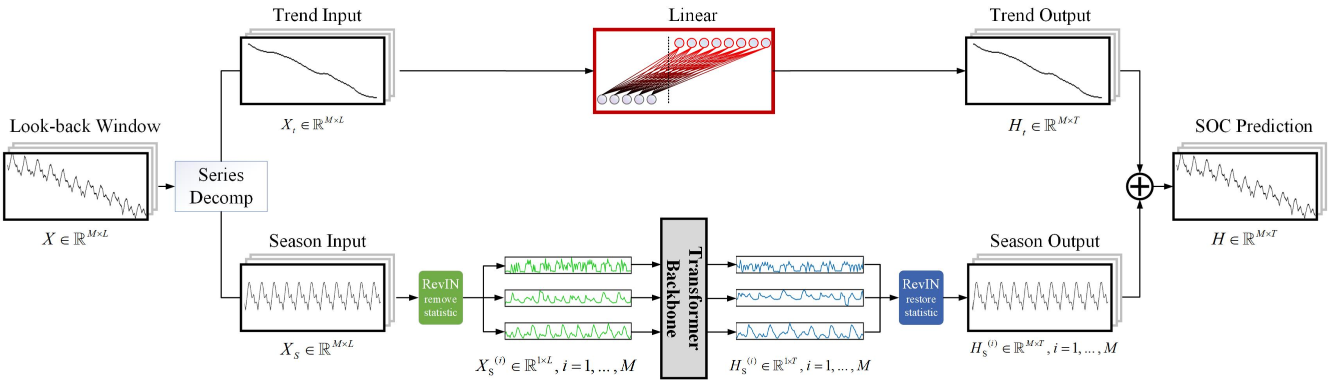


Figure 4. The structure of DLPformer.

3.4.1. Series Decomp

In order to capture complex battery discharge patterns, we adopt a decomposition approach that separates the sequence into trend and seasonal components. The trend component represents the battery's normal state, while the seasonal component represents its fluctuating state. However, directly predicting the discharge sequence for an unknown battery is not feasible. To address this challenge, we employ a sequence decomposition block that gradually extracts a stationary trend from the predicted intermediate hidden variables. Specifically, we smooth the periodic fluctuations by adjusting the moving average line. For an input series $X \in \mathbb{R}^{M \times L}$, the process is as follows:

$$X_t = AvgPool(Padding(X)) \tag{4}$$

$$X_s = X - X_t \tag{5}$$

where X_s and $X_t \in \mathbb{R}^{M \times L}$ represent the seasonal and trends parts, respectively. $AvgPool(X)$ combined with a padding method is exploited as the moving average operation.

3.4.2. Linear Model

The trend component of the DLPformer model is captured using a linear model. After decomposing the battery discharge sequence into trend (X_t) and seasonal (X_s) components, we focus on predicting the trend component using a simple linear model. The linear model is represented as $H_t = W \cdot X_t \in \mathbb{R}^{M \times T}$, where H_t is the predicted trend component for a given input sequence $X_t \in \mathbb{R}^{M \times L}$, W is the weight matrix, and the output length is T. The structure of the linear model is shown in Figure 5.

The linear model is well suited for capturing the trend component due to its ability to model linear relationships. As the trend represents the gradual change in the battery's state during discharge, a linear model is straightforward and intuitive, directly applying a linear transformation to the trend component without introducing complex nonlinear operations. Furthermore, the linear model has a low computational cost and requires fewer parameters compared to more complex models, which makes it computationally efficient and suitable for large-scale SOC prediction tasks.

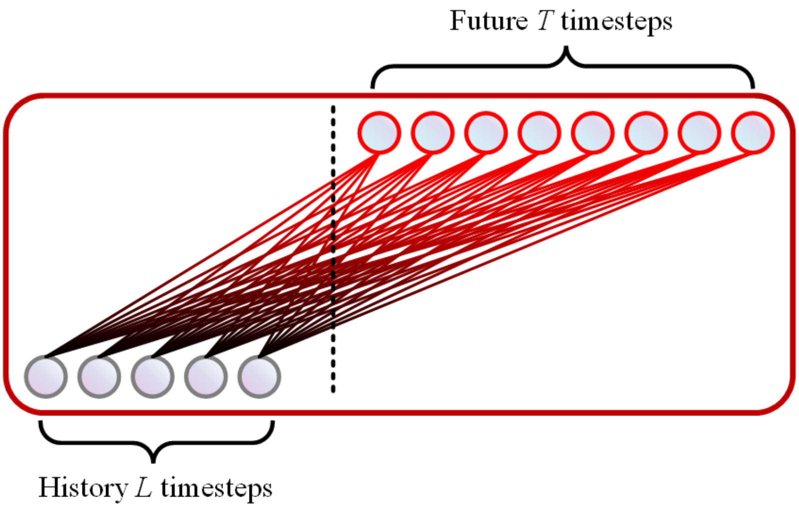


Figure 5. A single linear model.

3.4.3. RevIn Block

The RevIn block is a technique used for predicting the seasonal component in the battery discharge sequence. It leverages a two-step transformation process, involving normalization and denormalization, to improve forecasting accuracy.

In the normalization step, each time-series instance $X_s^{(i)}$ is normalized by subtracting its mean $\mu(i)$ and dividing by its standard deviation $\sigma(i)$, resulting in the following:

$$\hat{X}_s{}^{(i)} = \frac{X_s^{(i)} - \mu(i)}{\sigma(i)} \tag{6}$$

The denormalization step performs the inverse transformation to obtain the predicted output in the original scale. The denormalized prediction $\hat{H}_s{}^{(i)}$ is multiplied by the standard deviation $\sigma(i)$ and added back the mean $\mu(i)$, yielding the following:

$$H_s^{(i)} = \hat{H}_s{}^{(i)} \cdot \sigma(i) + \mu(i) \tag{7}$$

By employing the RevIn block, the model can effectively handle the distribution shift effect and improve the accuracy of seasonal component prediction in battery discharge sequences.

3.4.4. Patching

In the seasonal component prediction, each input univariate time series $\mathbf{x}^{(i)}$ is divided into patches using a patching technique. This patching process helps to reduce the memory usage and computational complexity of the model's attention mechanism, while allowing the model to capture longer historical sequences.

The input time series $\mathbf{x}^{(i)}$ is divided into patches of length P with a nonoverlapping region called the stride S. The patching process generates a sequence of patches $\mathbf{x}_p^{(i)} \in \mathbb{R}^{P \times N}$, where N is the number of patches. The number of patches is determined by the formula $N = \left\lfloor \frac{L-P}{S} \right\rfloor + 2$, where L is the length of the original time series. To ensure the continuity of the sequence, the last value $x_L^{(i)}$ is padded with S repeated numbers and appended to the end of the original sequence before patching.

By using patches, the number of input tokens is reduced from L to approximately L/S. This reduction in tokens leads to a quadratic decrease in the memory usage and computational complexity of the attention mechanism, resulting in improved efficiency.

3.4.5. Transformer Encoder

We utilize a vanilla Transformer encoder to establish the observed signals to latent representations. The patches are projected into the Transformer latent space of dimension D using a trainable linear projection $\mathbf{W}_p \in \mathbb{R}^{D\times P}$, and a learnable additive position encoding $\mathbf{W}_{\text{pos}} \in \mathbb{R}^{D\times N}$ is applied to preserve the temporal order of the patches. The transformed patches are denoted as $\mathbf{x}_d^{(i)} = \mathbf{W}_p\mathbf{x}_p^{(i)} + \mathbf{W}_{\text{pos}}$, where $\mathbf{x}_d^{(i)} \in \mathbb{R}^{D\times N}$ represents the input to the Transformer encoder.

For each head $h = 1, \ldots, H$ in the multi-head attention mechanism, the input is transformed into query matrices $\mathbf{Q}^{(i)} = (\mathbf{x}_d^{(i)})^T\mathbf{W}^Q$, key matrices $\mathbf{K}^{(i)} = (\mathbf{x}_d^{(i)})^T\mathbf{W}^K$, and value matrices $\mathbf{V}^{(i)} = (\mathbf{x}_d^{(i)})^T\mathbf{W}^V$, where $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{D\times d_k}$ and $\mathbf{W}^V \in \mathbb{R}^{D\times D}$. The attention output $\mathbf{O}^{(i)} \in \mathbb{R}^{D\times N}$ is obtained using scaled dot-product attention:

$$\text{Attention}(\mathbf{Q}^{(i)}, \mathbf{K}^{(i)}, \mathbf{V}^{(i)}) = \text{Softmax}\left(\frac{\mathbf{Q}^{(i)}\mathbf{K}^{(i)T}}{\sqrt{d_k}}\right)\mathbf{V}^{(i)} \tag{8}$$

The multi-head attention block also includes BatchNorm layers and a feed-forward network with residual connections, as shown in Figure 6. The resulting representation is denoted as $\mathbf{z}^{(i)} \in \mathbb{R}^{D\times N}$. Finally, a flatten layer with a linear head is used to obtain the prediction result $x(i) = x_{L+1}(i), \ldots, x_{L+T}(i) \in \mathbb{R}^{1\times T}$.

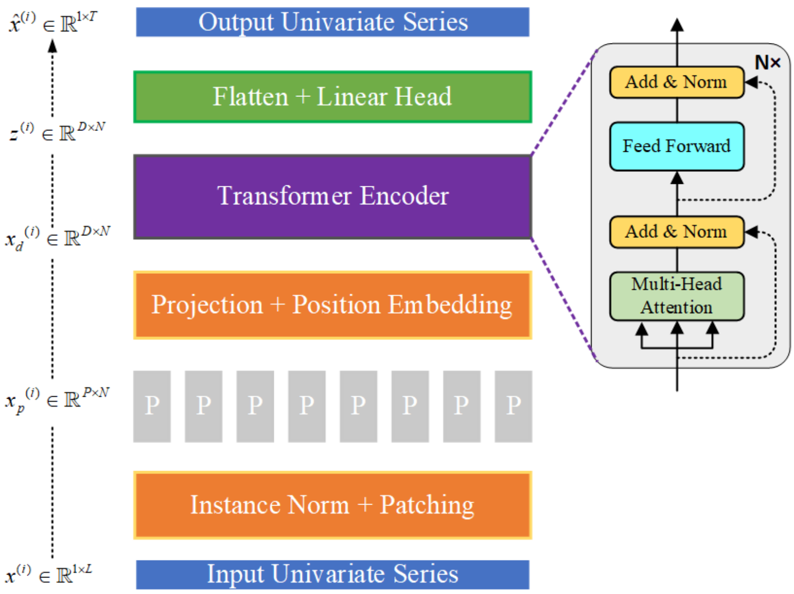


Figure 6. Transformer backbone.

4. Experiment Setup

4.1. Dataset

The dataset employed in our study consists of two heterogeneous batteries: the LG HG2 cell (McMaster University in Hamilton, ON, Canada) [41] and the Panasonic 18650PF cell (University of Wisconsin, Madison, WI, USA) [42]. In the subsequent discussions, we will refer to them as the LG battery and Panasonic battery, respectively. We demonstrated the estimation efficacy of the proposed model based on data sampled at room temperature and selected these batteries for their distinct characteristics in terms of their cathode, anode, capacity, and other specifications. The detailed specifications of these battery cells are provided in Table 2. Figure 7 illustrates the voltage, current, and temperature profiles of both batteries under different driving cycles.

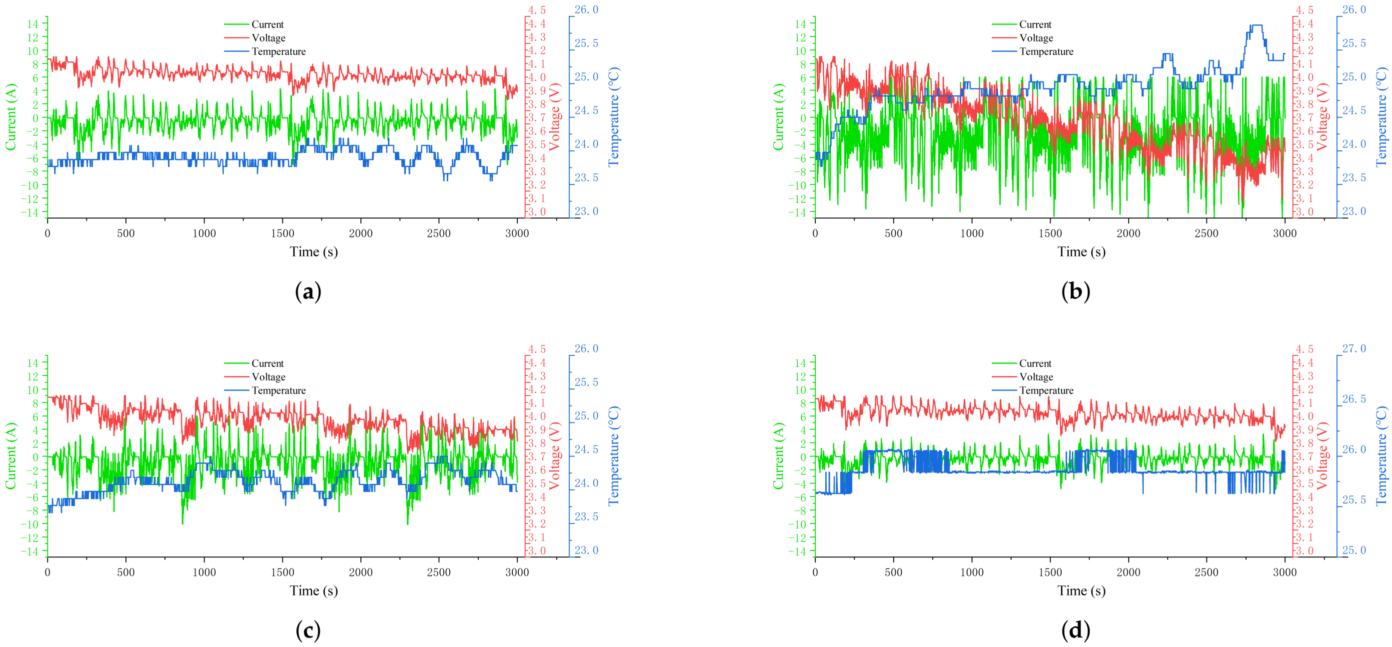


Figure 7. Voltage, current, and temperature profiles of the LG HG2 and Panasonic 18650PF batteries: (**a**) LG battery under UDDS; (**b**) LG battery under US06; (**c**) LG battery under LA92; (**d**) Panasonic battery under UDDS.

Table 2. Specifications of battery cells.

Battery Type	LG HG2	Panasonic 18650PF
Testing institution	McMaster	Wisconsin–Madison
Cathode	LiNiCoMnO2 (NCM)	LiNiCoAlO2 (NCA)
Anode	carbon	carbon
Capacity (mAh)	3000	2900
Thermal chamber volume	8 cu.ft.	8 cu.ft.
Current rating (A)	75	18
Voltage rating (V)	5	5

Then, we utilized these datasets within the framework proposed in our research. Figure 8 showcases partly operational data of a 3Ah LG HG2 battery under the UDDS driving cycle, alongside the vehicle's external data obtained using our framework.

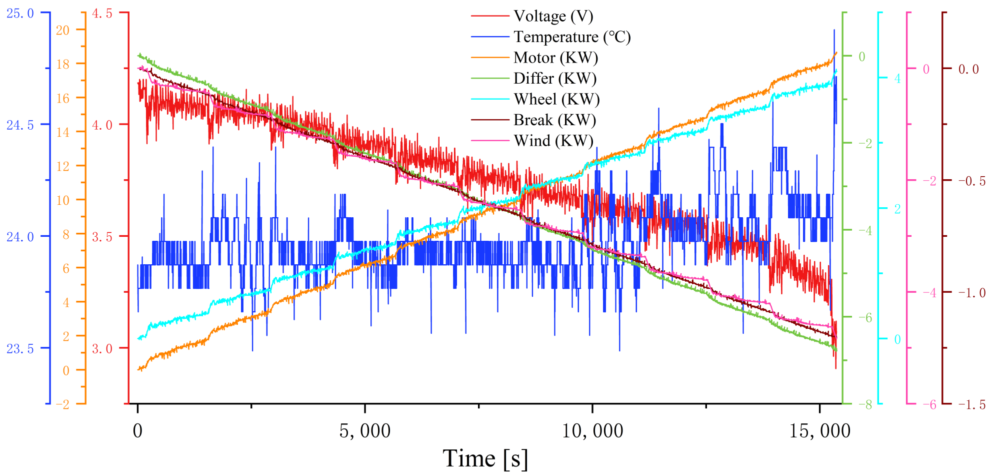


Figure 8. Illustration of battery and vehicle dataset.

4.2. Training and Hyperparameters

Instantiating a deep learning model involves various stochastic processes. To ensure the reproducibility and consistency of the results obtained, all experiments were conducted using a preset seed value. We also introduced a learning rate decay strategy. This strategy aims to expedite the training process and prevent overfitting and speed up the training process. The training process is illustrated in Figure 9.

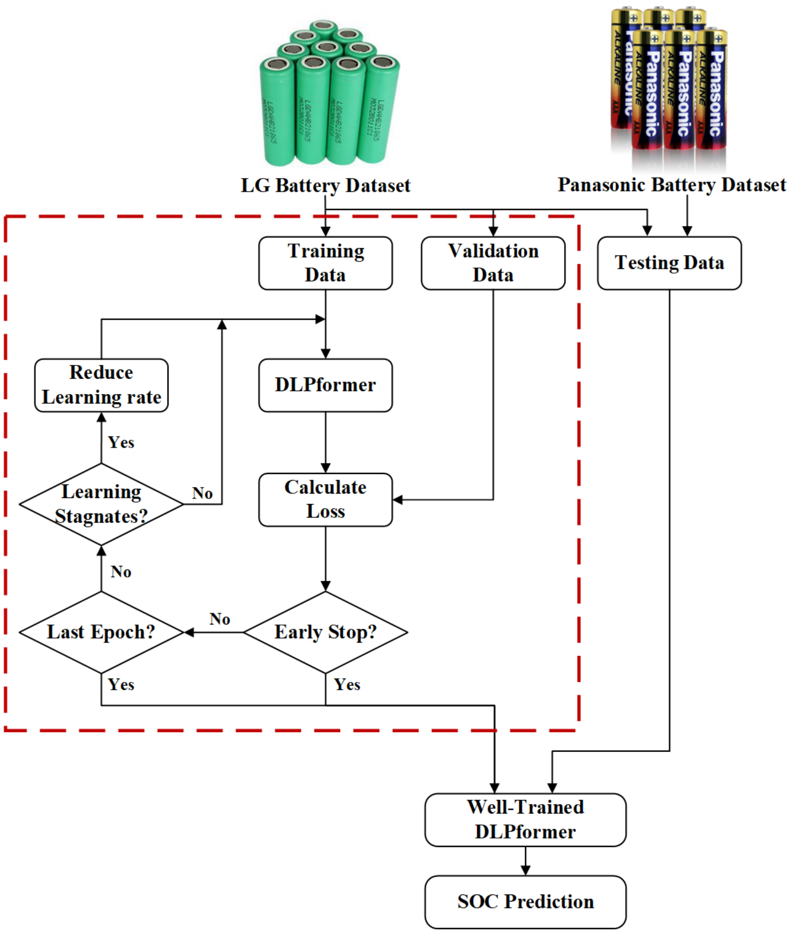


Figure 9. A learning rate decline strategy for training.

During the training process, the model uses Adam to optimize the network parameters, and model parameters were updated based on the loss function computed on the training set, with the batch being 32 and the initial learning rate and training epochs being 0.001 and 30, respectively. Model adjustments were made using the performance metrics evaluated on the validation set. Through validation on the dedicated set, we could examine whether the model was overfitting or underfitting and fine-tune the hyperparameters to optimize the performance. Once the model passed the validation phase and demonstrated a satisfactory performance on the testing battery data, we could be confident in its ability to predict the battery SOC effectively.

4.3. Evaluation Metrics

SOC prediction can be evaluated using various metrics, including the MAE, MSE, and RMSE. These metrics provide quantitative measures of how well a prediction algorithm is performing compared to actual SOC values.

Mean absolute error (MAE): The MAE is given by the following formula:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\mathrm{SOC}_{\mathrm{est}} - \mathrm{SOC}_{\mathrm{act}}| \tag{9}$$

The MAE maintains consistency between the scale of the error metric and the original values. It does not suffer from error amplification due to squaring, making it more robust to outliers compared to the MSE when used as a loss function.

Mean squared error (MSE): The MSE is calculated as follows:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{SOC}_{\mathrm{est}} - \mathrm{SOC}_{\mathrm{act}})^2 \tag{10}$$

The MSE provides a simple and straightforward expression for prediction error. However, squaring the error value amplifies the impact of outliers, making the model highly susceptible to their influence when used as a loss function for training models for SOC prediction.

Root mean squared error (RMSE): The RMSE is computed as the square root of the MSE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathrm{SOC}_{\mathrm{est}} - \mathrm{SOC}_{\mathrm{act}})^2} \tag{11}$$

The RMSE is essentially the square root of the MSE, which helps to ensure that the evaluation metric is consistent with the original value scale.

The performance of the proposed framework or network is evaluated under MAE, MSE, and RMSE criteria.

5. Results and Discussion

In this section, we begin by establishing the superiority of our proposed prediction framework through an examination of the model performance. We substantiate this through comparative experiments and extrapolation analysis. Subsequently, we conduct ablation experiments to demonstrate the effectiveness of individual modules within the prediction framework. Following this, we delve into a study of computational efficiency through experimentation and analysis of the model's training efficiency.

5.1. Model Performance

To comprehensively assess the model's performance and adaptability, we meticulously designed two types of experiments: a comparative experiment and an extrapolation experiment. The details of the experiment setup are shown in Table 3.

Table 3. Model performance experimental settings.

Experiment	Train/Valid		Test	
	Dataset	**Shape**	**Dataset**	**Shape**
Comparison	LG_UDDS, LG_US06	122,195	LG_LA92	54,865
Extrapolation	LG_UDDS, LG_US06	122,195	Panasonic_UDDS	44,264

In the comparative experiment, we utilize the LG battery dataset under UDDS and US06 driving cycles to construct the training and validation sets. These cycles are chosen due to their ability to simulate typical urban driving conditions, encompassing scenarios such as stop-and-go traffic and moderate acceleration patterns. Additionally, the US06 driving cycle accounts for aggressive high-speed acceleration behavior, involving rapid speed fluctuations. To evaluate the model's performance on unseen data, we employ the LG battery dataset under LA92 as the test dataset. By assessing the model's accuracy in predicting the SOC across diverse driving scenarios, we aim to gain insights into the model's inherent limitations. Conversely, in the extrapolation experiment we train the model on the LG dataset, employing the UDDS and US06 driving cycles, and subsequently evaluate its performance on the Panasonic 18650PF cell dataset under the UDDS driving cycle. The primary objective is to gauge the model's adaptability to batteries with distinct characteristics.

5.1.1. Comparison Experiment

First, we conducted a comprehensive benchmark of the proposed model against other recently proposed models, both Transformer-based and non-Transformer models. To ensure a fair evaluation, all models were subjected to the same experimental setups as mentioned above. Table 4 provides an overview of these models, including their architectural details, attention mechanisms, and the results obtained from our experiments. Additionally, we selected the three top-performing prediction models, as depicted in Figure 10, to further illustrate their predictive capabilities and differences in SOC prediction.

Table 4. Performance comparison results of different models.

Model	Struct	Attention	MAE (%)	MSE (%)	RMSE (%)
Transformer [32] (2017)	Encoder–decoder	Full attention	1.1241	0.0475	2.1802
Informer [43] (2021)	Encoder–decoder	Prob sparse	2.1114	0.0698	2.6426
Autoformer [44] (2021)	Encoder–decoder	Auto correlation	0.7876	0.0257	1.6061
FEDformer [45] (2022)	Encoder–decoder	Fourier correlation	0.8080	0.0095	0.9759
DLinear [46] (2023)	Decomp–linear	None	0.2316	0.0012	0.2861
PatchTST [47] (2023)	Encoder	Full attention	0.2701	0.0041	0.6339
DLPformer (proposed)	Decomp–linear–encoder	Full attention	**0.1860**	0.0006	0.2464

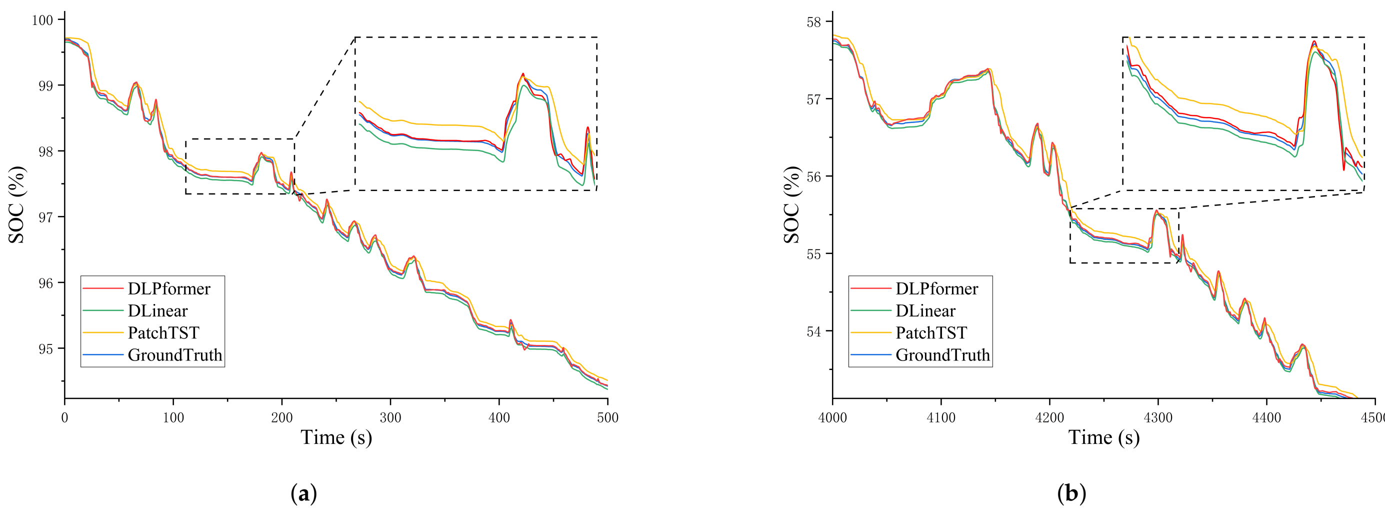


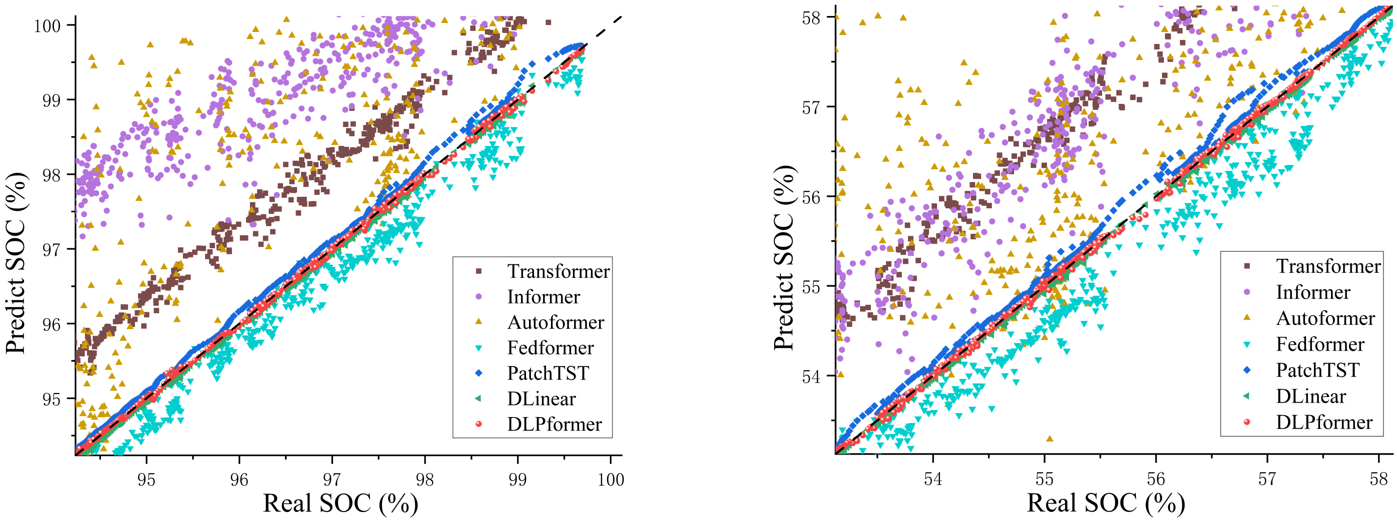
Figure 10. Visualization results of comparison experiments: (**a**) prediction results of top three models in the first 500 s; (**b**) prediction results of top three models in the middle 500 s.

From Table 4, we can conclude that DLPformer achieved impressive results with an MAE of 0.1860%, MSE of 0.0006%, and RMSE of 0.2464%, demonstrating its effectiveness in SOC prediction. Figure 10a,b provide a visual representation of the SOC prediction results for the top three models. These models were trained under the UDDS and US06 driving cycles and subsequently tested on the LA92 driving cycle, specifically during the first 500 s and the middle 500 s. Upon closer examination of the magnified 100 s data, we noticed distinct behaviors among these models. PatchTST is good at aligning with peak data points but substantially deviates from the trend data. We attribute this to the attention mechanism within the Transformer backbone in PatchTST, which can better capture nonlinear relationships and dependencies in the data with spikes. DLinear's predictions adhere more closely to the trend data but struggle to capture rapid fluctuations. In stark contrast, our model, DLPformer, demonstrates an exceptional ability to capture both the trend and periodic components of SOC values. This observation highlights the rationale behind the empirical approach integrated into our model's design, allowing it to effectively select and predict both periodic and trend components, contributing to its superior performance.

5.1.2. Extrapolation Experiment

The extrapolation experiment aims to assess the model's ability to generalize across different battery types, a crucial factor in validating the robustness and versatility of our model in real-world scenarios. To conduct the extrapolation experiment, we first trained the DLPformer on the LG dataset using the UDDS and US06 driving cycles, following the same training procedure as outlined in the comparison experiment. After ensuring the model's proficiency in predicting the SOC for LG batteries, we then put it to the test by evaluating its performance on the Panasonic 18650PF cell dataset under the UDDS driving cycle.

From Table 5, we observe that DLPformer outperforms other models by a significant margin in terms of its MAE, MSE, and RMSE. Specifically, DLPformer achieved an MAE of 0.1010%, MSE of 0.0002%, and RMSE of 0.1413%, showcasing its superior performance in extrapolating SOC predictions to batteries with distinct characteristics. Figure 11a,b illustrate the results of all models on the test dataset. Based on the results of the extrapolation experiment, we find that the Transformer model and its variants, such as Informer, Autoformer, and FEDformer, perform poorly on our heterogeneous battery dataset, making it challenging to achieve accurate predictions for heterogeneous batteries. We consider that the Transformer model and its variants tend to focus on capturing global dependencies in sequences, resulting in weaker modeling capabilities for different battery types and characteristics. In contrast, DLinear, PatchTST, and our proposed DLPformer achieve successful predictions of the SOC for heterogeneous batteries. This suggests that the combination of sequence decomposition and patching can enhance the extrapolation ability and generalizability of the models, which can strengthen the predictions for heterogeneous batteries.



(**a**) (**b**)

Figure 11. Visualization results of extrapolation experiments: (**a**) prediction results of all models in the first 500 s; (**b**) prediction results of all models in the middle 500 s.

Table 5. Results of extrapolation experiments on different models.

Model	MAE (%)	MSE (%)	RMSE (%)
Transformer [32] (2017)	2.5327	0.0838	2.8947
Informer [43] (2021)	3.9213	0.2111	4.5942
Autoformer [44] (2021)	3.2012	0.3533	5.9440
FEDformer [45] (2022)	1.9904	0.1506	3.8807
DLinear [46] (2023)	0.1357	0.0007	0.1848
PatchTST [47] (2023)	0.2590	0.0059	0.7738
DLPformer (proposed)	0.1010	0.0002	0.1413

5.2. Ablation Study

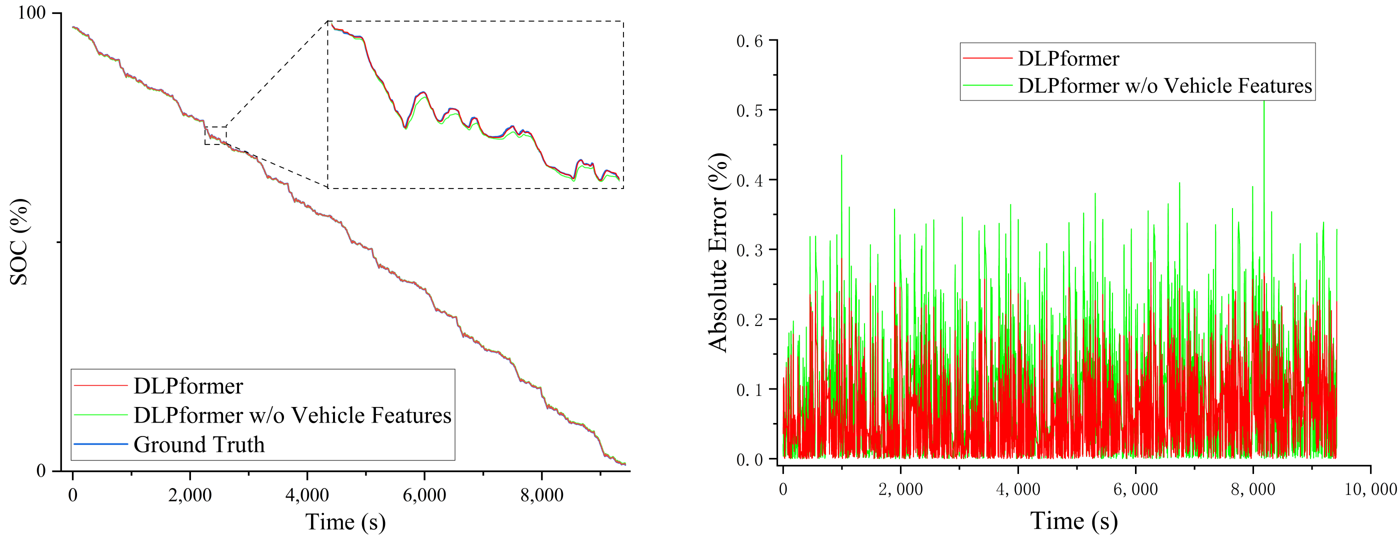
The above experiments have already validated the model's performance. Now, we examine the roles of various modules within DLPformer.

To gain a precise understanding of how the integration of external vehicle data features and the trend and seasonal components of the model contribute, we conduct some ablation experiments. These modules include a linear model, a patch-input attention mechanism, a RevIn module, and input features related to vehicle characteristics. We individually remove each of these components to observe their respective effects on SOC prediction results. The results of these ablation experiments are presented in Table 6.

Table 6. Ablation experimental results of DLPformer.

Method	MAE (%)	MSE (%)	RMSE (%)
DLPformer w/o patch input	0.2476	0.0011	0.2996
DLPformer w/o linear	0.3838	0.0021	0.4889
DLPformer w/o RevIn	0.2258	0.0009	0.2981
DLPformer w/o vehicle features	0.2108	0.0008	0.2979
DLPformer	0.1860	0.0006	0.2464

In the case of "DLPformer w/o Patch-input", where the patch-input Transformer backbone, responsible for handling seasonal patterns, is removed, the model struggles to capture the intricate cyclic variations in the SOC. In contrast, "DLPformer w/o Linear" discards the linear model, which typically handles the trend component prediction. Without this element, the model lacks the capability to identify and predict overarching trends in the SOC, leading to a loss of accuracy in long-term predictions. Additionally, "DLPformer w/o RevIn" reveals that the RevIn module's absence hampers data normalization and denormalization. This impacts the model's ability to interpret and process the data effectively, resulting in suboptimal SOC predictions. Lastly, "DLPformer w/o Vehicle Features" signifies that excluding external vehicle features from the model affects its ability to incorporate valuable contextual information. This omission results in less accurate SOC predictions, emphasizing the significance of integrating vehicle features into the prediction framework. Figure 12a,b visually depict the prediction outcomes and absolute error plots of the DLPformer model and DLPformer without vehicle features. Conversely, "DLPformer", our complete DLPformer model, which integrates all four modules, the linear model for trend prediction, the patch-input attention mechanism for seasonal prediction, the RevIn module for data normalization and denormalization, and vehicle features integration, achieved the best performance, effectively capturing both trend and seasonal patterns.



(**a**) (**b**)

Figure 12. Visualization results of DLPformer with and without vehicle features: (**a**) prediction results of DLPformer; (**b**) absolute error of DLPformer.

In addition, we separately studied the effect of external vehicle feature input into other models to demonstrate the universality of this module. As shown in Table 7, we introduced external vehicle data into various models, both Transformer-based and non-Transformer-based. In the original SOC prediction framework, the focus was exclusively on battery features, with input limited to voltage, current, and temperature data. However, upon implementing our proposed SOC prediction framework, which incorporates vehicle-related features, we observed a significant reduction in errors for most models.

This improvement can be attributed to a holistic assimilation of vehicle-related attributes, which evidently contribute to a more comprehensive and refined understanding of the intricate interactions between the battery and the vehicular dynamics. The enriched data landscape facilitates the model's aptitude to capture and contextualize nuanced operational variations, thereby enabling more accurate and robust SOC predictions across diverse driving scenarios.

Table 7. Results of different models with and without vehicle features.

Method	MAE (%)	MSE (%)	RMSE (%)
Transformer [32]	1.1241	0.0475	2.1802
Transformer with vehicle features	1.0886	0.0439	1.9227
Informer [43]	2.1114	0.0698	2.6426
Informer with vehicle features	1.9711	0.0681	2.6096
Autoformer [44]	0.7876	0.0257	1.6061
Autoformer with vehicle features	0.6290	0.0157	1.2540
FEDformer [45]	0.8080	0.0095	0.9759
FEDformer with vehicle features	1.0864	0.0205	1.1430
DLinear [46]	0.2316	0.0012	0.2861
DLinear with vehicle features	0.2254	0.0010	0.2600
PatchTST [47]	0.2701	0.0041	0.6339
PatchTST with vehicle features	0.2277	0.0032	0.4520

5.3. Computation Efficiency

Computation efficiency [48] is a critical aspect to consider when employing deep learning models, as it directly affects the feasibility and scalability of the model in practical applications. Some of the most used definitions are (not limited to) the training time of the model, power consumption, carbon footprint of the model, execution time of a trained model (run-time), model size, and model parameters. In this section, we use the training time of the model to analyze the computational efficiency of various battery prediction models. Models utilized in this work were trained on an RTX4080 GPU (Nvidia Corporation, Santa Clara, CA, USA) with a memory of 16 GB with the PyTorch 1.13.1 DL library. The processor is an Intel i7-13700F (Intel Corporation, Santa Clara, CA, USA) and RAM (Kingston Technology, Shanghai, China) at 32 GB.

Table 8 and Figure 13 present the experimental results, showcasing the number of training epochs, time per epoch, and MAE (mean absolute error) for each model. The number of epochs denotes the termination point of the training process, determined by our predefined patience threshold of 3. The time per epoch indicates the duration required to complete a single epoch of model training, while the MAE quantifies the prediction error of the models in terms of SOC estimation. Figure 13 reveals that DLPformer notably attains the lowest MAE of 0.18% and requires only 54 s per training epoch.

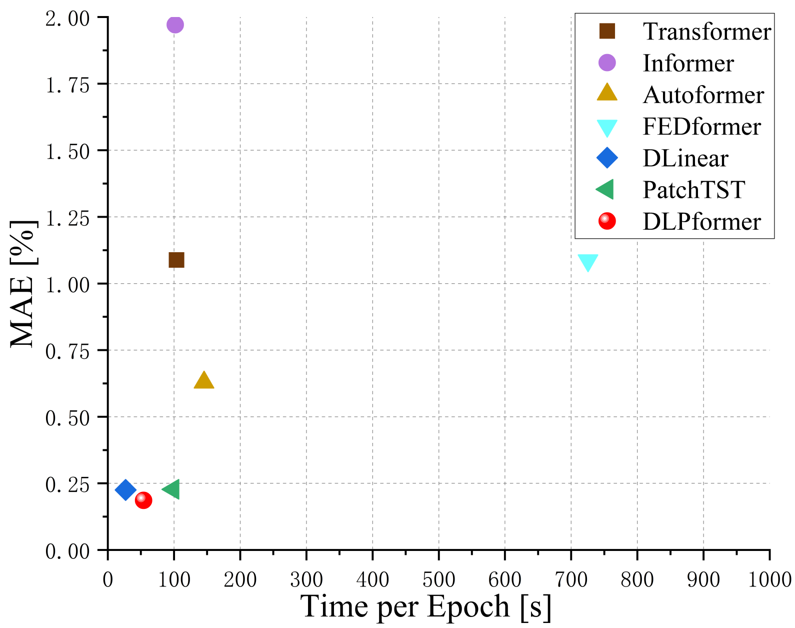


Figure 13. Visualization results of all models in computation efficiency.

Table 8. Computation efficiency of battery prediction models.

Model	Epoch	Time per Epoch (s)	MAE (%)
Transformer [32]	8	103.67	1.0886
Informer [43]	8	101.90	1.9711
Autoformer [44]	9	145.28	0.6290
FEDformer [45]	10	725.54	1.0864
DLinear [46]	20	26.88	0.2254
PatchTST [47]	14	98.76	0.2277
DLPformer	16	54.03	0.1860

Epoch referred to the training method's early stopping criterion mentioned in Figure 9.

6. Conclusions

In this research, we tackle the vital challenge of accurate state of charge prediction, a pivotal aspect of battery management systems for enhancing battery efficiency and safety. The prevailing SOC prediction frameworks have primarily focused on battery-related features, disregarding crucial vehicle information. To address this gap, we introduce a novel framework integrating laboratory battery data and vehicle features, resulting in improved SOC prediction precision. Through the utilization of Matlab/Simulink simulations and

Spearman correlation analysis, we identify pivotal vehicle features, such as the mechanical losses of electric motors, differential, and aerodynamic drag. Moreover, our data fusion method improves the synchronization of heterogeneous datasets, enabling accurate predictions of abrupt electric vehicle changes. Furthermore, the DLPformer prediction model is developed to effectively capture complex SOC variations related to both trend and seasonal patterns. Our DLPformer model is a fusion of a linear trend prediction component and a patch-input attention mechanism for seasonal component prediction. Through comprehensive testing using various driving cycles and heterogeneous battery datasets, our prediction framework showcases remarkable SOC prediction accuracy and robustness. Our prediction framework achieves outstanding performance with MAE values of 0.18% and 0.10% across distinct driving cycles and battery types.

We note that our study has not yet included the evaluation of battery SOC testing under various temperature conditions, and there is further room for improvement in our predictive model. In the future, we plan to focus on these two potential areas for enhancement. First, by including battery data from various temperature conditions we can make the model more adaptable to different situations since batteries function differently under different temperatures. Secondly, for real-world scenarios where the SOC varies dynamically in electric vehicles, having real-time SOC updates could be highly beneficial. Currently, our time-series Transformer provides SOC values every 96 s, which is the sliding window length. However, using shorter sliding window lengths like 48 or 24 s could increase the frequency of SOC updates, leading to more timely and accurate SOC predictions.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W.; data curation, N.C.; formal analysis, L.R.; investigation, G.F.; resources, D.Y.; writing—original draft preparation, Y.W. and D.Y.; visualization, S.C.; supervision, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (NSFC, No. 61702320).

Data Availability Statement: Data in this paper are available from the corresponding authors upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, F.; Wang, D.; Zhao, Y.; Tsui, K.L.; Bae, S.J. A study of the relationship between coulombic efficiency and capacity degradation of commercial lithium-ion batteries. *Energy* **2018**, *145*, 486–495. [CrossRef]
2. Xu, Y.; Liang, J.; Cheng, B.; He, W.; Shu, J.; Li, K.; Xie, Y. Research on the impact of re-electrication in the transportation sector on carbon emission and pollutant emission in Yunnan province. In Proceedings of the 2021 Power System and Green Energy Conference (PSGEC), Shanghai, China, 20–22 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 407–412.
3. Hannan, M.A.; Lipu, M.H.; Hussain, A.; Mohamed, A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renew. Sustain. Energy Rev.* **2017**, *78*, 834–854. [CrossRef]
4. Rahimi-Eichi, H.; Ojha, U.; Baronti, F.; Chow, M.Y. Battery management system: An overview of its application in the smart grid and electric vehicles. *IEEE Ind. Electron. Mag.* **2013**, *7*, 4–16. [CrossRef]
5. Chang, W.Y. The state of charge estimating methods for battery: A review. *Int. Sch. Res. Not.* **2013**, *2013*, 953792. [CrossRef]
6. Lipu, M.H.; Ansari, S.; Miah, M.S.; Meraj, S.T.; Hasan, K.; Shihavuddin, A.; Hannan, M.; Muttaqi, K.M.; Hussain, A. Deep learning enabled state of charge, state of health and remaining useful life estimation for smart battery management system: Methods, implementations, issues and prospects. *J. Energy Storage* **2022**, *55*, 105752. [CrossRef]
7. Zhou, W.; Zheng, Y.; Pan, Z.; Lu, Q. Review on the battery model and SOC estimation method. *Processes* **2021**, *9*, 1685. [CrossRef]
8. Khawaja, Y.; Shankar, N.; Qiqieh, I.; Alzubi, J.; Alzubi, O.; Nallakaruppan, M.; Padmanaban, S. Battery management solutions for li-ion batteries based on artificial intelligence. *Ain Shams Eng. J.* **2023**, 102213. [CrossRef]
9. Martyushev, N.V.; Malozyomov, B.V.; Sorokova, S.N.; Efremenkov, E.A.; Qi, M. Mathematical Modeling of the State of the Battery of Cargo Electric Vehicles. *Mathematics* **2023**, *11*, 536. [CrossRef]
10. Wang, H.; Zheng, Y.; Yu, Y. Lithium-ion battery SOC estimation based on adaptive forgetting factor least squares online identification and unscented kalman filter. *Mathematics* **2021**, *9*, 1733. [CrossRef]
11. Vellingiri, M.T.; Mehedi, I.M.; Palaniswamy, T. A novel deep learning-based state-of-charge estimation for renewable energy management system in hybrid electric vehicles. *Mathematics* **2022**, *10*, 260. [CrossRef]

12. Jafari, S.; Shahbazi, Z.; Byun, Y.C.; Lee, S.J. Lithium-ion battery estimation in online framework using extreme gradient boosting machine learning approach. *Mathematics* **2022**, *10*, 888. [CrossRef]
13. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [CrossRef]
14. Ali, M.U.; Zafar, A.; Nengroo, S.H.; Hussain, S.; Junaid Alvi, M.; Kim, H.J. Towards a smarter battery management system for electric vehicle applications: A critical review of lithium-ion battery state of charge estimation. *Energies* **2019**, *12*, 446. [CrossRef]
15. Martyushev, N.V.; Malozyomov, B.V.; Sorokova, S.N.; Efremenkov, E.A.; Qi, M. Mathematical Modeling the Performance of an Electric Vehicle Considering Various Driving Cycles. *Mathematics* **2023**, *11*, 2586. [CrossRef]
16. Alshareef, S.M.; Fathy, A. Efficient Red Kite Optimization Algorithm for Integrating the Renewable Sources and Electric Vehicle Fast Charging Stations in Radial Distribution Networks. *Mathematics* **2023**, *11*, 3305. [CrossRef]
17. Raoofi, T.; Yildiz, M. Comprehensive review of battery state estimation strategies using machine learning for battery Management Systems of Aircraft Propulsion Batteries. *J. Energy Storage* **2023**, *59*, 106486. [CrossRef]
18. Ragone, M.; Yurkiv, V.; Ramasubramanian, A.; Kashir, B.; Mashayek, F. Data driven estimation of electric vehicle battery state-of-charge informed by automotive simulations and multi-physics modeling. *J. Power Sources* **2021**, *483*, 229108. [CrossRef]
19. Zhang, X.; Jin, X.; Gopalswamy, K.; Gupta, G.; Park, Y.; Shi, X.; Wang, H.; Maddix, D.C.; Wang, Y. First de-trend then attend: Rethinking attention for time-series forecasting. *arXiv* **2022**, arXiv:2212.08151.
20. Ren, L.; Zhao, L.; Hong, S.; Zhao, S.; Wang, H.; Zhang, L. Remaining Useful Life Prediction for Lithium-Ion Battery: A Deep Learning Approach. *IEEE Access* **2018**, *6*, 50587–50598. [CrossRef]
21. Zhang, D.; Zhong, C.; Xu, P.; Tian, Y. Deep Learning in the State of Charge Estimation for Li-Ion Batteries of Electric Vehicles: A Review. *Machines* **2022**, *10*, 912. [CrossRef]
22. Caliwag, A.C.; Lim, W. Hybrid VARMA and LSTM method for lithium-ion battery state-of-charge and output voltage forecasting in electric motorcycle applications. *IEEE Access* **2019**, *7*, 59680–59689. [CrossRef]
23. Chaoui, H.; Ibe-Ekeocha, C.C. State of charge and state of health estimation for lithium batteries using recurrent neural networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8773–8783. [CrossRef]
24. Hannan, M.A.; How, D.N.; Mansor, M.B.; Lipu, M.S.H.; Ker, P.J.; Muttaqi, K.M. State-of-charge estimation of li-ion battery using gated recurrent unit with one-cycle learning rate policy. *IEEE Trans. Ind. Appl.* **2021**, *57*, 2964–2971. [CrossRef]
25. Tian, J.; Xiong, R.; Shen, W.; Lu, J. State-of-charge estimation of LiFePO4 batteries in electric vehicles: A deep-learning enabled approach. *Appl. Energy* **2021**, *291*, 116812. [CrossRef]
26. Huang, Z.; Yang, F.; Xu, F.; Song, X.; Tsui, K.L. Convolutional gated recurrent unit–recurrent neural network for state-of-charge estimation of lithium-ion batteries. *IEEE Access* **2019**, *7*, 93139–93149. [CrossRef]
27. Huang, S.; He, Z.; Li, X. A Method of SOC Estimation for Electric Vehicle Based on Limited Information. In Proceedings of the 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), Nanjing, China, 30 October–2 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4. [CrossRef]
28. Li, R.; Wang, H.; Dai, H.; Hong, J.; Tong, G.; Chen, X. Accurate state of charge prediction for real-world battery systems using a novel dual-dropout-based neural network. *Energy* **2022**, *250*, 123853. [CrossRef]
29. Shi, D.; Zhao, J.; Eze, C.; Wang, Z.; Wang, J.; Lian, Y.; Burke, A.F. Cloud-Based Artificial Intelligence Framework for Battery Management System. *Energies* **2023**, *16*, 4403. [CrossRef]
30. Tran, M.K.; Panchal, S.; Khang, T.D.; Panchal, K.; Fraser, R.; Fowler, M. Concept review of a cloud-based smart battery management system for lithium-ion batteries: Feasibility, logistics, and functionality. *Batteries* **2022**, *8*, 19. [CrossRef]
31. Yang, S.; Zhang, Z.; Cao, R.; Wang, M.; Cheng, H.; Zhang, L.; Jiang, Y.; Li, Y.; Chen, B.; Ling, H.; et al. Implementation for a cloud battery management system based on the CHAIN framework. *Energy AI* **2021**, *5*, 100088. [CrossRef]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
33. Hannan, M.A.; How, D.N.; Lipu, M.H.; Mansor, M.; Ker, P.J.; Dong, Z.; Sahari, K.; Tiong, S.K.; Muttaqi, K.M.; Mahlia, T.I.; et al. Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model. *Sci. Rep.* **2021**, *11*, 19541. [CrossRef]
34. Shen, H.; Zhou, X.; Wang, Z.; Wang, J. State of charge estimation for lithium-ion battery using Transformer with immersion and invariance adaptive observer. *J. Energy Storage* **2022**, *45*, 103768. [CrossRef]
35. Sitapure, N.; Kulkarni, A. Exploring Different Time-series-Transformer (TST) Architectures: A Case Study in Battery Life Prediction for Electric Vehicles (EVs). *arXiv* **2023**, arXiv:2308.03260.
36. Kasera, J.; Chaplot, A.; Maherchandani, J.K. Modeling and simulation of wind-PV hybrid power system using Matlab/Simulink. In Proceedings of the 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 1–2 March 2012; IEEE: Piscataway, NJ, USA , 2012; pp. 1–4.
37. Tran, M.K.; Akinsanya, M.; Panchal, S.; Fraser, R.; Fowler, M. Design of a hybrid electric vehicle powertrain for performance optimization considering various powertrain components and configurations. *Vehicles* **2020**, *3*, 20–32. [CrossRef]
38. Myers, L.; Sirois, M.J. Spearman correlation coefficients, differences between. In *Encyclopedia of Statistical Sciences*; Wiley Online Library: Hoboken, NJ, USA, 2004; Volume 12.
39. Cohen, I.; Huang, Y.; Chen, J.; Benesty, J.; Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.

40. Rashed, M.T. Lagrange interpolation to compute the numerical solutions of differential, integral and integro-differential equations. *Appl. Math. Comput.* **2004**, *151*, 869–878. [CrossRef]
41. Kollmeyer, P.; Vidal, C.; Naguib, M.; Skells, M. *LG 18650HG2 Li-Ion Battery Data and Example Deep Neural Network xEV SOC Estimator Script*; Mendeley Data; McMaster University: Hamilton, ON, Canada, 2020; Version 3. [CrossRef]
42. Kollmeyer, P.J. *Panasonic 18650PF Li-Ion Battery Data*; McMaster University: Hamilton, ON, Canada, 2018.
43. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *arXiv* **2021**, arXiv:2012.07436.
44. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *arXiv* **2022**, arXiv:2106.13008.
45. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 27268–27286.
46. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are transformers effective for time series forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11121–11128.
47. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv* **2022**, arXiv:2211.14730.
48. Strubell, E.; Ganesh, A.; McCallum, A. Energy and policy considerations for modern deep learning research. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13693–13696.