energies

MDPI

Article

Dynamic K-Decay Learning Rate Optimization for Deep Convolutional Neural Network to Estimate the State of Charge for Electric Vehicle Batteries

Neha Bhushan [1,*], Saad Mekhilef [1,2,*], Kok Soon Tey [3], Mohamed Shaaban [4], Mehdi Seyedmahmoudian [2] and Alex Stojcevski [5]

[1] Power Electronics and Renewable Energy Research Laboratory, Department of Electrical Engineering, Universiti Malaya, Kuala Lumpur 50603, Malaysia

[2] School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Hawthorn, VIC 3122, Australia; mseyedmahmoudian@swin.edu.au

[3] Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur 50603, Malaysia; koksoon@um.edu.my

[4] Department of Electrical Engineering, National Chung Cheng University, 168 University Road, Minxiong, Chiayi 621301, Taiwan; m.shaaban@ieee.org

[5] Level 3 Unit 03-08 The Alpha, Curtin University Singapore, 10 Science Park Road, Science Park II, Singapore 117684, Singapore; astojcevski@swin.edu.au

* Correspondence: nehaphdmalaya@um.edu.my (N.B.); smekhilef@swin.edu.au (S.M.)

Abstract: This paper introduces a novel convolutional neural network (CNN) architecture tailored for state of charge (SoC) estimation in battery management systems (BMS), accompanied by an advanced optimization technique to enhance training efficiency. The proposed CNN architecture comprises multiple one-dimensional convolutional (Conv1D) layers followed by batch normalization and one-dimensional max-pooling (MaxPooling1D) layers, culminating in dense layers for regression-based SoC prediction. To improve training effectiveness, we introduce an advanced dynamic k-decay learning rate scheduling method. This technique dynamically adjusts the learning rate during training, responding to changes in validation loss to fine-tune the training process. Experimental validation was conducted on various drive cycles, including the dynamic stress test (DST), Federal Urban Driving Schedule (FUDS), Urban Dynamometer Driving Schedule (UDDS), United States 2006 Supplemental Federal Test Procedure (US06), and Worldwide Harmonized Light Vehicles Test Cycle (WLTC), spanning four temperature conditions (−5 °C, 5 °C, 25 °C, 45 °C). Notably, the test error of DST and US06 drive cycles, the CNN with optimization achieved a mean absolute error (MAE) of 0.0091 and 0.0080, respectively at 25 °C, and a root mean square error (RMSE) of 0.013 and 0.0095, respectively. In contrast, the baseline CNN without optimization yielded higher MAE and RMSE values of 0.011 and 0.014, respectively, on the same drive cycles. Additionally, training time with the optimization technique was significantly reduced, with a recorded time of 324.14 s compared to 648.59 s for the CNN without optimization at room temperature. These results demonstrate the effectiveness of the proposed CNN architecture combined with advanced dynamic learning rate scheduling in accurately predicting SoC across various battery types and drive cycles. The optimization technique not only improves prediction accuracy but also substantially reduces training time, highlighting its potential for enhancing battery management systems in electric vehicle applications.

Keywords: Li-ion battery; electric vehicle; SoC estimation; dynamic k-decay learning rate optimization; convolutional neural network

check for updates

Citation: Bhushan, N.; Mekhilef, S.; Tey, K.S.; Shaaban, M.; Seyedmahmoudian, M.; Stojcevski, A. Dynamic K-Decay Learning Rate Optimization for Deep Convolutional Neural Network to Estimate the State of Charge for Electric Vehicle Batteries. *Energies* **2024**, *17*, 3884. https://doi.org/10.3390/en17163884

Academic Editor: Carlos Miguel Costa

Received: 16 May 2024
Revised: 5 June 2024
Accepted: 20 June 2024
Published: 6 August 2024

1. Introduction

Electric vehicles present a potential solution for environmental concerns. To guarantee the long life and safety of lithium-ion (Li-ion) batteries, accurate SoC prediction is required.

The ratio of existing capacity, Q, at the time to nominal capacity is the formal definition of SoC, as stated in Equation (1) [1].

$$\text{SoC} = \frac{Q(t)}{Q_n} \tag{1}$$

Accurate SoC estimation is essential for maximizing the operational efficiency and lifespan of Li-ion batteries [2]. It enables intelligent battery management systems to optimize charging and discharging processes, thereby mitigating degradation and ensuring safe operation [3]. Moreover, precise SoC prediction is crucial for enhancing driver confidence and facilitating the seamless integration of EVs into everyday transportation infrastructure [4].

Traditional methods of SoC estimation often rely on simplistic models or empirical approaches, which may lack the flexibility and adaptability required to accommodate the diverse operating conditions encountered in real-world EV applications [5]. In contrast, the emergence of advanced machine learning techniques, particularly convolutional neural networks (CNNs), offers a promising avenue for enhancing SoC prediction accuracy [6,7].

This paper presents a novel CNN architecture specifically tailored for SoC estimation in battery management systems. Leveraging the inherent capacity of CNNs to capture complex patterns in sequential data, the proposed architecture comprises multiple Conv1D layers, augmented with batch normalization and MaxPooling1D layers to extract salient features from battery voltage and current measurements. The network culminates in dense layers for regression-based SoC prediction, enabling robust performance across a range of operating conditions.

CNNs can be computationally expensive, especially deep ones, due to their extensive parameterization and the intensive computations involved in convolutional and pooling layers, which can result in longer training times and higher resource requirements [8]. Dynamic k-decay learning rate optimization could indirectly address this by potentially reducing the number of training epochs needed to achieve good performance. Fewer epochs mean less computational time and resources required.

Furthermore, to address the challenges of training efficiency and convergence speed inherent in CNN-based models, an advanced optimization technique is introduced. The proposed dynamic k-decay learning rate scheduling method dynamically adjusts the learning rate during training, responsive to changes in validation loss. This adaptive approach ensures efficient exploration of the parameter space, facilitating rapid convergence and enhancing prediction accuracy. Focusing on optimizing the learning rate could potentially help the model generalize better, even with limited data. A well-optimized learning rate schedule can prevent overfitting (where the model learns the training data too well but fails to generalize to new data), which is crucial when data is scarce.

Experimental validation conducted across various drive cycles and temperature conditions demonstrates the efficacy of the proposed CNN architecture combined with the advanced optimization technique. Comparative analysis against baseline CNN models highlights significant improvements in prediction accuracy, with simultaneous reductions in training time. These findings underscore the potential of the proposed approach to revolutionize battery management systems in EV applications, fostering greater sustainability and reliability in future transportation paradigms.

In summary, this paper contributes a comprehensive framework for SoC estimation in EV battery management systems, leveraging state-of-the-art CNN architectures and advanced optimization techniques. By enhancing prediction accuracy and training efficiency, the proposed methodology lays a foundation for the widespread adoption of electric vehicles, driving toward a greener and more sustainable future.

1.1. Literature Survey

Recently, deep learning models have been getting attention in the field of BMS. The most significant work comprises long short-term memory (LSTM) [9,10], deep neural network (DNN) [11], recurrent neural network (RNN) [12], and CNN [13], which have been shown to achieve good results. The RNN, LSTM, and gated recurrent unit (GRU) [14] can handle the temporal data as well, but the computational cost is more than CNN. In Ref. [15], the hybrid CNN–LSTM approach is used for SoC prediction, where spatial dynamics of features are achieved using CNN, and the LSTM model is used to extract temporal dynamics. Also, transfer learning (TL) has recently been introduced as a novel approach to learning properties of source data, and transferring them to target data provides an opportunity to raid modeling of state estimation of Li-ion batteries from a small amount of data [16–18]. In Ref. [19], transfer learning is applied for SoC estimation under varying temperatures. However, reducing the training time and efficiency of the DL model is still challenging in this area. Recent advancements in SoC estimation span a variety of approaches. Within direct measurement methods, adaptive Coulomb counting has gained traction, integrating traditional Coulomb counting with adaptive filtering to mitigate sensor drift and uncertainties. This approach aims to improve accuracy by continuously adjusting to real-time data [20].

Adaptive systems have also seen progress, exploring advanced Kalman filtering techniques like the ensemble Kalman filter (EnKF) and cubature Kalman filter (CKF) for enhanced accuracy and robustness [21]. In [22], the study presents a novel approach for estimating the state of charge (SoC) in lithium-ion batteries. The method combines two algorithms, the multi-innovation adaptive robust untracked Kalman filter (MIARUKF) and the untracked Kalman filter (UKF), operating on different time scales. This combined approach aims to reduce the computational load on the battery management system (BMS) while improving SoC estimation accuracy and robustness against noise. Some of the literature focused on sliding mode observers, demonstrating their robustness to uncertainties and disturbances, making them attractive for real-world applications [23]. The fusion of physics-based and data-driven models is emerging as a promising direction. Physics-informed neural networks (PINNs) integrate physics-based knowledge into neural network architectures, resulting in improved interpretability and generalization for SoC estimation [24]. In [25], the study presents a comprehensive approach to improving DL-based SoC estimation through advanced hyperparameter optimization and innovative outlier handling. The dynamic k-decay learning rate optimization for deep convolutional neural networks (CNNs) was proposed to address the limitations of existing state of charge (SoC) estimation methods while harnessing the power of deep learning. Traditional methods, such as Coulomb counting and open-circuit voltage, often struggle with accuracy under dynamic conditions and are sensitive to environmental factors and battery aging. Model-based approaches, like Kalman filters, rely on accurate battery models, which are difficult to obtain and may not generalize well. Machine learning methods, while promising, often require extensive feature engineering and can suffer from overfitting. This paper introduces the k-decay learning rate scheduler, a new method for optimization of neural networks that is superior to other learning rate optimization methods. In [26], Cycle learning rates have been proposed for determining the learning rate that removes the requirement to determine the experimentally global learning rates through best values and schedules. Instead of decreasing the learning rate in a monotonous manner, this strategy allows it to vary cyclically between reasonable border values. Training with cyclical learning rates rather than fixed values enhances accuracy in fewer iterations. In [27], The Stochastic decent restart technique has been proposed for enhancing deep neural networks during training time. The history of gradient-like adaptive learning rate algorithms Adam, Adadelta, Adagrad, Adamax, Nadam, and RMSprop decides the change of initialization of parameters [14]. Learning rate optimizers aim to lower the learning rate during neural net training according to a predetermined function. Step decay, exponential decay, cosine decay, k-decay, and polynomial decay are some popular learning rate schedulers that have been used for improving training time and optimization of deep learning networks. The polynomial learning rate with a warm restart is another learning rate optimization method, and it requires only a single warm restart [17].

The above-mentioned learning rates also improve the performance of deep learning models but with more hyperparameter settings and varying training than the k-decay learning rate optimization method.

Additionally, the advanced dynamic k-decay method offers several advantages over cyclic learning rate techniques. Unlike the cyclic learning rate, which typically involves predefined cycles of learning rate adjustments, the dynamic k-decay method continuously adjusts the learning rate based on real-time changes in validation loss. This adaptability allows for more precise optimization, especially in scenarios where the validation loss landscape may change dynamically during training. Consequently, the proposed method offers improved convergence and performance, making it a superior choice for optimizing SoC estimation models and enhancing battery management systems.

1.2. Contributions

The main contributions of the paper:

(1) Advanced optimization technique: The paper proposes an advanced optimization technique, the advanced dynamic k-decay learning rate scheduling method, to enhance training efficiency. This technique dynamically adjusts the learning rate during training based on changes in validation loss, optimizing the training process and improving prediction accuracy. This contribution enhances the effectiveness of SoC estimation models, leading to more reliable battery management systems.

(2) Experimental validation: The effectiveness of the proposed CNN architecture and optimization technique is validated through extensive experimentation. Experimental validation is conducted across various drive cycles and temperature conditions, spanning a range of real-world scenarios. Additionally, dynamic temperature generation and Gaussian noise injection are integrated into the dataset to enhance realism and robustness. The results demonstrate the superior performance of the proposed approach in accurately predicting SoC across different battery types and operating conditions.

Overall, the paper’s contributions lie in its innovative CNN architecture, advanced optimization technique, rigorous experimental validation, and comparative performance analysis, all of which contribute to advancing the field of SoC estimation in battery management systems. By addressing key challenges and proposing effective solutions, the paper provides valuable insights for researchers and practitioners in the field, paving the way for improved battery management systems in various applications.

1.3. Organization of Paper

The proposed method is further divided into 5 sections. Section 2 describes the proposed CNN architecture and dynamic k-decay learning rate optimization. Section 3 describes the experimental process, data preprocessing, and drive cycle dataset for SoC estimation for electric vehicle applications. This section further describes the initial results and explanation of CNN architecture and learning rate optimization. Section 4 defines and compares the results with the error rate in SoC prediction, convergence, and computational speed of different models. Section 5 concludes the paper.

2. Proposed CNN Model with Learning Rate Optimization

2.1. Proposed CNN Architecture

The battery voltage, Vk, current, Ik, and temperature, Tk, are the input matrices for the suggested design. The matrix is then subjected to four more temporal convolutions, with each kernel sliding across the time dimension. This study employs a convolutional neural network (CNN) architecture designed specifically for state of charge (SoC) estimation in battery management systems. The CNN architecture is structured to process input sequences comprising voltage and current measurements over time, aiming to predict the remaining charge in a battery. The architecture is composed of multiple layers, each serving a distinct role. Beginning with two Conv1D layers, the network extracts pertinent features from the input sequences. The first Conv1D layer, containing eight filters, is followed by a second layer with sixteen filters. Both layers utilize a kernel size of 3 and ReLU activation, with padding to maintain input size integrity. To prevent overfitting, L2 regularization with a parameter of 0.01 is applied to the kernel weights of these layers, initialized using the He normal method. Subsequent MaxPooling1D layers with a pool size of two downsample the feature maps, preserving essential information. Batch normalization layers are integrated after each Conv1D layer and the initial dense layer to normalize activations and facilitate training stability. Following the convolutional layers, a Flatten layer converts the feature maps into a one-dimensional vector, subsequently processed by a dense layer with 32 units and ReLU activation. L2 regularization and batch normalization further enhance the robustness of this dense layer. To mitigate overfitting, a Dropout layer with a dropout rate of 0.1 is inserted after the first dense layer, randomly deactivating input units during training. The final layer consists of a single neuron representing the SoC estimation output, without applying any activation function, as the task is formulated as a regression problem. This CNN architecture is tailored to effectively handle sequential data and yield precise SoC estimations in battery management systems.

2.2. Advanced Dynamic K-Decay Learning Rate Optimization

Presently, DL is becoming a popular method for estimating SoC. Recent research has demonstrated that adjusting the learning rate (LR) schedule for deep neural networks can be a very precise and efficient technique for training them. It is commonly known that a training algorithm with a low learning rate converges slowly, while a training strategy with a high learning rate diverges [28]. As a result, different learning rates and schedules must be tried. In general, stochastic gradient descent (SGD) is utilized to update the parameters of deep neural networks.

$$x' = x - \eta \nabla L \tag{2}$$

where x and x' are parameters,

$\boldsymbol{\eta}$ = learning rate parameters, which control the update speed of parameters,

L = loss function.

When η is big, the model converges quickly, although local minimum values may be avoided. Local minimum values can be determined when η are small, but the model converges slowly. This conflict can be resolved using the LR schedule $\boldsymbol{\eta}$(t), which regulates the decay from maximum, LR $\boldsymbol{\eta}_0$, to minimum, LR $\boldsymbol{\eta}$e [29].

We propose a novel method, k-decay, which adds an extra item to the widely used and simple LR schedule (exp, cosine, and polynomial), effectively improving the performance of these schedules and outperforming state-of-the-art LR schedule algorithms.

$$\eta^{\mathrm{k}}(\mathrm{t})' = \eta^{\mathrm{k}}(\mathrm{t}) + \Delta \mathrm{f}^{\mathrm{k}}(\mathrm{t}) \tag{3}$$

where K ∈ N,

$\Delta f^k(t)$ = increment of $\eta^k(t)$, which monitors the rate of change of η(t) in the kth order.

Adaptive learning rates are local learning rates rather than global learning rates.

In this research endeavor, we implement a sophisticated optimization strategy, known as dynamic k-decay learning rate optimization, to refine the training process of our Convolutional Neural Network (CNN) architecture. This methodological approach is encapsulated within a custom callback termed advanced learning rate scheduler, seamlessly integrated into the TensorFlow framework. The overarching objective of this optimization technique is to dynamically modulate the learning rate during the training phase, leveraging insights gleaned from the behavior of the validation loss. By adaptively adjusting the learning rate, we aim to bolster the efficiency, convergence, and overall performance of the CNN model.

The advanced learning rate scheduler callback is instantiated with several pivotal parameters, each playing a distinct role in steering the optimization process. These parameters include the initial learning rate (η_0, the decay factor (α), the patience (P), and the decay patience (D). At the commencement of training, the learning rate is initialized to η_0, providing an initial reference point for subsequent adjustments.

Throughout the training epochs, the callback vigilantly monitors the performance metric, typically the validation loss, and enacts the following rules to dynamically modulate the learning rate:

(1) Decay Rule

If the validation loss fails to exhibit improvement over a predetermined number of consecutive epochs, as defined by the patience parameter (P), the learning rate undergoes a gradual decay. Specifically, the learning rate (η_{t}) at epoch t is updated according to an exponential decay formula:

$$\eta_{\mathrm{t}} = \alpha \cdot \mathrm{n}_{\mathrm{t}-1} \quad (4)$$

where α denotes the decay factor.

This decay mechanism allows the learning rate to decrease gradually over successive epochs, facilitating smoother optimization and preventing abrupt changes that may hinder convergence. By adapting the learning rate based on the model's performance over consecutive epochs, the decay rule encourages stable and efficient training dynamics.

The choice of the decay factor (α) influences the magnitude of the learning rate reduction, with smaller values resulting in slower decay rates and vice versa. Careful selection of the decay factor is crucial to balance between exploration and exploitation during training, ensuring optimal convergence towards the global minimum of the loss function.

In summary, the decay rule provides a mechanism for fine-tuning the learning rate during training, promoting stable convergence, and preventing overfitting. By dynamically adjusting the learning rate based on the model's performance, this regularization technique enhances the robustness and generalization capability of the trained model, contributing to improved performance on unseen data.

(2) Sharp Decay Rule

If the validation loss remains stagnant for an extended duration, surpassing the threshold defined by the decay patience parameter (D), indicative of a prolonged lack of progress, the learning rate undergoes a sharper decay. This entails a more substantial reduction in the learning rate, thereby exerting a pronounced influence on the optimization trajectory.

Let L_{v} represent the validation loss, D_d denote the decay patience parameter, and α represent the initial learning rate; if the validation loss remains stagnant for a duration surpassing the defined threshold D_d, indicative of a lack of progress, the learning rate undergoes a sharper decay. This can be mathematically expressed as:

$$\mathrm{L}_{\mathrm{v}}(\mathrm{t}) \geq \mathrm{L}_{\mathrm{v}}(\mathrm{t} - \mathrm{D}_{\mathrm{d}}) \quad (5)$$

$$\alpha(\mathrm{t}) = \alpha(\mathrm{t} - 1) \times \mathrm{P} \quad (6)$$

where $L_v(t)$ represents the validation loss at time t. P represents the decay factor, determining the magnitude of the learning rate reduction.

These mathematical expressions succinctly capture the operational dynamics of the sharp decay rule within the dynamic k-decay learning rate optimization method. The parameters α, P, and D_d serve as tunable hyperparameters that can be adjusted to fine-tune the optimization process according to specific training task requirements. By integrating this optimization technique into the CNN training pipeline, enhanced convergence and performance of the model are aimed to be achieved, leading to improved predictive accuracy in tasks such as battery management systems and sensor data analysis.

By integrating the advanced learning rate scheduler callback into the CNN training pipeline, we aim to achieve enhanced convergence and performance of the model. This optimization technique plays a crucial role in facilitating efficient training of the CNN architecture, leading to improved predictive accuracy in tasks such as battery management systems and sensor data analysis.

3. Experiment Setup Dataset and Initial Results Explanation

The experimental setup in Figure 1 comprises a battery testing apparatus (Neware's BTS4000), a programmable temperature-controlled chamber (ESPEC's SU-241), a central host computer, and the battery cell being studied. This setup is devised to subject the battery cell to charging and discharging cycles following predefined load profiles facilitated by the BTS4000. Throughout these cycles, data is systematically collected and transmitted to the host computer for analysis. The BTS4000 operates within specified maximum current and voltage limits of 10 A and 5 V, respectively.

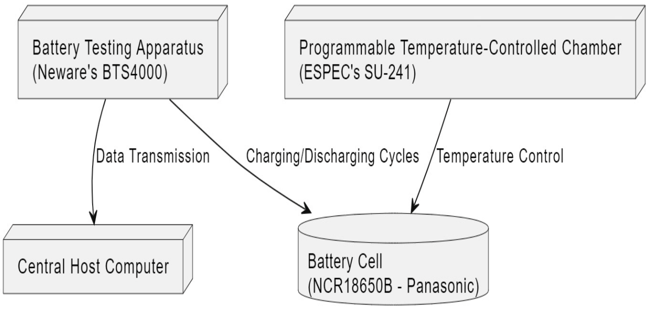

Figure 1. Experimental Setup.

3.1. Data Preprocessing

Initially, battery performance data was collected at constant temperatures of −5 °C, 5 °C, 25 °C, and 45 °C using the programmable temperature-controlled chamber. This step provided baseline data on how the battery behaves under stable, controlled conditions. To enhance the realism and applicability of the experimental results, a dynamic temperature profile was generated to mimic real-world conditions more closely.

(1) Obtaining Dynamic Temperature Data

The experimental validation process incorporates dynamic temperature generation and Gaussian noise injection to enhance the realism of the dataset and improve the robustness of forecasting models. The dynamic artificial battery temperature model meticulously integrates various factors to replicate real-world conditions affecting battery temperature dynamics. Through simulated driving patterns, environmental temperature fluctuations, battery usage impacts, and cooling system efficiency, the model comprehensively captures the complexities of thermal behavior. Dynamic temperature data was generated by combining several factors that influence battery temperature. The daily temperature variation was modeled using a sine wave function with a 24 h period and an amplitude reflecting typical daily temperature changes, combined with random Gaussian noise to introduce natural variability. Mathematically, this can be represented as:

$$Temp\ Variation = 4\sin\left(\frac{2\pi Time(hours)}{24}\right) + \mathcal{N}(0,1) \tag{7}$$

where N (0, 1) represents Gaussian noise, with a mean of 0 and a standard deviation of 1. Simulated driving patterns were generated to account for the impact of real-world driving conditions on battery temperature. This was done using a random walk process, which was then smoothed to resemble realistic acceleration and deceleration patterns. The impact of battery usage on temperature was modeled by scaling the absolute values of these patterns while cooling efficiency was represented by scaling the patterns negatively. The overall dynamic temperature profile was calculated by combining these components.

(2) Adding Gaussian Noise for Robustness

In addition to the dynamic temperature profile, Gaussian noise was introduced to the training data to improve the robustness and generalization capabilities of the model. Gaussian noise, characterized by a normal distribution with a small standard deviation (e.g., 0.01), was added to the feature values. This ensures that the model does not overfit the specific conditions of the training data and can handle slight variations in input data effectively. The noise can be mathematically represented as:

$$\text{Noisy Data} = \text{Original Data} + \mathcal{N}(0, 0.01) \tag{8}$$

where N(0, 0.01) is Gaussian noise, with a mean of 0 and a standard deviation of 0.01.

Furthermore, to enhance the robustness of forecasting models, Gaussian noise is incorporated into the dataset, mimicking the uncertainties encountered during electric vehicle operations. This noise injection aids in training models to handle real-world variations, resulting in more resilient and generalizable forecasting performance, crucial for ensuring the reliability of electric vehicle systems. The chosen standard deviation for the Gaussian noise injection is 0.1, representing a moderate level of variability introduced into features such as voltage and current. This level of noise emulation effectively mirrors the fluctuations and uncertainties commonly encountered during electric vehicle operations, ensuring that the forecasting models trained on augmented data are equipped to handle real-world variations with resilience and accuracy. By incorporating this level of noise, the models develop a robust understanding of the inherent variability in sensor data, leading to more reliable predictions in diverse environmental and operational conditions.

To ensure consistency and facilitate model training, the target variable (SoC) is normalized to a range between 0 and 1. This normalization process aids in maintaining uniformity in the scale of the target variable across different datasets. Additionally, numerical features undergo scaling using MinMaxScaler exclusively fitted on the training data to prevent data leakage and ensure consistent scaling across features. Feature scaling is imperative in machine learning to prevent biases caused by features with larger magnitudes dominating the model and to promote faster convergence during training.

$$\mathrm{X}_{\text{scaled}} = \frac{(\mathrm{X} - \mathrm{X}_{\text{min}})}{(\mathrm{X}_{\text{max}} - \mathrm{X}_{\text{min}})} \tag{9}$$

For time series forecasting, the dataset is organized into sequences using a sliding window approach, with each sequence comprising 90 timesteps. This methodology effectively captures temporal dependencies in the data, allowing models to learn from past patterns. Subsequently, separate sequences are generated for training and testing datasets, preserving temporal dynamics. The training data undergoes a 70-30 split for training and validation, after being resampled to 1 Hz and normalized to the 0–1 range to ensure uniformity and stability across the dataset.

3.2. Dataset Description

In the experimental setup for SoC estimation, the NCR18650B battery in Table 1 from Panasonic plays a pivotal role as the primary power source. Its advanced NCA chemistry ensures stable voltage output, vital for accurate data collection. With a nominal capacity of 3400 mAh and a max continuous discharge of 4.87 A, this battery provides sufficient energy for powering all the components of the experimental apparatus. Additionally, its robust cycle life of approximately 500 cycles ensures long-term reliability, minimizing interruptions during experimentation. The integration of the NCR18650B battery enhances the precision and reliability of the SoC estimation experiments, contributing to advancement in battery management systems.

The initial dataset, focusing on the first derivative, employs the DST and US06 drive cycles for testing, while the remaining drive cycles are allocated for training purposes. This segmentation strategy helps refine predictive accuracy.

Battery performance under specific condition takes a more comprehensive approach, incorporating all available drive cycles into both the testing and training phases. This ensemble includes DST, FUDS, UDDS, US06, and WLTP spanning a temperature range of

−5 to 45 degrees Celsius. This approach enhances the robustness of battery performance evaluation, ensuring a comprehensive analysis that considers diverse operational contexts.

Table 1. Specification of battery chemistry.

Specification	**NCR18650B (Panasonic)**
Chemistry	NCA
Nominal Voltage	3.6 V
Cut-off Voltage (Discharge-Charge)	2.5–4.2 V
Nominal Capacity	3400 mAh
Max Continuous Discharge	4.87 A
Cycle Life	500

By employing this dual-methodological approach, the study effectively examines and distinguishes the performance characteristics of each battery chemistry across a range of drive cycles. This thorough evaluation not only illuminates the operational strengths and limitations of each chemistry but also lays the groundwork for informed advancements in battery technology tailored to meet the requirements of future electric vehicle (EV) drive cycles. Specifically, the DST and US06 drive cycles are utilized for testing, while the remaining cycles are used for training. The training data is divided into a 70-30 split for training and validation, respectively. Additionally, the data is resampled to 1 Hz and normalized to the 0–1 range to ensure uniformity and stability.

3.3. Hyperparameter Tuning and Training Process

The neural network model was configured with specific hyperparameters to govern its behavior during training. The input shape of the model was defined as (90, 3), reflecting the dimensions of the input data comprising time–series measurements such as voltage and current readings. The model architecture consisted of convolutional and dense layers, each incorporating activation functions, regularization techniques, and optimization algorithms. Notably, the model utilized the mean squared error (MSE) as the loss function, which is a common choice for regression tasks. Additionally, mean absolute error (MAE) and root mean squared error (RMSE) were employed as evaluation metrics to gauge the model's performance during training.

The optimizer chosen for training was the adagrad optimizer, with a manually specified learning rate of 0.01. adagrad dynamically adapts the learning rate for each parameter based on the historical gradients, making it suitable for non-stationary objectives and sparse data. Furthermore, the training process was augmented with custom callbacks, including early stopping and model check point, to enhance training efficiency and prevent overfitting. The early stopping callback halted training if the validation loss failed to decrease significantly for a prolonged period, while the model check point callback saved the best model based on the monitored validation loss metric.

The training process commenced with the compilation of the neural network model, wherein the MSE loss function, adagrad optimizer, and evaluation metrics were specified. The compiled model was then trained using the fit method, with a maximum of 1000 epochs and a batch size of 128. During training, the dataset was split into training and validation sets, with 30% of the data reserved for validation purposes. The training process was monitored using custom callbacks, which dynamically adjusted the learning rate and saved the best model based on the validation loss metric. The training time was recorded using the time module, providing insights into the computational resources consumed during the training phase.

The hyperparameter tuning and training process outlined above aimed to cultivate a robust and high-performing neural network model for accurate state-of-charge estimation, leveraging a combination of meticulously chosen hyperparameters, advanced optimization algorithms, and efficient training methodologies.

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N}}}\sum\nolimits_{\mathrm{k}=1}^{\mathrm{N}}(\mathrm{SOC}_{\mathrm{k}} - \mathrm{SOC}_{\mathrm{k}}^{*})^2 \tag{10}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum\nolimits_{\mathrm{k}=1}^{\mathrm{N}}(\mathrm{SOC}_{\mathrm{k}} - \mathrm{SOC}_{\mathrm{k}}^{*})^2 \tag{11}$$

where N = total number of training samples,

SOC_k = the model's estimated SoC at timestep k, and

SOC_k^* = the ground truth SoC value at timestep k in the loss term.

3.4. Computational Framework

The research utilized Python 3.10.12, a specific version of the Python programming language. TensorFlow 2.8.0 was the framework of choice for constructing and training the neural network models. To speed up the training and inference (the process of making predictions) of the neural network models, NVIDIA T4 GPUs were employed. Graphics processing units (GPUs) are processors that can perform many operations in parallel, making them particularly suited to the matrix and vector computations commonly found in machine learning. Google Colab is mentioned as the platform providing access to T4 GPUs.

4. Final Result and Analysis

4.1. SoC Prediction at Ambient and Variable Temperatures

Figures 2–5 depict the diagrams for state-of-charge (SoC) estimation across four different temperature conditions (−5 °C, 5 °C, 25 °C, and 45 °C) for test results obtained from the dynamic stress test (DST) and the Urban Dynamometer Driving Schedule 2006 (US06) drive cycles. These diagrams provide visual representations of the SoC estimation performance under varying temperature environments, showcasing the efficacy of the proposed methodology across a range of operating conditions. By examining the SoC estimation results across different temperatures for both the DST and US06 drive cycles, valuable insights can be gained into the robustness and adaptability of the proposed approach in real-world scenarios.

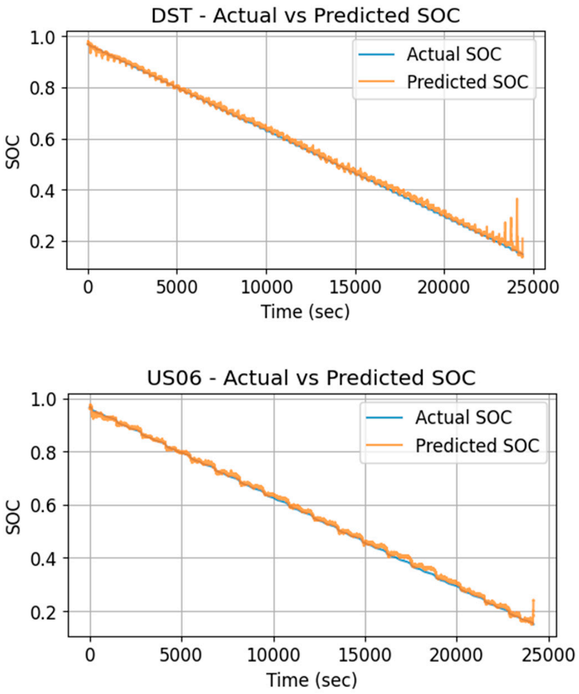

Figure 2. Test Results of US06 and DST at 25 °C.

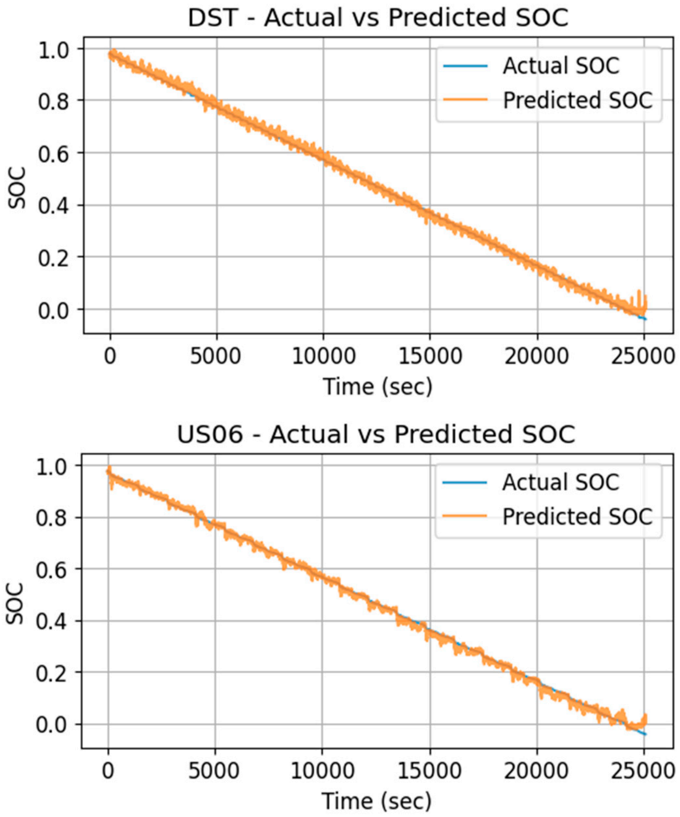

Figure 3. Test Results of US06 and DST at 45 °C.

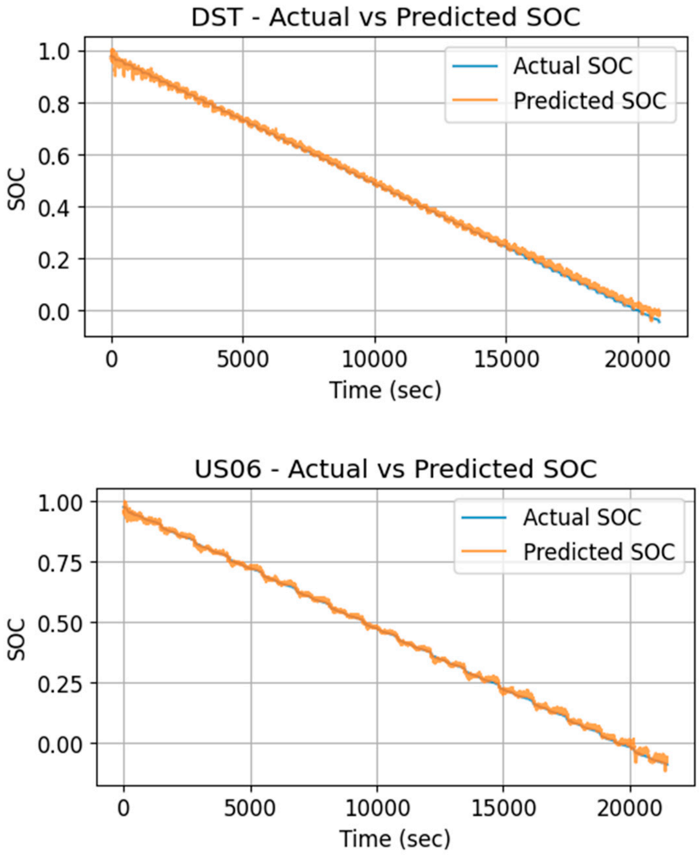

Figure 4. Test Results of US06 and DST at 5 °C.

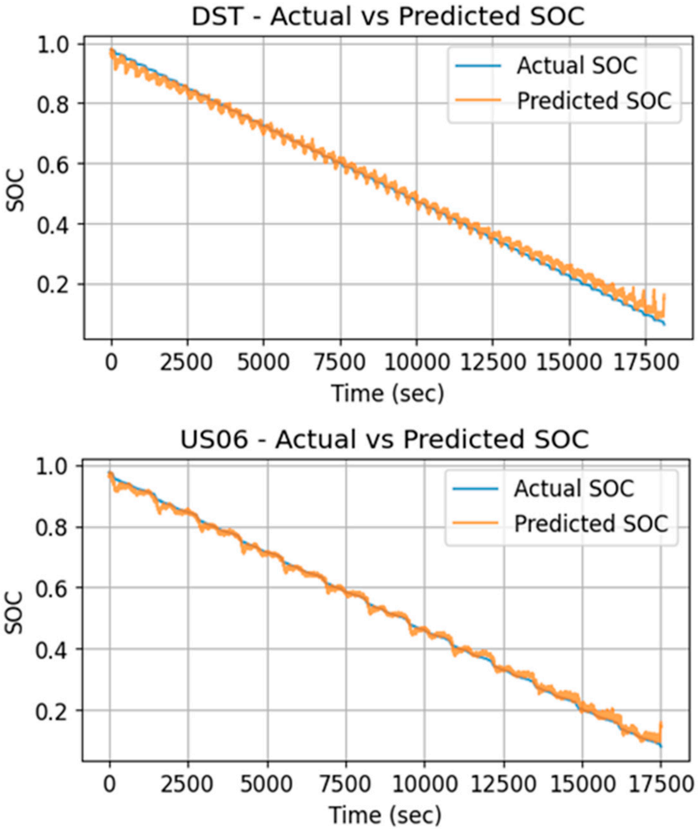

Figure 5. Test Results of US06 and DST at −5 °C.

(1) Test error of DST drive cycle

In the Table 2, it appears that implementing k-decay optimization on the convolutional neural network (CNN) led to improvements in the model's performance, particularly in terms of mean absolute error (MAE) and root mean squared error (RMSE) across different ambient temperature conditions. At an ambient temperature of 25 °C, the original CNN exhibited an MAE of 0.012 and RMSE of 0.016, whereas the k-decay-optimized CNN achieved a lower MAE of 0.0091 and RMSE of 0.013, indicating enhanced accuracy in estimating the state of charge (SoC) of the battery.

Table 2. The DST test error at different temperatures and comparison of architectures.

Model	**Ambient Temperature (25 °C)**		**Ambient Temperature (45 °C)**		**Ambient Temperature (5 °C)**		**Ambient Temperature (−5 °C)**	
	Test Error (%)		**Test Error (%)**		**Test Error (%)**		**Test Error (%)**	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
Original CNN	1.2	1.6	2.4	3.1	1.6	3.37	2.1	2.5
K-decay optimize CNN	0.91	1.3	1.9	2.3	0.93	1.2	1.5	1.8

Similarly, at higher ambient temperatures such as 45 °C, the k-decay-optimized CNN demonstrated superior performance compared to the original CNN, with a lower MAE of 0.019 and RMSE of 0.023, compared to the respective values of 0.024 and 0.031 obtained by the original CNN. Even at extreme ambient temperatures, such as 5 °C and −5 °C, the k-decay-optimized CNN showcased remarkable improvements in accuracy, with substantially reduced MAE and RMSE values compared to the original CNN. At 5 °C, the MAE decreased from 1.6 to 0.0093, and the RMSE decreased from 3.37 to 0.012. Similarly, at −50 °C, the MAE reduced from 0.021 to 0.015, and the RMSE decreased from 0.025 to 0.018.

Overall, the dynamic k-decay optimization technique has proven to be highly effective in enhancing the performance of the CNN model for state-of-charge estimation across a wide range of ambient temperature conditions, leading to more accurate and reliable predictions, which is crucial for battery management systems.

The SoC predictions in the CNN model for US06 and DST are shown in Figures 1–4.

(2) Test error at US06.

In Table 3, it is evident that applying k-decay optimization to the convolutional neural network (CNN) has resulted in significant improvements in the model's performance across various ambient temperature conditions. At an ambient temperature of 25 °C, the k-decay-optimized CNN achieved a mean absolute error (MAE) of 0.0081 and a root mean squared error (RMSE) of 0.0095, which are notably lower than the MAE of 0.012 and RMSE of 0.013 obtained by the original CNN. Similarly, at higher ambient temperatures such as 45 °C, the k-decay-optimized CNN exhibited superior performance with an MAE of 0.010 and RMSE of 0.012, compared to the respective values of 0.0195 and 0.023 achieved by the original CNN. Even at extreme ambient temperatures, such as 5 °C and −5 °C, the k-decay-optimized CNN outperformed the original CNN, demonstrating reduced MAE and RMSE values. At 5 °C, the MAE decreased from 0.021 to 0.0078, and the RMSE decreased from 0.025 to 0.010. Similarly, at −5 °C, the MAE decreased from 0.015 to 0.0090, and the RMSE decreased from 0.018 to 0.012.

Table 3. The US06 test error at different temperatures and comparison of architectures.

Model	**Ambient Temperature (25 °C)**		**Ambient Temperature (45 °C)**		**Ambient Temperature (5 °C)**		**Ambient Temperature (−5 °C)**	
	Test Error (%)		**Test Error (%)**		**Test Error (%)**		**Test Error (%)**	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
Original CNN	1.2	1.3	1.9	2.3	2.1	2.5	1.5	1.8
K-decay-optimized CNN	0.80	0.95	1.0	1.2	0.78	1.0	0.90	1.2

Overall, these results indicate that implementing k-decay optimization led to significant enhancements in CNN accuracy and reliability for state-of-charge estimation, making it a promising approach for improving battery management systems, particularly in diverse ambient temperature environments.

4.2. Comparison of Architecture and Computational Cost

This section provides a detailed comparison of different neural network architectures in terms of the number of parameters, training time, and performance metrics, specifically focusing on their application to estimating the state of charge (SoC) of batteries. The comparison in Table 4 includes the original CNN, the proposed CNN with dynamic k-decay learning rate optimization, LSTM, Bi-LSTM, and CNN–LSTM models. The original CNN model has 12,033 parameters and takes approximately 10.81 min (648.59 s) to train. It achieves a test error rate of 1.2%, with MAE and RMSE values of 1.3 and 12,033, respectively. This baseline model serves as a reference point for evaluating improvements in other architectures. The dynamic k-decay-optimized CNN maintains the same number of parameters (12,033) but significantly reduces the training time to 5.40 min (324.14 s). It achieves a lower test error rate of 0.80%, with improved MAE and RMSE values of 0.95 and 12,033, respectively. This demonstrates the efficiency and enhanced performance added by the optimization technique, making it a superior choice for SoC estimation. The LSTM model, with 15,900 parameters, requires 55 min to train and results in a higher test error rate of 0.90%, with MAE and RMSE values of 1.54 and 15,900, respectively. Despite its complexity and longer training time, it does not perform as well as the optimized CNN, indicating that LSTM may not be the most efficient model for this specific application. The Bi-LSTM model, having 42,701 parameters, takes 60 min to train and also yields a test error rate of 0.90%, with MAE and RMSE values of 1.50 and 42,701, respectively. Its performance is similar to the LSTM but with a larger parameter set and slightly longer training time, highlighting that increasing model complexity does not necessarily lead to better performance in this context. The CNN–LSTM model combines convolutional and recurrent layers, resulting in 56,849 parameters. It takes the longest to train at 130 min and produces a test error rate of 0.80%, with MAE and RMSE values of 1.37 and 56,849, respectively. While it shows a lower error rate, the increased computational cost and extended training time make it less efficient compared to the optimized CNN.

Table 4. Comparison of architecture and computational cost.

Model			**Number of Parameters**	**Training Time (min)**
	Test Error (%)			
	MAE	**RMSE**		
Original CNN	1.2	1.3	12,033	10.81 (648.59 s)
Dynamic K-decay-optimized CNN	0.80	0.95	12,033	5.40 (324.14 s)
LSTM	0.90	1.54	15,900	55
Bi-LSTM	0.90	1.50	42,701	60
CNN-LSTM	0.80	1.37	56,849	130

The comparison of different architectures and their computational costs provides valuable insights into their performance and efficiency. The original CNN architecture, despite its higher number of parameters, demonstrates a significant training time and moderate test error. However, the introduction of the dynamic k-decay-optimized CNN results in a reduction in parameters while achieving a notable improvement in both training time and test error rate. This optimization showcases the efficacy of architectural modifications in enhancing model efficiency without compromising performance. In contrast, the LSTM and Bi-LSTM [30] models exhibit longer training times and higher test errors compared to the CNN-based architectures, suggesting potential limitations in their suitability for the given task. Particularly, the CNN–LSTM hybrid model [30] stands out with the longest training time and the highest test error among all architectures, indicating potential inefficiencies in its design or implementation. Overall, the dynamic k-decay-optimized CNN emerges as the preferred choice, offering a compelling balance between model complexity, training efficiency, and performance accuracy.

In terms of the number of parameters, the dynamic k-decay-optimized CNN has the lowest, followed closely by the LSTM and Bi-LSTM models. However, the dynamic k-decay-optimized CNN achieves the lowest test error rate (MAE) among all models, indicating superior performance in estimating the state of charge. Additionally, it has the shortest training time, making it efficient in terms of computational resources. Conversely, the CNN–LSTM model exhibits the highest test error rate and longest training time, suggesting that it may be less suitable for this application compared to the other models.

In conclusion, the proposed dynamic k-decay-optimized CNN stands out as the most efficient model, striking a balance between low computational cost and high accuracy in SoC estimation. Its performance improvements and reduced training time highlight the effectiveness of the optimization technique, making it a highly practical choice for battery management system applications.

5. Conclusions

In this study, we evaluated the performance of various neural network architectures for state of charge (SoC) estimation in battery management systems. The architectures considered include the original CNN and dynamic k-decay-optimized CNN models. Each architecture was assessed based on several key metrics, including the number of parameters, training time, and test error rate, measured by the mean absolute error (MAE).

The dynamic k-decay-optimized CNN demonstrated promising results, boasting the lowest number of parameters and achieving a substantially reduced MAE compared to the original CNN. This optimized CNN architecture also exhibited a shorter training time, highlighting its efficiency in computational resources. Conversely, the LSTM and Bi-LSTM models, while having a slightly higher number of parameters, yielded higher MAE values, indicating less accurate SoC estimation.

The CNN–LSTM model, despite having a comparable number of parameters to the dynamic k-decay-optimized CNN, displayed the longest training time and the highest MAE, suggesting a tradeoff between computational complexity and accuracy. Although the Bi-LSTM and CNN–LSTM models offered relatively high complexity, their performance in terms of accuracy fell short compared to the dynamic k-decay-optimized CNN.

The proposed approach for state of charge (SoC) estimation in battery management systems exhibits limitations that warrant attention. These limitations include potential constraints due to limited dataset variability and size, which could hinder the model's ability to generalize to diverse real-world scenarios. Additionally, the model may struggle to perform effectively under extreme temperature conditions outside the training range, underscoring the need for strategies to enhance its resilience in such scenarios. Furthermore, the approach's applicability across different battery types may be limited, necessitating research into adaptable models capable of accommodating various battery chemistries and designs. Addressing these limitations could pave the way for future research to explore methods for augmenting dataset variability, developing adaptable models, enhancing resilience to extreme conditions, and assessing environmental impacts for more sustainable battery management practices.

In summary, the dynamic k-decay-optimized CNN emerges as the preferred architecture for SoC estimation in battery management systems due to its optimal balance between training efficiency and accuracy. With fewer parameters and a significantly lower MAE, this architecture demonstrates superior performance and efficiency compared to its counterparts, making it the most suitable choice for real-world implementations requiring accurate and efficient SoC estimation.

Author Contributions: Conceptualization, data curation, formal analysis, investigation, software, validation, methodology, writing—original draft, N.B.; Supervision, resources, project administration, S.M. and K.S.T.; Supervision, project administration, writing—review & editing, M.S. (Mohamed Shaaban); Funding acquisition, writing—review & editing, M.S. (Mehdi Seyedmahmoudian) and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is financially supported by Universiti Malaya Matching Grant: MG022-2022.

Data Availability Statement: Data are contained within the article.

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

References

1. Li, K.; Wei, F.; Tseng, K.J.; Soong, B.-H. A Practical Lithium-Ion Battery Model for State of Energy and Voltage Responses Prediction Incorporating Temperature and Ageing Effects. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6696–6708. [CrossRef]
2. Wu, F.; Wang, S.; Liu, D.; Cao, W.; Fernandez, C.; Huang, Q. An improved convolutional neural network-bidirectional gated recurrent unit algorithm for robust state of charge and state of energy estimation of new energy vehicles of lithium-ion batteries. *J. Energy Storage* **2024**, *82*, 110574. [CrossRef]
3. Bockrath, S.; Lorentz, V.; Pruckner, M. State of health estimation of lithium-ion batteries with a temporal convolutional neural network using partial load profiles. *Appl. Energy* **2023**, *329*, 120307. [CrossRef]
4. Qin, P.; Zhao, L. A novel transfer learning-based cell SOC online estimation method for a battery pack in complex application conditions. *IEEE Trans. Ind. Electron.* **2023**, *71*, 1606–1615. [CrossRef]
5. Wadi, A.; Abdel-Hafez, M.; Hashim, H.A.; Hussein, A.A. An Invariant Method for Electric Vehicle Battery State-of-Charge Estimation Under Dynamic Drive Cycles. *IEEE Access* **2023**, *11*, 8663–8673. [CrossRef]
6. Mazzi, Y.; Ben Sassi, H.; Errahimi, F. Lithium-ion battery state of health estimation using a hybrid model based on a convolutional neural network and bidirectional gated recurrent unit. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107199. [CrossRef]
7. Wang, Q.; Ye, M.; Wei, M.; Lian, G.; Li, Y. Deep convolutional neural network based closed-loop SOC estimation for lithium-ion batteries in hierarchical scenarios. *Energy* **2023**, *263*, 125718. [CrossRef]

8. Wang, X.; Hao, Z.; Chen, Z.; Zhang, J. Joint prediction of li-ion battery state of charge and state of health based on the DRSN-CW-LSTM model. *IEEE Access* **2023**, *11*, 70263–70273. [CrossRef]
9. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6730–6739. [CrossRef]
10. Shu, X.; Li, G.; Zhang, Y.; Shen, S.; Chen, Z.; Liu, Y. Stage of Charge Estimation of Lithium-Ion Battery Packs Based on Improved Cubature Kalman Filter with Long Short-Term Memory Model. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 1271–1284. [CrossRef]
11. How, D.N.T.; Hannan, M.A.; Lipu, M.S.H.; Sahari, K.S.M.; Ker, P.J.; Muttaqi, K.M. State-of-Charge Estimation of Li-Ion Battery in Electric Vehicles: A Deep Neural Network Approach. *IEEE Trans. Ind. Appl.* **2020**, *56*, 5565–5574. [CrossRef]
12. Zhao, R.; Kollmeyer, P.J.; Lorenz, R.D.; Jahns, T.M. A Compact Methodology Via a Recurrent Neural Network for Accurate Equivalent Circuit Type Modeling of Lithium-Ion Batteries. *IEEE Trans. Ind. Appl.* **2019**, *55*, 1922–1931. [CrossRef]
13. Hannan, M.A.; How, D.N.T.; Lipu, M.S.H.; Ker, P.J.; Dong, Z.Y.; Mansur, M.; Blaabjerg, F. SOC Estimation of Li-ion Batteries with Learning Rate-Optimized Deep Fully Convolutional Network. *IEEE Trans. Power Electron.* **2020**, *36*, 7349–7353. [CrossRef]
14. Liu, Y.; Xiao, B. Accurate state-of-charge estimation approach for lithium-ion batteries by gated recurrent unit with ensemble optimizer. *IEEE Access* **2019**, *7*, 54192–54202. [CrossRef]
15. Song, X.; Yang, F.; Wang, D.; Tsui, K.-L. Combined CNN-LSTM Network for State-of-Charge Estimation of Lithium-Ion Batteries. *IEEE Access* **2019**, *7*, 88894–88902. [CrossRef]
16. Bian, C.; Yang, S.; Miao, Q. Cross-Domain State-of-Charge Estimation of Li-Ion Batteries Based on Deep Transfer Neural Network with Multiscale Distribution Adaptation. *IEEE Trans. Transp. Electrif.* **2020**, *7*, 1260–1270. [CrossRef]
17. Bhattacharjee, A.; Verma, A.; Mishra, S.; Saha, T.K. Estimating State of Charge for xEV Batteries Using 1D Convolutional Neural Networks and Transfer Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3123–3135. [CrossRef]
18. Shu, X.; Shen, J.; Li, G.; Zhang, Y.; Chen, Z.; Liu, Y. A Flexible State-of-Health Prediction Scheme for Lithium-Ion Battery Packs with Long Short-Term Memory Network and Transfer Learning. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 2238–2248. [CrossRef]
19. Qin, Y.; Adams, S.; Yuen, C. Transfer Learning-Based State of Charge Estimation for Lithium-Ion Battery at Varying Ambient Temperatures. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7304–7315. [CrossRef]
20. Cheng, X.; Liu, X.; Li, X.; Yu, Q. An intelligent fusion estimation method for state of charge estimation of lithium-ion batteries. *Energy* **2024**, *286*, 129462. [CrossRef]
21. Demirci, O.; Taskin, S.; Schaltz, E.; Demirci, B.A. Review of battery state estimation methods for electric vehicles—Part I: SOC estimation. *J. Energy Storage* **2024**, *87*, 111435. [CrossRef]
22. Hu, X.; Liu, X.; Wang, L.; Xu, W.; Chen, Y.; Zhang, T. SOC Estimation Method of Lithium-Ion Battery Based on Multi-innovation Adaptive Robust Untraced Kalman Filter Algorithm. In Proceedings of the 2023 5th International Conference on Power and Energy Technology, ICPET 2023, Tianjin, China, 27–30 July 2023; pp. 321–328. [CrossRef]
23. Ghaeminezhad, N.; Ouyang, Q.; Wei, J.; Xue, Y.; Wang, Z. Review on state of charge estimation techniques of lithium-ion batteries: A control-oriented approach. *J. Energy Storage* **2023**, *72*, 108707. [CrossRef]
24. Tang, A.; Huang, Y.; Xu, Y.; Hu, Y.; Yan, F.; Tan, Y.; Jin, X.; Yu, Q. Data-physics-driven estimation of battery state of charge and capacity. *Energy* **2024**, *294*, 130776. [CrossRef]
25. Korkmaz, M. A novel approach for improving the performance of deep learning-based state of charge estimation of lithium-ion batteries: Choosy SoC Estimator (ChoSoCE). *Energy* **2024**, *294*, 130913. [CrossRef]
26. Smith, L.N. Cyclical learning rates for training neural networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472. [CrossRef]
27. Loshchilov, I.; Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017—Conference Track Proceedings, Toulon, France, 24–26 April 2017; pp. 1–16.
28. Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.
29. Zhang, T.; Li, W. k-decay: A New Method for Learning Rate Schedule. *arXiv* **2020**, arXiv:2004.05909.
30. Sherkatghanad, Z.; Ghazanfari, A.; Makarenkov, V. A self-attention-based CNN-Bi-LSTM model for accurate state-of-charge estimation of lithium-ion batteries. *J. Energy Storage* **2024**, *88*, 111524. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.