

Article

A Novel Deep Learning-Based State-of-Charge Estimation for Renewable Energy Management System in Hybrid Electric Vehicles

Mahendiran T. Vellingiri ^{1,*}, Ibrahim M. Mehedi ^{1,2}  and Thangam Palaniswamy ¹

¹ Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia; imehedi@kau.edu.sa (I.M.M.); tswamy@kau.edu.sa (T.P.)

² Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, Jeddah 21589, Saudi Arabia

* Correspondence: mvellingiri@kau.edu.sa

Abstract: In recent years, alternative engine technologies are necessary to resolve the problems related to conventional vehicles. Electric vehicles (EVs) and hybrid electric vehicles (HEVs) are effective solutions to decarbonize the transportation sector. It also becomes important to shift from traditional houses to smart houses and from classical vehicles to EVs or HEVs. It is needed to combine renewable energy sources (RESs) such as solar photovoltaics, wind energy systems, and various forms of bio-energies. Among various HEV technologies, an effective battery management system (BMS) still remains a crucial issue that is majorly used for indicating the battery state of charge (SOC). Since over-charging and over-discharging result in inevitable impairment to the batteries, accurate SOC estimation desires to be presented by the BMS. Although several SOC estimation techniques exist to regulate the SOC of the battery cell, it is needed to improvise the SOC estimation performance on HEVs. In this view, this paper focuses on the design of a novel deep learning (DL) with SOC estimation model for secure renewable energy management (DLSOC-REM) technique for HEVs. The presented model employs a hybrid convolution neural network and long short-term memory (HCNN-LSTM) model for the accurate estimation of SOC. In order to improve the SOC estimation outcomes of the HCNN-LSTM model, the barnacles mating optimizer (BMO) is applied for the hyperpower tuning process. The utilization of the HCNN-LSTM model makes the modeling process easier and offers a precise depiction of the input–output relationship of the battery model. The design of BMO based HCNN-LSTM model for SOC estimation shows the novelty of the work. An extensive experimental analysis highlighted the supremacy of the proposed model over other existing methods in terms of different aspects.

Keywords: hybrid electric vehicles; state of charge estimation; renewable energy; deep learning; metaheuristics



Citation: 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. <https://doi.org/10.3390/math10020260>

Academic Editor: Liangxiao Jiang

Received: 21 December 2021

Accepted: 10 January 2022

Published: 15 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, energy deficiency and ecological degradation have become important global issues, particularly increasing the stricter necessities on vehicle engineering. For reducing waste gas emissions and reproducing energy in driving processes, few new energy vehicles have been presented for replacing conventional vehicles functioning with gasoline, such as electric vehicles (EVs) and hybrid electric vehicles (HEVs). The HEV has presently treated the highly feasible alternate propulsion system [1] over EV, with the benefit of high speed and long-distance travel. The effective battery management system (BMS) of HEVs acts as a major part in achieving vehicle safety, extending battery life, reducing cost, and improving driving range [2]. Many BMSs are placed for equalization management, fault diagnosis, etc. Amongst them, the major process of the sampling circuit is to determine the voltage, current, and temperature signals. This is followed by the control circuit utilizing the signals

to estimate the state of charge (SOC), state of health (SOH), state of power (SOP), and state of life (SOL) of batteries. Meanwhile, the controller of the vehicles deals with the circuits and realizes the management of the automobile energy and power distribution [3]. The SOC of batteries in a BMS is comparable to the fuel meter in a traditional fuel car. The major operation of the SOC is to interconnect the instinctive battery state to the driver and prevent issues such as overcharging and over-discharging [4].

SOC approximation of Li-ion batteries can be commonly determined by the use of three approaches such as conventional, model-based, and machine learning (ML) approaches. The classical models are simpler ones but are not suitable for online operation [5]. Furthermore, the model-based approaches are highly efficient in modeling the characteristics of the Li-ion batteries precisely. However, they find difficulty in the design of an effective model for Li-ion battery SOC estimation [6]. Alternatively, the ML-based SOC estimation techniques make use of the influx of data and efficient processors in estimating the SOC with restricted earlier knowledge regarding the interesting features of the battery and chemical reaction [7]. However, the efficiency of ML models is mainly based on the quantity as well as the quality of the training data. Conversely, recently developed deep learning (DL) models can be utilized for effective SOC estimation. The authors in [8] presented an LSTM model for SOC estimation. The observed voltage, current, and temperature are immediately passed into the derived network that learns the mapping among the input time series and the target SOC. Likewise, the SOC estimation by the use of LSTM [9] and the gated recurrent neural network. An unscented Kalman filter (UKF) is derived for smoothing the prediction outcomes of the LSTM. These works have offered precise SOC estimates for varying dynamic profiles, such as the US06 test, dynamic stress test (DST), and federal urban driving schedule (FUDS). However, the actual EV driving profile of the batteries is distinct from the standard dynamic profile, as they differ in various regions, drivers, and durations [10].

This paper presents an effective deep learning (DL) based SOC estimation model for renewable energy management (DLSOC-REM) technique for HEVs. The presented model involves the design of a hybrid convolution neural network and long short-term memory (HCNN-LSTM) based predictive model for accurate SOC estimation. Moreover, the barnacles mating optimizer (BMO) is used for the proper hyper-parameter tuning of the HCNN-LSTM model in such a way that the accurate estimation of SOC can be achieved. In order to demonstrate the enhanced performance of the DLSOC-REM technique, a series of simulations take place and the results are inspected under various aspects. In short, the contributions of this works are summarized as follows.

- Development of a novel DLSOC-REM technique for the SOC estimation in HEVs.
- Aims to design a new HCNN-LSTM model for the prediction of SOC estimation and proper management of renewable energy.
- Employs the BMO for the hyperparameter tuning of the HCNN-LSTM model and thereby improves the prediction performance.
- Validate the performance of the DLSOC-REM technique using different drive cycles and inspect the results under several aspects.

The rest of the study is arranged as follows. Section 2 provides a detailed survey of existing SOC estimation techniques. Then, Section 3 provides the comprehensive working of the DLSOC-REM technique and Section 4 elaborates and analyzes the results. Lastly, Section 5 draws the concluding remarks of the study.

2. Literature Review

Zhang et al. [11] developed a particle filter-based hybrid filtering approach specifically developed to estimate the SOC of Li-ion cells in EV. The sampling importance resampling particle filtering approach can be used in integrating with the conventional Kalman filter (KF) and unscented KF as a proposal dispersal for the particle filter to make it precise and fast. In Zahid et al. [12], a novel SOC estimation technique by the use of subtractive clustering enabled neuro-fuzzy system was developed and simulated by the use of an advanced

vehicle simulator related to BPNN and Elman neural network (ENN). In Lai et al. [13], an efficient SOC estimation technique using the SOC increment with maximum credibility was presented for dealing with huge errors of sensors and models. Primarily, the features of the SOC error increment related to the ampere-hour counting (AHC) and extended Kalman filter (EKF) approaches have been investigated, and SOC increment with high credibility was computed. Then, a hybrid SOC estimation approach was developed which makes use of the AHC and EKF methods.

Veerendra et al. [14] assessed the possible improvements in fuel economy and efficiency of a fuel-cell series hybrid electric vehicle (FCHEV) integrated into a supercapacitor (SC) utilizing hybrid power management (HPM) approach. Integration of the EKF and conventional Coulomb counting (CC) can be employed for SOC estimation. Chandran et al. [15] presented an efficient SOC estimation model for Li-ion batteries by the use of 6 ML models such as ANN, SVM, LR, Gaussian process regression (GPR), ensemble bagging (EBa), and ensemble boosting (EBo). A detailed error analysis of the method was performed for the optimization of the battery performance parameter. How et al. [16] presented an effective SOC estimation approach for a Li-ion battery by the use of an enhanced DNN model for electric vehicle application. It was identified that the DNN with an adequate number of hidden layers is capable of predicting the SOC of the unseen drive cycles at the time of training. A set of DNN methods were trained with a distinct number of hidden layers and the training process was investigated under varying drive cycles. In Tian et al. [17], a DNN model was presented for SOC estimation in 10 min charging voltage and current data as the input. This allowed designing a rapid and precise SOC estimate model with an error rate of <2.03% over the whole range of SOC batteries. Therefore, it can be applied for calibrating the SOC estimation for the Ampere-hour counting approach. It is noted that the integration of the DNN with the Kalman filter can result in highly robust performance.

3. The Proposed SOC Estimation Model

This paper has developed an effective DLSOC-REM technique for accurate SOC estimation in HEVs. The DLSOC-REM technique encompasses two major processes such as HCNN-LSTM based prediction and BMO based hyperparameter tuning. At the initial stage, the input and output of the DLSOC-REM technique can be determined. Based on the concepts of NN, the sampling process of the SOC taking place at step k , $SOC(k)$ can be considered as the model input as it denotes the current condition of the battery. It has a non-linear relationship with influencing factors such as battery current and voltage [18]. The direct parameter current $I(k)$ is considered the input and the battery terminal voltage $V(k)$ can be represented as the output. In addition, the terminal voltage at sampling step $k - 1$, $V(k - 1)$ can be selected as the third input for the DLSOC-REM method. $v(k - 1)$ signifies the condition of the battery at the final stage and it denotes the earlier working condition. The terminal voltage at step k can be denoted as:

$$V(k) = OCV(SOC(k)) + R_s I(k) + U_{RC}(k) \quad (1)$$

where R_s signifies the battery internal resistance, U_{RC} signifies the RC circuit voltage related to $U_{RC}(k - 1)$ using the first-order differential equation [19]. For making the model input directly measured variables, $U_{RC}(k - 1)$ can be considered in $V(k - 1)$. Therefore, $V(k - 1)$ holds a direct connection with $V(k)$. A function can be attained by the synthetization of the unknown variables.

$$V(k) = f(V(k - 1), I(k), SOC(k)), \quad (2)$$

Which undergoes approximation using the learning model. The input and output vectors of the presented battery method are defined as $p(k) = [V(k - 1) \ I \ SOC(k)]^T$ and $V(k)$.

$$F(p(k)) = V(k) \quad (3)$$

The input output instances of $\{p(k) \sim V(k)\}$ can be attained prior to training the model. Consider $p(k) = x_j$ and $V(k) = t_j$. The training set can be represented by $\{(x_j, t_j) | x_j \in R^n, t_j \in R^m, j = 1, \dots, N\}$.

3.1. Design of HCNN-LSTM Based SOC Estimation

In this study, the HCNN-LSTM model was utilized for the SOC estimation in the HEVs. In the HCNN-LSTM model, the CNN layer is applied to extract the patterns in an automated manner. The sequence of features is learned again from the LSTM layer. The HCNN-LSTM model continually adjusts the hyperparameters based on the outcomes from the learning process of the CNN and LSTM methods. The CNN method is utilized for the extraction of correlations that exist in the data and derive variables that are needed for the classification process, which can be performed by the class activation map. Equation (4) represents the convolution operation l to derive a set of features. The convolution process carries out a product operation on the trained dataset by the use of a feature map of size m_1^{l-1} . The kernel $K_{i,j}^l$ denotes various weights in every region in extracting the significant regions of the feature map. Moreover, the correlation among the nearby features is derived via the product operations. Moreover, the bias matrix B_i^l can be utilized for altering the weight from the NN operation. The product operation is carried out on the count of feature maps m_1^{l-1} and passes y_i^l to the subsequent convolution layer [20]. Figure 1 illustrates the framework of the CNN-LSTM technique.

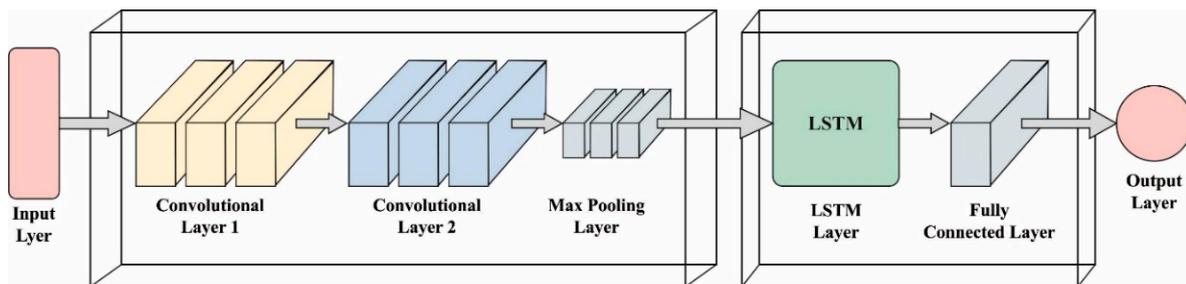


Figure 1. Structure of CNN-LSTM.

For creating a non-linear decision boundary, $f(z)$ in Equation (5) is an activation function such as ReLU utilized from layer l . The extraction of features takes place via several layers of the convolution operation.

$$x_i^l = B_i^l + \sum_{j=1}^{m_1^{l-1}} K_{ij}^l * X_j^{l-1} \tag{4}$$

$$y_i^l = g_i f(y_i^{l-1}), f(z) = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases} \tag{5}$$

The pooling layer can be utilized for improvising the classifier outcome and reducing the computational cost. Equation (6) signifies the pooling layer function, which enables to reduce overfitting and proficiently derive features. T indicates the stride and R denotes the size of the pooling area.

$$p_{ij}^l = \max_{r \in R} Y_{i \times T + r, j}^{l-1} \tag{6}$$

For modeling the sequential data, the LSTM model can be employed for storing temporal details. The LSTM model is mainly employed for learning temporal data using the features derived from the CNN model. Equation (7) denotes the three gate states which accomplish the LSTM operation which manages the sequential data as a continuous value in the range of 0 and 1. Each individual cell holds input, output, and forget gates. Equation (7) can be defined by the output values of i, f , and o for every gate. Moreover,

for storing long-term data, the hidden state h_r of the LSTM cell is transcribed for each r step. Equation (9) denotes the hidden state of the LSTM. Lastly, Equation (9) displays the cell state to transfer the state from present to succeeding ones in the LSTM model. Here, each individual cell stores the weight W vector and adjusts the bias b vector value. The activation function σ , similar to sigmoid and hyperbolic tangents, can be utilized for the generation of non-linear decision boundaries.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \text{tanh} \end{pmatrix} w^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \end{pmatrix} \tag{7}$$

$$c_r = f_t^O c_{t-1} + i_t^O g \tag{8}$$

$$h_t = o_t^O \sigma(c_t) \tag{9}$$

Equation (10) indicates the function of the fully connected layer. The outcome of the FC layer can be categorized into 0 or 1 using the softmax function.

$$d_i^l = \sum_j \sigma(W_{ji}^{l-1}(h_j^{l-1}) + b_i^{l-1}) \tag{10}$$

$$P(c|d) = \operatorname{argmax}_{c \in C} \frac{\exp(d^{L-1}w^L)}{\sum_{k=1}^{N_c} \exp(d^{L-1}w_k)} \tag{11}$$

where C indicates the class, L is the former layer index, and N_c is the total number of classes.

3.2. Design of BMO Based Hyperparameter Optimization

The hyperparameters of the HCNN-LSTM model such as the number of filters, layers, and filter sizes can be adjusted by using the BMO algorithm. Barnacles are microbes that become attached to the objects in the water. Barnacles have a long penis, and they undergo mating with every neighbor and competitor in reach of their penis. The BMO algorithm is stimulated by the mating procedure of the barnacles [21]. Primarily, the candidate solutions are considered as the barnacles and the population can be initialized by the use of Equation (12). The validation of the population and storing procedure take place in locating the global solutions attained from the top of X . Following this, the parents to be mated can be chosen using Equations (13) and (14).

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^n \\ \vdots & \ddots & \vdots \\ x_N^1 & \cdots & x_N^n \end{bmatrix} \tag{12}$$

$$\text{barnacle_d} = \text{randperm}(N) \tag{13}$$

$$\text{barnacle_m} = \text{randperm}(N) \tag{14}$$

where N indicates the total barnacle population, n represents the control variable count, and barnacle_d and barnacle_m signify the mating parents. As there is no particular equation in deriving the reproduction process of the barnacles, the BMO algorithm will emphasize the genotype frequency of the parents in producing the offspring depending on the Hardy-Weinberg principle. It can be noted that the length of the penis (pl) acts as a vital part to determine the exploitation and exploration process. When the choice of barnacles undergoes mating in the limit of the pl of *Dad* barnacle, the exploitation procedure takes place. Equation (15) is presented in producing new parameters of offspring from the barnacles' parent.

$$x_i^{N-new} = p x_{\text{barnacle_d}}^N + q x_{\text{barnacle_m}}^N \tag{15}$$

where p indicates the arbitrary number in the uniform distribution of $[0, 1]$, $q = (1 - p)$, $x_{barnacle_d}^N$ and $x_{barnacle_m}^N$ denotes the parameters of *Dad* and *Mum* barnacles which can be chosen using Equations (13) and (14). p and q indicate the genotype frequency of *Dad* and *Mum* barnacles in the new offspring. Then, the generation of offspring takes place using the sperm casting process, which can be treated as the exploration procedure, as given in the following:

$$x_i^{n-new} = rand() \times x_{barnacle_m}^n \tag{16}$$

where $rand()$ is the random number between $[0, 1]$ From the above equation, it is noted that the newly created offspring from the mother barnacle as it attains the sperms which are released into the water by other barnacles [22]. During the iteration process, the location of the barnacles is upgraded based on Equation (15) or Equation (16). Lastly, the BMO can be represented for the approximation of the global optimum for optimization problems. Figure 2 demonstrates the flowchart of the BMO technique. The BMO algorithm derives an objective function based on mean square error (MSE) and it is used to predict the testing output of the HCNN-LSTM model. It can be defined as follows.

$$MSE = \frac{\sum_N^i |y_i - \hat{y}_i|^2}{N} \tag{17}$$

where y indicates the number of rounds, y_i denotes the experimental value, and \hat{y}_i represents the predicted values, respectively.

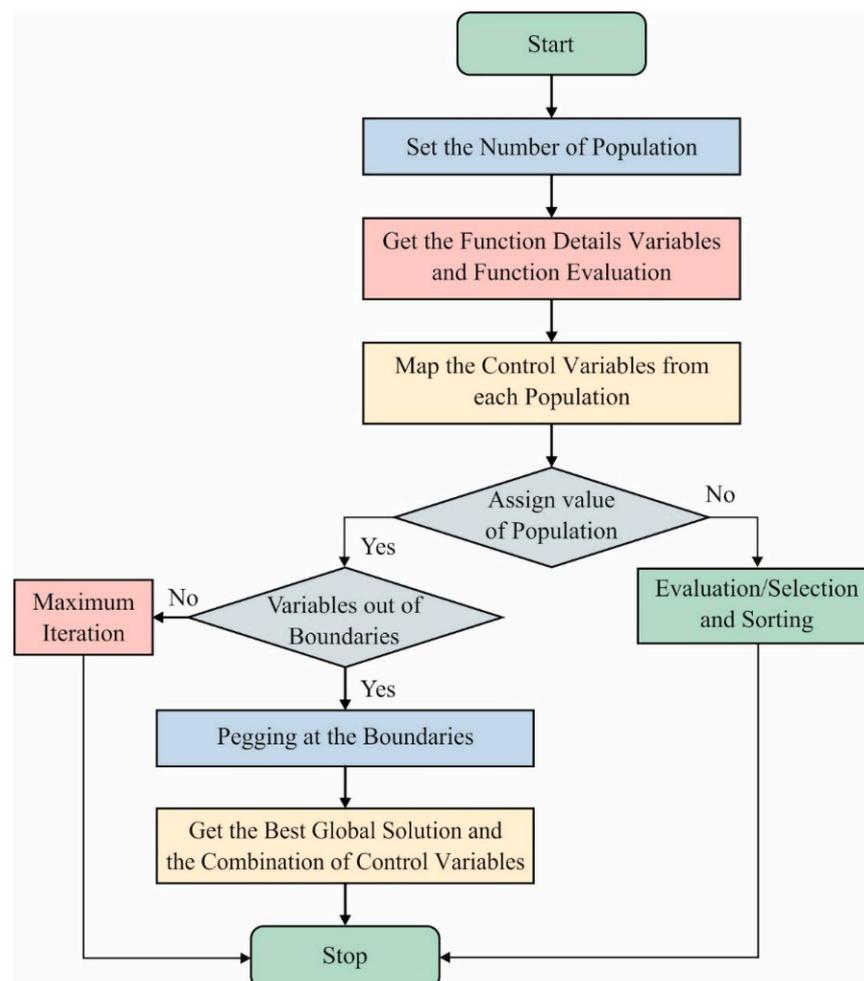


Figure 2. BMO Flowchart.

4. Performance Validation

The performance validation of the proposed model takes place using FUDS, BJDST, and US06 drive cycles. The results are inspected under various measures such as MSE, RMSE, MAE, and MAPE. Table 1 and Figure 3 provide the result analysis of the proposed model under distinct drive cycles and epoch count. The results show that the DLSOC-REM technique has resulted in effective SOC estimation results with the least values of MSE, RMSE, MAE, and MAPE. On the test BJDST drive cycle, with 10 epochs, the DLSOC-REM technique has obtained MSE, RMSE, MAE, and MAPE of 0.008921, 0.094451, 0.296270, and 3.188007, respectively. Similarly, with 60 epochs, the DLSOC-REM technique has attained MSE, RMSE, MAE, and MAPE of 0.008759, 0.093590, 0.293568, and 3.158928, respectively. Likewise, with 100 epochs, the DLSOC-REM technique has achieved MSE, RMSE, MAE, and MAPE of 0.008699, 0.093268, 0.292560, and 3.148090, respectively. Simultaneously, under the test US06 drive cycle, with 10 epochs, the DLSOC-REM technique has obtained MSE, RMSE, MAE, and MAPE of 0.001922, 0.043841, 0.137517, and 0.280993, respectively. Concurrently, under the test FUDS drive cycle, with 10 epochs, the DLSOC-REM technique has obtained MSE, RMSE, MAE, and MAPE of 0.010348, 0.101725, 0.319087, and 4.359018, respectively.

Table 1. SOC estimation results analysis of DLSOC-REM technique with different epochs.

Drive Cycle	No. of Epochs	MSE	RMSE	MAE	MAPE	Rank
BJDST	Epoch-10	0.008921	0.094451	0.296270	3.188007	10
	Epoch-20	0.008601	0.092742	0.290908	3.130307	4
	Epoch-30	0.008568	0.092563	0.290349	3.124296	2
	Epoch-40	0.008569	0.092569	0.290366	3.124478	3
	Epoch-50	0.008713	0.093343	0.292796	3.150622	7
	Epoch-60	0.008759	0.093590	0.293568	3.158928	8
	Epoch-70	0.008848	0.094064	0.295055	3.174936	9
	Epoch-80	0.008661	0.093064	0.291921	3.141206	5
	Epoch-90	0.008525	0.092331	0.289620	3.116446	1
	Epoch-100	0.008699	0.093268	0.292560	3.148090	6
	Average	0.008686	0.093199	0.292341	3.145732	
US06	Epoch-10	0.001922	0.043841	0.137517	0.280993	4
	Epoch-20	0.001885	0.043417	0.136187	0.278276	3
	Epoch-30	0.002204	0.046947	0.147261	0.300902	7
	Epoch-40	0.002048	0.045255	0.141953	0.290058	6
	Epoch-50	0.001633	0.040410	0.126758	0.259008	1
	Epoch-60	0.002474	0.049739	0.156020	0.233493	10
	Epoch-70	0.002380	0.048785	0.153027	0.227369	8
	Epoch-80	0.002428	0.049275	0.154563	0.218241	9
	Epoch-90	0.001983	0.044531	0.139683	0.197230	5
	Epoch-100	0.001738	0.041689	0.130769	0.177651	2
	Average	0.002070	0.045389	0.142374	0.246322	

Table 1. Cont.

Drive Cycle	No. of Epochs	MSE	RMSE	MAE	MAPE	Rank
FUDS	Epoch-10	0.010348	0.101725	0.319087	4.359018	7
	Epoch-20	0.010417	0.102064	0.320149	4.373526	9
	Epoch-30	0.010363	0.101799	0.319318	4.362176	8
	Epoch-40	0.010327	0.101622	0.318763	4.354592	5
	Epoch-50	0.009930	0.099649	0.312576	4.270070	3
	Epoch-60	0.010465	0.102299	0.320886	4.383591	10
	Epoch-70	0.009826	0.099126	0.310935	4.247651	2
	Epoch-80	0.010342	0.101696	0.318994	4.357754	6
	Epoch-90	0.010003	0.100015	0.313723	4.285737	4
	Epoch-100	0.009782	0.098904	0.310238	4.238130	1
	Average		0.010180	0.100890	0.316467	4.323224

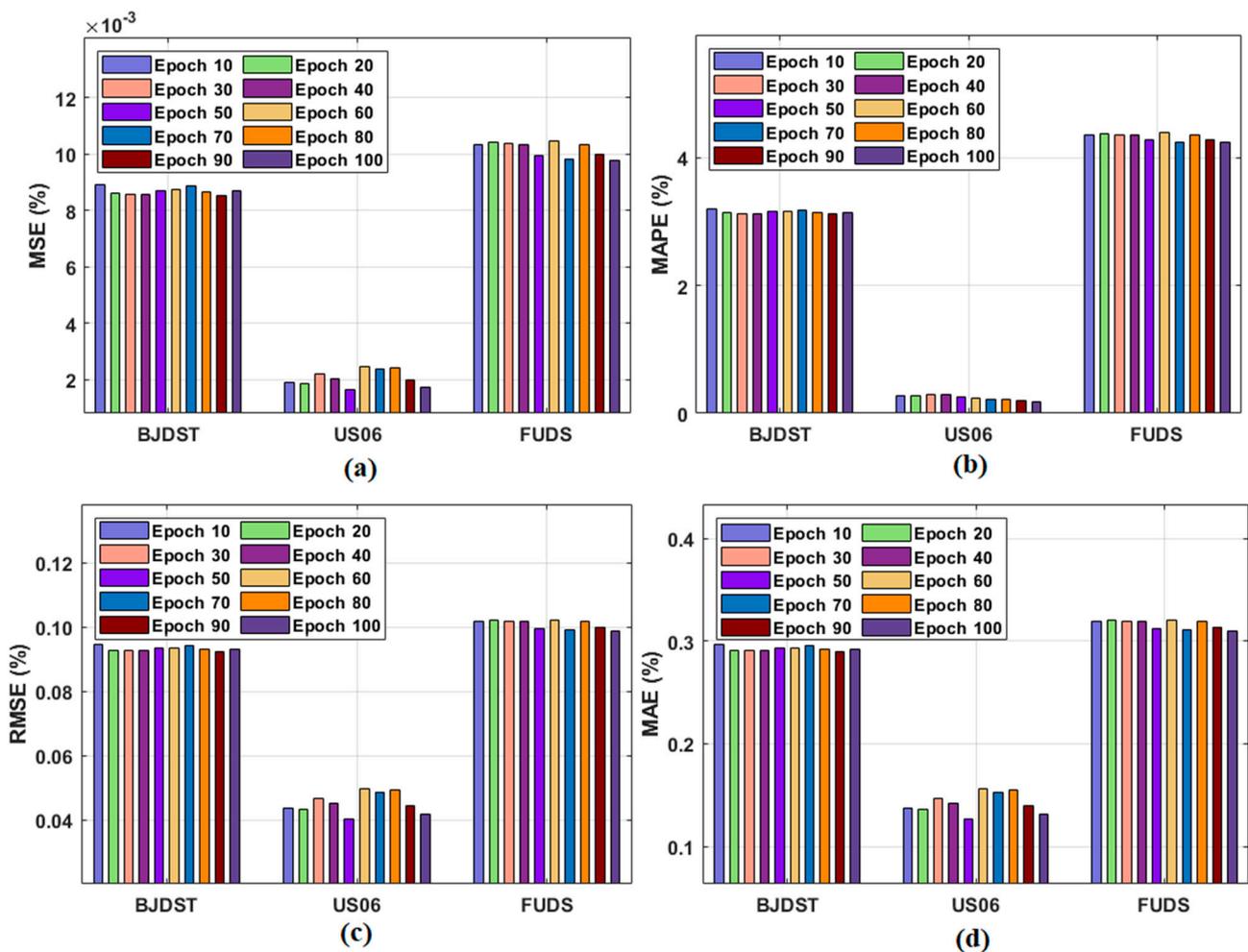


Figure 3. Result analysis of DLSOC-REM technique with varying epochs and drive cycles. (a) MSE, (b) MAPE, (c) RMSE, and (d) MAE.

Table 2 provides a detailed comparative result analysis of the DLSOC-REM technique with recent methods [23]. Figure 4 shows the results offered by the DLSOC-REM technique with existing techniques under the BJDST drive cycle. The figure shows that the DLSOC-

REM technique has accomplished effective outcomes with lower values of MSE, RMSE, and SOC error.

Table 2. Comparative result analysis of DLSOC-REM technique with recent approaches under different drive cycles.

Drive Cycle	Model	Temperature	RMSE (%)	MSE (%)	SOC Error (%)
BJDST	BPNN-GSA	0 °C	0.108628	0.011800	[−5.08 + 5.18]
		25 °C	0.950000	0.008700	[−5.2 + 5.1]
		45 °C	0.870000	0.007200	[−4.8 + 4.6]
	RBFNN-GSA	0 °C	0.110454	0.012200	[−6.51 + 6.87]
		25 °C	1.120000	0.011000	[−6.6 + 6.8]
		45 °C	0.970000	0.009200	[−5.3 + 4.9]
	ELM-GSA	0 °C	0.095917	0.009200	[−2.99 + 3.7]
		25 °C	0.760000	0.005900	[−3.1 + 3.6]
		45 °C	0.680000	0.005200	[−2.9 + 3]
	DLSOC-REM	0 °C	0.093199	0.008686	[−1.99 + 2.5]
		25 °C	0.075895	0.005760	[−2.51 + 2.8]
		45 °C	0.071063	0.005050	[−2.7 + 2.9]
US06	BPNN-GSA	0 °C	0.184391	0.034000	[−6.14 + 8.76]
		25 °C	1.680000	0.032000	[−6.2 + 8.7]
		45 °C	1.380000	0.018000	[−5.1 + 5.7]
	RBFNN-GSA	0 °C	0.303480	0.092100	[−8.61 + 9.25]
		25 °C	2.200000	0.089000	[−8.7 + 9.2]
		45 °C	1.880000	0.064000	[−6.3 + 6.8]
	ELM-GSA	0 °C	0.164317	0.027000	[−5.49 + 6.58]
		25 °C	1.560000	0.024000	[−5.6 + 6.5]
		45 °C	1.200000	0.014000	[−4.8 + 5.3]
	DLSOC-REM	0 °C	0.045389	0.002070	[−4.87 + 5.1]
		25 °C	0.154499	0.023870	[−5.3 + 5.5]
		45 °C	0.117686	0.013850	[−4.7 + 4.9]
FUDS	BPNN-GSA	0 °C	0.187350	0.035100	[−6.06 + 8.88]
		25 °C	0.181659	0.033000	[−6.15 + 8.77]
		45 °C	0.077460	0.006000	[−5.12 + 5.71]
	RBFNN-GSA	0 °C	0.281069	0.079000	[−8.51 + 9.35]
		25 °C	0.279285	0.078000	[−8.61 + 9.29]
		45 °C	0.236643	0.056000	[−6.3 + 6.85]
	ELM-GSA	0 °C	0.139284	0.019400	[−5.57 + 6.68]
		25 °C	0.126491	0.016000	[−5.69 + 6.58]
		45 °C	0.144914	0.021000	[−4.78 + 5.23]
	DLSOC-REM	0 °C	0.100890	0.010180	[−5.1−5.3]
		25 °C	0.126095	0.015900	[−4.2 + 4.7]
		45 °C	0.076551	0.005860	[−4.6 + 5]

Furthermore, the RBFNN-GSA technique obtained poor performance with the maximum values of MSE, RMSE, and SOC error. At the same time, the BPNN-GSA and ELM-GSA techniques offer somewhat closer values of MSE, RMSE, and SOC error. However, the DLSOC-REM technique reached superior performance with minimal RMSE, MSE, and SOC error at 45 °C of 0.071063, 0.005050, and $[-2.7 + 2.9]$, respectively.

Figure 5 illustrates the results of the DLSOC-REM technique with recent approaches under the US06 drive cycle. The figure shows that the DLSOC-REM algorithm accomplished effective outcomes with lesser values of MSE, RMSE, and SOC error. Moreover, the RBFNN-GSA system attained the worst performance with maximal values of MSE, RMSE, and SOC error. In addition, the BPNN-GSA and ELM-GSA techniques offered somewhat closer values of MSE, RMSE, and SOC error. However, the DLSOC-REM methodology gained superior performance with lower RMSE, MSE, and SOC error at 45 °C of 0.117686, 0.013850, and $[-4.7 + 4.9]$ correspondingly.

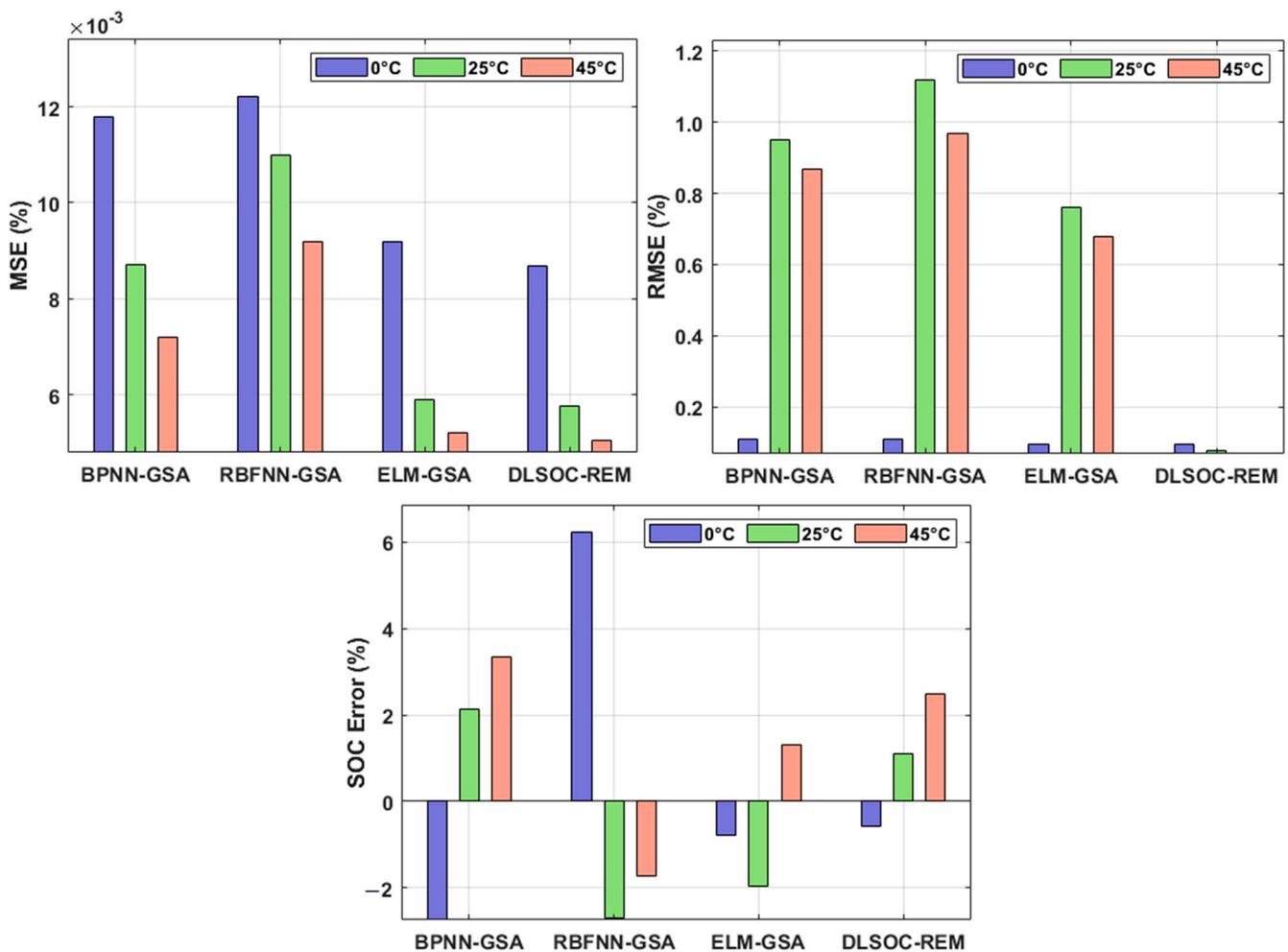


Figure 4. Comparative SOC estimation analysis of DLSOC-REM technique under BJDST drive cycle.

Figure 6 depicts the results obtained by the DLSOC-REM algorithm with existing methods under the FUDS drive cycle. The figure shows that the DLSOC-REM system accomplished effectual outcomes with minimum values of MSE, RMSE, and SOC error. This was followed by the RBFNN-GSA approach which obtained the least performance with increased values of MSE, RMSE, and SOC error. Simultaneously, the BPNN-GSA and ELM-GSA techniques show slightly closer values of MSE, RMSE, and SOC error. Eventually, the DLSOC-REM algorithm obtained higher performance with lower RMSE, MSE, and SOC error at 45 °C of 0.076551, 0.005860, and $[-4.6 + 5]$ correspondingly.

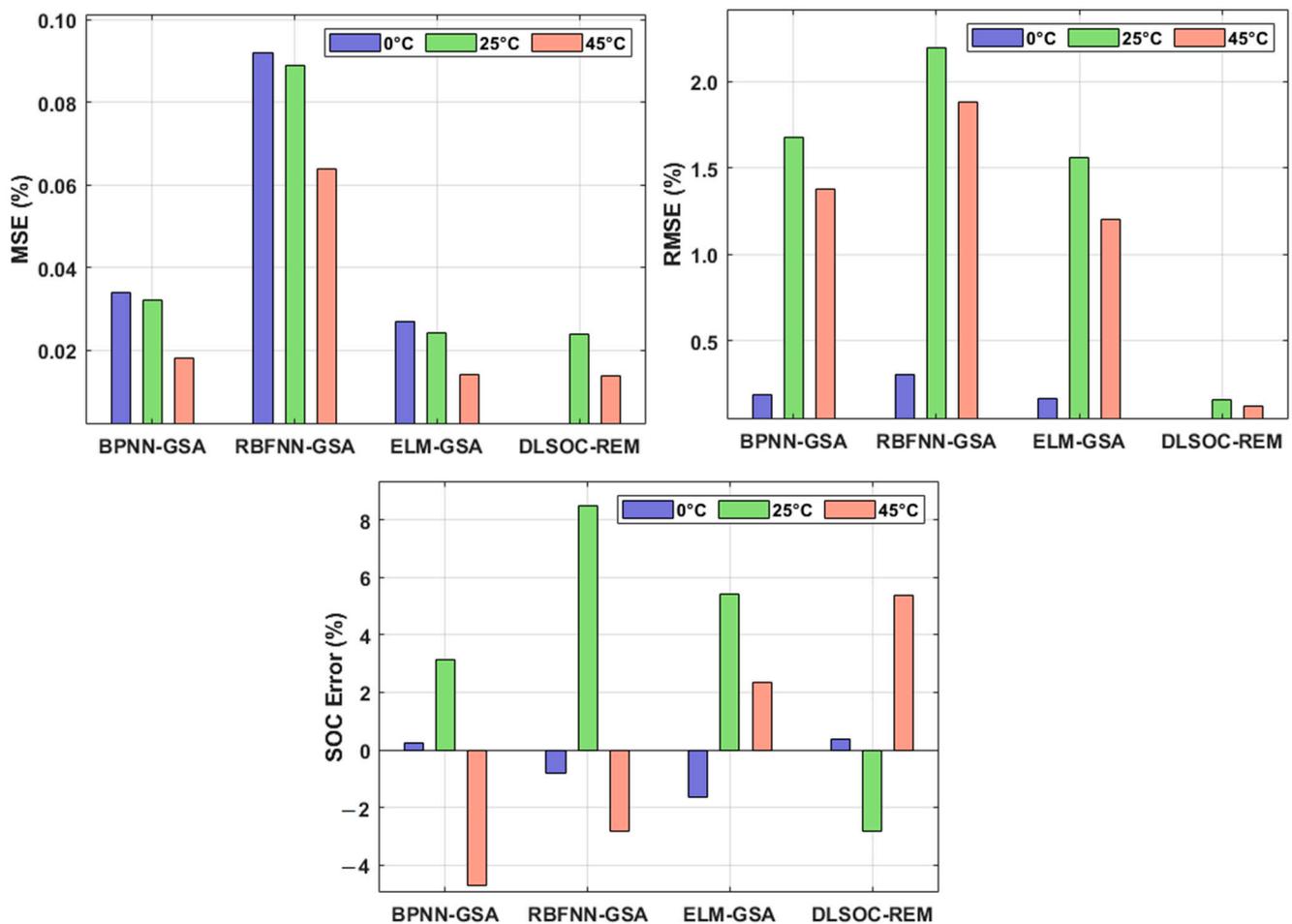


Figure 5. Comparative SOC estimation analysis of DLSOC-REM technique under US06 drive cycle.

Figure 7 shows the result analysis of the DLSOC-REM technique on the test BJDST drive cycle. The figure shows the estimated values of SOC by the DLSOC-REM technique and the actual SOC values. The results demonstrated that the SOC values remain high at the earlier time step and tend to be low with a decrease in time step. It is also evident that the DLSOC-REM technique estimated the SOC values which are much closer to the actual SOC values.

Figure 8 demonstrates the result analysis of the DLSOC-REM system on the test US06 drive cycle. The figure outperformed the estimated values of SOC by the DLSOC-REM technique and the actual SOC values. The results demonstrated that the SOC values remain high at the earlier time step and tend to be low with a reduction in time step. It is also clear that the DLSOC-REM methodology has predictable SOC values which are more nearer to the actual SOC values.

Figure 9 illustrates the result analysis of the DLSOC-REM approach on the test FUDS drive cycle. The figure shows the estimated values of SOC by the DLSOC-REM algorithm and the actual SOC values. The outcomes depicted that the SOC values remain high at the previous time step and are inclined to decrease with a decrease in time step. It is also obvious that the DLSOC-REM algorithm estimated SOC values that are closer to the actual SOC values. By looking into the above-mentioned results and discussion, it is evident that the DLSOC-REM technique accomplished effective SOC estimation outcomes. Therefore, the DLSOC-REM technique can be applied as an efficient tool for SOC estimation in HEVs.

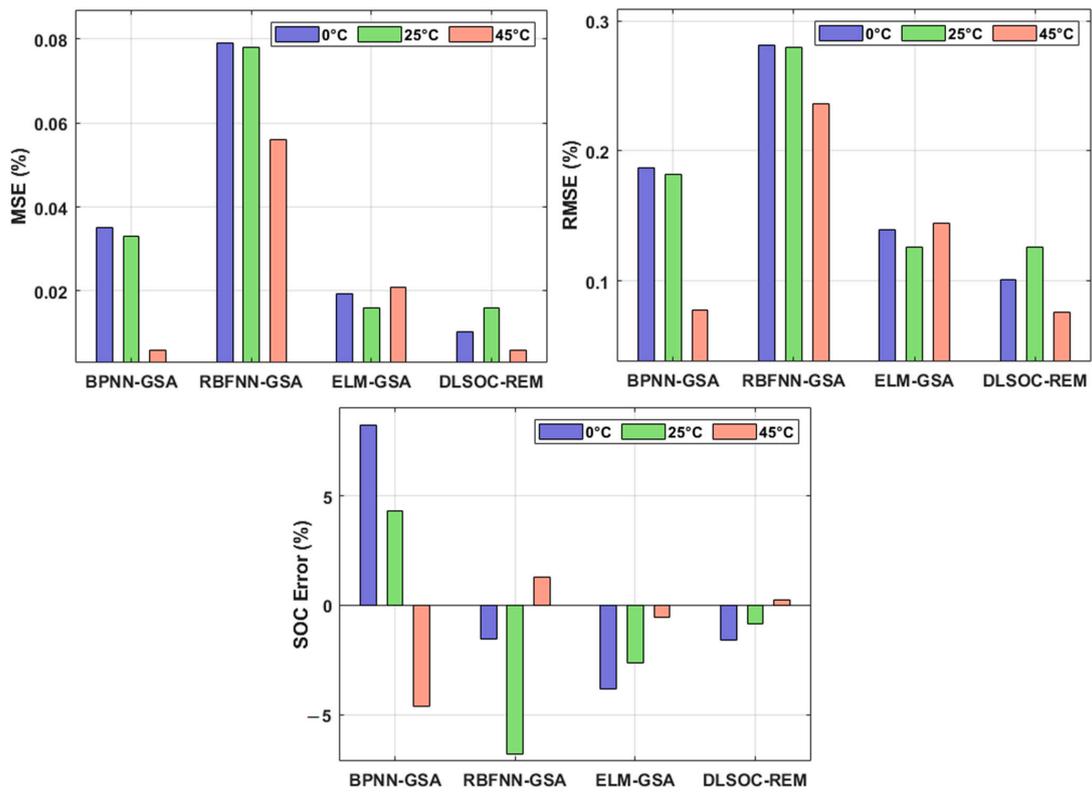


Figure 6. Comparative SOC Estimation analysis of DLSOC-REM technique under FUDS drive cycle.

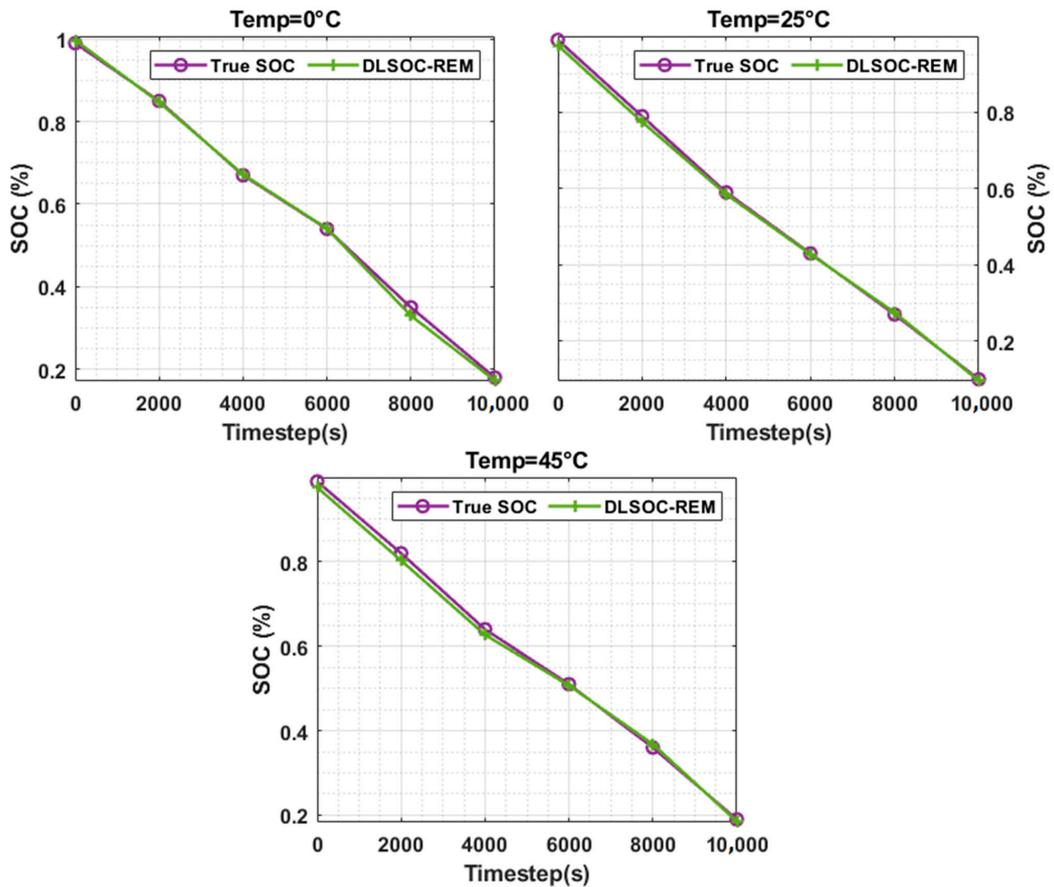


Figure 7. SOC analysis of DLSOC-REM technique on BJDST drive cycle.

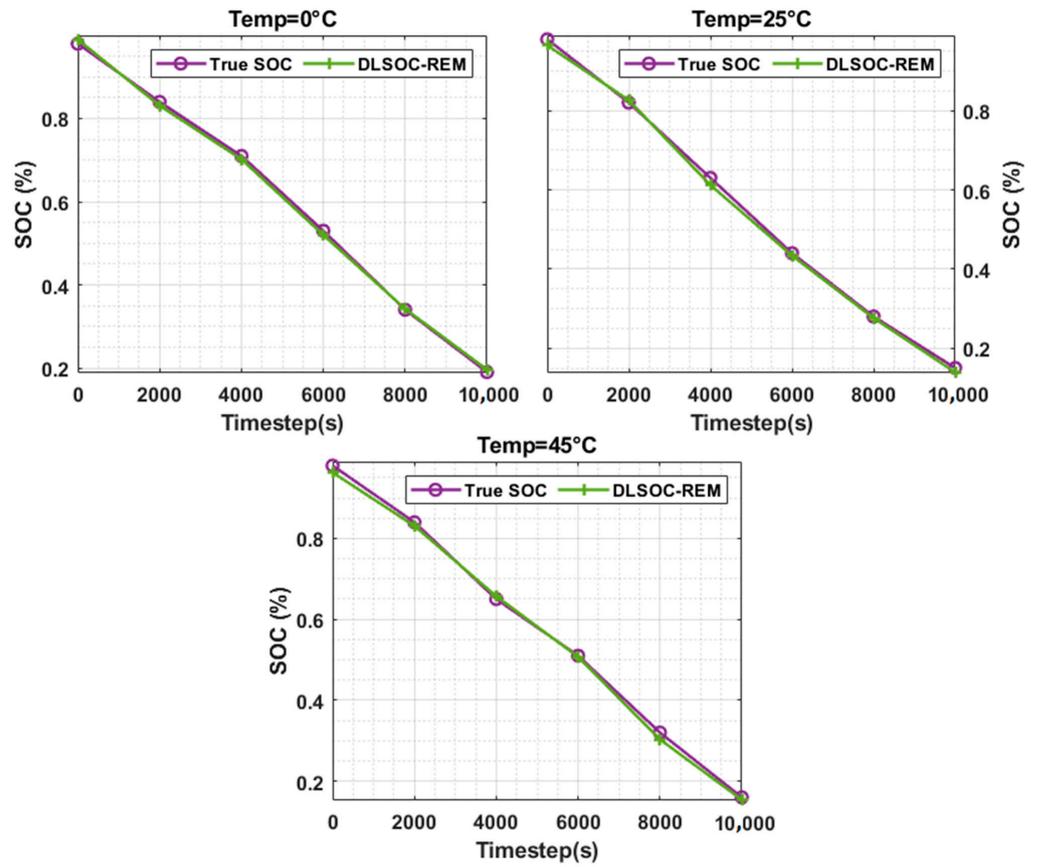


Figure 8. SOC analysis of DLSOC-REM technique on US06 drive cycle.

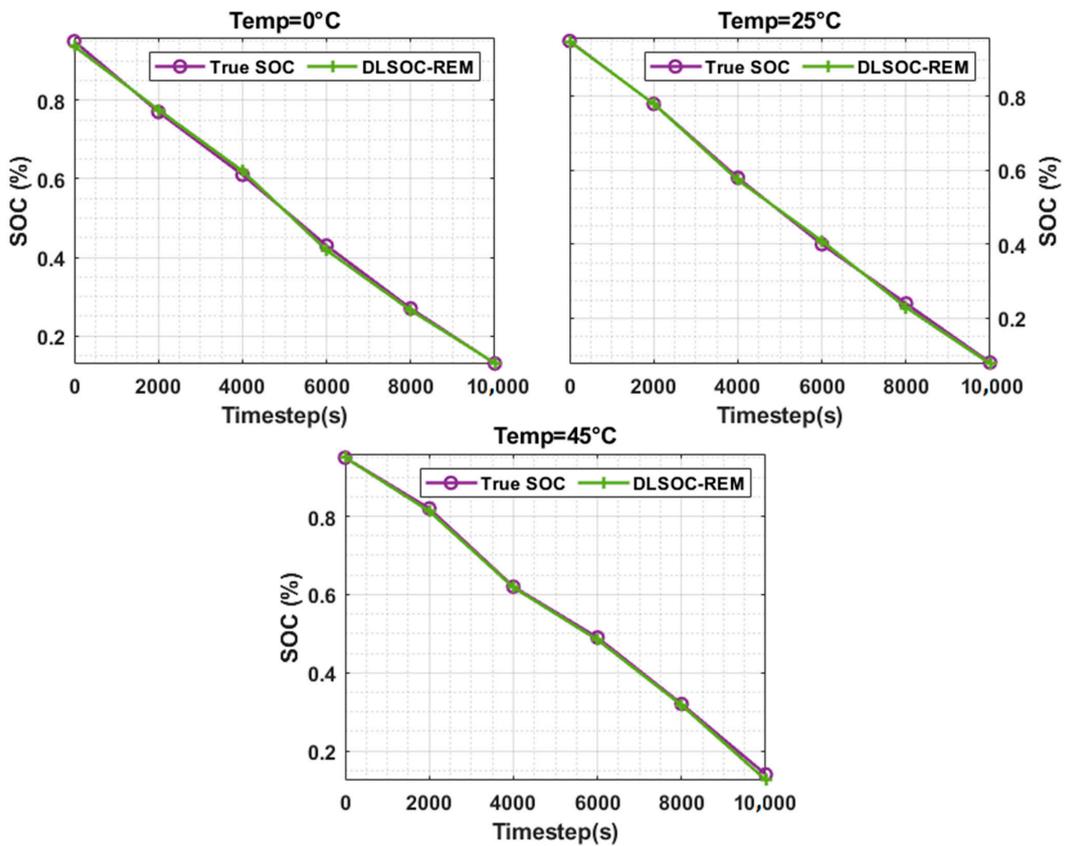


Figure 9. SOC analysis of DLSOC-REM technique on FUDS drive cycle.

5. Conclusions

This paper developed an effective DLSOC-REM technique for accurate SOC estimation in HEVs. The DLSOC-REM technique encompasses two major processes such as HCNN-LSTM-based prediction and BMO-based hyperparameter tuning. The utilization of the HCNN-LSTM model makes the modeling process easier and offers a precise depiction of the input–output relationship of the battery model. Furthermore, the optimal choice of hyperparameters reduces the error rate and improves the predictive outcomes. In order to demonstrate the enhanced performance of the DLSOC-REM technique, a series of simulations took place and the results were inspected under various aspects. An extensive comparative analysis reported the betterment of the DLSOC-REM technique over recent state-of-the-art approaches in terms of different aspects. Therefore, the DLSOC-REM technique can be employed as an effective tool for accurate and rapid SOC estimation in HEVs. In the future, hybrid metaheuristic optimization algorithms can be designed for improved SOC estimation outcomes and can be deployed in a real-time environment.

Author Contributions: Conceptualization, M.T.V.; data curation, M.T.V.; formal analysis, M.T.V. and I.M.M.; investigation, I.M.M.; methodology, T.P.; project administration, I.M.M.; software, T.P.; supervision, M.T.V.; validation T.P.; visualization, T.P.; writing—original draft, I.M.M.; writing—review and editing, M.T.V. All authors have read and agreed to the published version of the manuscript.

Funding: Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IFPIP: 536-135-1442) and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IFPIP: 536-135-1442) and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

References

1. Anselma, P.G. Computationally efficient evaluation of fuel and electrical energy economy of plug-in hybrid electric vehicles with smooth driving constraints. *Appl. Energy* **2022**, *307*, 118247. [[CrossRef](#)]
2. Xiong, R.; Ma, S.; Li, H.; Sun, F.; Li, J. Toward a safer battery management system: A critical review on diagnosis and prognosis of battery short circuit. *Iscience* **2020**, *23*, 101010. [[CrossRef](#)] [[PubMed](#)]
3. Hua, Y.; Wang, N.; Zhao, K. Simultaneous unknown input and state estimation for the linear system with a rank-deficient distribution matrix. *Math. Probl. Eng.* **2021**, *2021*, 1–11. [[CrossRef](#)]
4. Xiong, R.; Yang, R.; Chen, Z.; Shen, W.; Sun, F. Online fault diagnosis of external short circuit for lithium-ion battery pack. *IEEE Trans. Ind. Electron.* **2019**, *67*, 1081–1091. [[CrossRef](#)]
5. Zuo, H.; Zhang, B.; Huang, Z.; Wei, K.; Zhu, H.; Tan, J. Effect analysis on SOC values of the power lithium manganate battery during discharging process and its intelligent estimation. *Energy* **2022**, *238*, 121854. [[CrossRef](#)]
6. Xiong, R.; Wang, J.; Shen, W.; Tian, J.; Mu, H. Co-estimation of State of charge and capacity for Lithium-ion batteries with multi-stage model fusion method. *Engineering* **2021**, *7*, 1469–1482. [[CrossRef](#)]
7. Chen, X.; Shen, W.; Dai, M.; Cao, Z.; Jin, J.; Kapoor, A. Robust adaptive sliding-mode observer using RBF neural network for lithium-ion battery state of charge estimation in electric vehicles. *IEEE Trans. Veh. Technol.* **2015**, *65*, 1936–1947. [[CrossRef](#)]
8. 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.* **2017**, *65*, 6730–6739. [[CrossRef](#)]
9. Kim, W.Y.; Lee, P.Y.; Kim, J.; Kim, K.S. A robust state of charge estimation approach based on nonlinear battery cell model for lithium-ion batteries in electric vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5638–5647. [[CrossRef](#)]
10. Jiaqiang, E.; Zhang, B.; Zeng, Y.; Wen, M.; Wei, K.; Huang, Z.; Chen, J.; Zhu, H.; Deng, Y. Effects analysis on active equalization control of lithium-ion batteries based on intelligent estimation of the state-of-charge. *Energy* **2022**, *238*, 121822.
11. Zhang, M.; Wang, K.; Zhou, Y.T. Online state of charge estimation of lithium-ion cells using particle filter-based hybrid filtering approach. *Complexity* **2020**, *2020*, 1–10. [[CrossRef](#)]

12. Zahid, T.; Xu, K.; Li, W.; Li, C.; Li, H. State of charge estimation for electric vehicle power battery using advanced machine learning algorithm under diversified drive cycles. *Energy* **2018**, *162*, 871–882. [[CrossRef](#)]
13. Lai, X.; Wang, S.; He, L.; Zhou, L.; Zheng, Y. A hybrid state-of-charge estimation method based on credible increment for electric vehicle applications with large sensor and model errors. *J. Energy Storage* **2020**, *27*, 101106. [[CrossRef](#)]
14. Veerendra, A.S.; Mohamed, M.R.; Leung, P.K.; Shah, A.A. Hybrid power management for fuel cell/supercapacitor series hybrid electric vehicle. *Int. J. Green Energy* **2021**, *18*, 128–143. [[CrossRef](#)]
15. Chandran, V.; Patil, C.K.; Karthick, A.; Ganeshaperumal, D.; Rahim, R.; Ghosh, A. State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *World Electr. Veh. J.* **2021**, *12*, 38. [[CrossRef](#)]
16. How, D.N.; Hannan, M.A.; Lipu, M.S.H.; Sahari, K.S.; 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](#)]
17. 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](#)]
18. 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](#)]
19. Du, J.; Liu, Z.; Wang, Y. State of charge estimation for Li-ion battery based on model from extreme learning machine. *Control. Eng. Pract.* **2014**, *26*, 11–19. [[CrossRef](#)]
20. Kim, T.Y.; Cho, S.B. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **2019**, *182*, 72–81. [[CrossRef](#)]
21. Sulaiman, M.H.; Mustaffa, Z.; Saari, M.M.; Daniyal, H. Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103330. [[CrossRef](#)]
22. Jia, H.; Sun, K. Improved barnacles mating optimizer algorithm for feature selection and support vector machine optimization. *Pattern Anal. Appl.* **2021**, *24*, 1–26. [[CrossRef](#)] [[PubMed](#)]
23. Lipu, M.S.H.; Hannan, M.A.; Hussain, A.; Saad, M.H.; Ayob, A.; Uddin, M.N. Extreme learning machine model for state-of-charge estimation of lithium-ion battery using gravitational search algorithm. *IEEE Trans. Ind. Appl.* **2019**, *55*, 4225–4234. [[CrossRef](#)]