Energy 282 (2023) 128317

Contents lists available at ScienceDirect

Energy

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

Hybrid deep learning model for efficient state of charge estimation of Li-ion batteries in electric vehicles

Muhammad Hamza Zafar [a,1], Majad Mansoor [b,1], Mohamad Abou Houran [c], Noman Mujeeb Khan [d], Kamran Khan [e], Syed Kumayl Raza Moosavi [f], Filippo Sanfilippo [a,g,*]

[a] Department of Engineering Sciences, University of Agder, NO-4879, Grimstad, Norway
[b] Department of Automation, University of Science and Technology of China, Hefei, China
[c] School of Electrical Engineering, Xi'an Jiaotong University Shaanxi, 710049, China
[d] Department of Electrical Engineering, Beaconhouse College, Islamabad, Pakistan
[e] Department of Electrical Engineering, Hamdard University, Islamabad Campus, Islamabad, Pakistan
[f] SEECS, National University of Sciences and Technology, Islamabad, Pakistan
[g] Department of Software Engineering, Kaunas University of Technology, LT-44029 Kaunas, Lithuania

ARTICLE INFO

Handling Editor: X Ou

Keywords:
State of charge (SoC)
Deep neural network (DNN)
Mountain gazelle optimizer (MGO)
Relative error (RE)
Statistical analysis (SA)

ABSTRACT

State of charge (SoC) estimation is critical for the safe and efficient operation of electric vehicles (EVs). This work proposes a hybrid multi-layer deep neural network (HMDNN)-based approach for SoC estimation in EVs. This HMDNN uses Mountain Gazelle Optimizer (MGO) as a training algorithm for the deep neural network. Our method leverages the intrinsic relationship between the SoC and the voltage/current measurements of the EV battery to accurately estimate the SoC in real time. We evaluate our approach on a large dataset of real-world EV charging data and demonstrate its effectiveness in comparison to traditional SoC estimation methods. Four diverse Li-ion battery datasets of electric vehicles are employed which are the dynamic stress test (DST), Beijing dynamic stress test (BJDST), federal urban driving schedule (FUDS), and highway driving schedule (US06) with different temperatures of 0^oC, 25^oC, 45^oC. The comparison is made with Mayfly Optimization Algorithm based DNN, Particle Swarm Optimization based DNN and Back-Propagation based DNN. The evaluation indices used are normalized mean square error (NMSE), root mean square error (RMSE), mean absolute error (MAE), and relative error (RE). The proposed algorithm achieves 0.1% NMSE and 0.3% RMSE on average on all datasets, which validates the effective performance of the proposed model. The results show that the proposed neural network-based approach can achieve higher accuracy and faster convergence than existing methods. This can enable more efficient EV operation and improved battery life.

1. Introduction

The state of charge (SoC) of a battery refers to the amount of electrical energy stored in the battery at a given time. Accurately estimating the SoC is important for a variety of applications, including battery management, electric vehicle range prediction, and renewable energy systems scheduling [1]. To effectively manage and control a battery, it is important to have an accurate estimate of its SoC [2]. This information can optimize charging and discharging operations, prevent overcharging or over-discharging, and extend the overall battery life.

In the context of electric vehicles (EV), accurate SoC estimation is crucial for predicting the vehicle's range and ensuring that the driver has sufficient energy to reach their destination improving the EV driving experience. SoC adds optimum battery management, enhanced energy efficiency, and battery monitoring in real-time considering the operating condition in the long and short term. This information can also be used by the vehicle's control system to optimize energy usage and extend the vehicle's range [3]. In renewable energy systems, such as solar or wind power systems, accurate SoC estimation is important for optimizing the use of stored energy and ensuring sufficient energy is available to meet demand [4]. Overall, the ability to accurately estimate the SoC of a battery is important for a wide range of applications and continues to be

* Corresponding author. Department of Engineering Sciences, University of Agder, NO-4879, Grimstad, Norway.
E-mail addresses: hamzauetms@gmail.com (M.H. Zafar), majad@mail.ustc.edu.cn (M. Mansoor), houran@xjtu.edu.cn (M. Abou Houran), noman.mujeeb@bic.edu.pk (N.M. Khan), Kamran.khan@hamdard.edu.pk (K. Khan), smoosavi.msai21seecs@seecs.edu.pk (S.K. Raza Moosavi), filippo.sanfilippo@uia.no (F. Sanfilippo).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.energy.2023.128317
Received 17 February 2023; Received in revised form 20 May 2023; Accepted 2 July 2023
Available online 11 July 2023

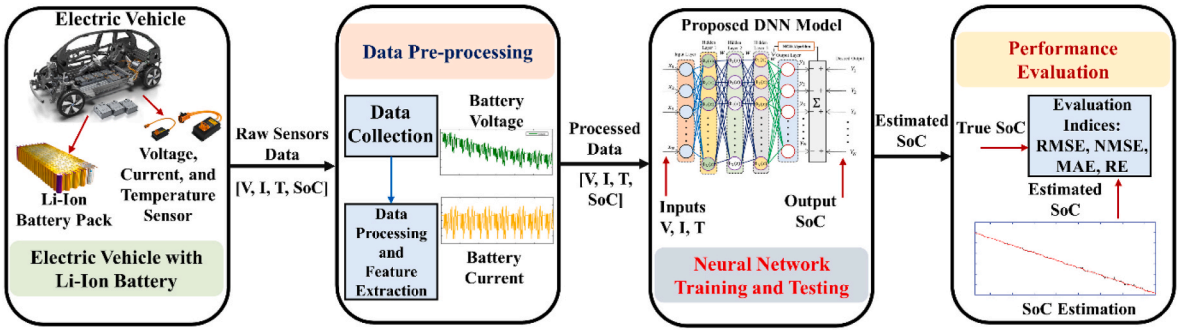

Fig. 1. Proposed SoC Estimation Process for Li-ion batteries in EV.

an active area of research in the fields of electrical engineering and battery technology. The use of EVs is growing in popularity due to their potential for reducing greenhouse gas emissions and dependence on fossil fuels [5]. Accurate estimation SoC of the battery is essential for the proper operation of an EV, as it indicates the amount of energy remaining in the battery and can affect the performance and range of the vehicle.

Traditionally, SoC estimation has been performed using mathematical models based on the electrochemical characteristics of the battery. However, these models can be complex and may not accurately reflect the real-world behavior of the battery, especially in cases where the battery has aged or experienced other factors that can affect its performance.

1.1. Literature review

State of charge consequently reveals how much energy is still in the battery. It must, however, be implicitly calculated from the apparent battery characteristics and factors since it cannot be specifically defined. Due to its uses in EV range prediction, SoC estimation is thus of utmost relevance. The literature has throughout the years presented a broad variety of battery SoC estimation techniques, which may be roughly classified into four categories: lookup table approach, ampere-hour integral method, model-based estimation method, and model-free parameter estimation.

The lookup table techniques have well-known drawbacks and have gradually given way to more advanced techniques like the Model-Based estimate method. By combining them with nonlinear state estimating algorithms, several studies have also sought to increase the estimation capability of Lookup Table (LUT), Coulomb-Counting [6], and Model based techniques [7]. The Kalman filter [8], Luenberger observer [9], proportional integral viewer [10], and sliding mode viewer [11] are examples of common algorithms cited in the literature.

The traditional Kalman filter is sensitive to the intricate non-linear process, temperature, and battery charging/discharging since it is only appropriate for linear systems. Extended Kalman filter (EKF), the unscented Kalman filter (UKF) have been effectively used to estimate SoC to overcome this challenge [12–14]. Fuzzy logic is a popular artificial intelligence approach that uses multi-valued logic to compute SoC while also taking into consideration a variety of other factors, like age, temperature, noise, and many more [15]. Fuzzy logic technique requires a huge amount of training data as well as long-term experience in data collection to develop a reliable rule. Support vector machine (SVM) [16], Gaussian process regression (GPR) [17] perform adequately, especially in nonlinear battery modeling. Multi-input parameters can be well handled by SVM and GPR extensions have been modified and improvised in recent research works. Additionally, recommended for clustering and battery state prediction is the Greenwald-Khanna method [18], where the entropy weight method is employed to determine the weight using Kernal functions and genetic evolution. These approaches provide more accurate capacity, battery cells package clustering, and energy estimates. Overly complex computing is a significant issue that prevents the procedure from being carried out in the battery management system (BMS).

A simple technique for calculating SoC is Coulomb counting [19], which accumulates the net charge at the most recent time in ampere-hours (Ah). Its effectiveness is heavily dependent on the accuracy of the current sensors and the accuracy of the initial SoC calculation. Coulomb counting is an open-loop estimator, though, thus it does not completely prevent the buildup of measurement errors and ambiguous perturbations. It also cannot account for variations in the first SoC brought on by self-discharging or identify the initial SoC [20]. This strategy will result in growing SoC estimation mistakes without knowledge of the starting SoC [21].

The data-driven estimating approach, on the other hand, has also been used in several studies to estimate the SoC. Support vector machines, fuzzy control, artificial neural networks (ANN) [22], and other combinations of the methods were often used in classical machine learning applications. Conventional machine learning methods typically involve no more than two layers of processing [23].

ANN is receiving increasing attention for state estimations under various battery dynamics, fluctuating load, and changing temperatures. The main advantage is that information fusion-based models properly represent the nonlinear behavior of the battery discharging and charging process because of the training process. There are difficulties in selecting the hidden nodes' activation function, figuring out how many neurons are in the hidden layer, and adjusting the learning rate. Traditional neural networks often involve complex computations, and when the training data is scarce, the trained model is not accurate for a diverse range of operations. Conversely, when the training data is huge, the model is easily caught in a local minimum condition generating rigid solutions [24].

Nevertheless, several advancements in recent years have made the traditional ANN very effective. The capability of the conventional ANN is considerably improved by adding more computational layers than were previously conceivable owing to software and hardware limitations. The deep-neural-network (DNN) [25], which is simply an ANN with extra computational layers, is the enhanced form of the ANN. The DNN now holds the best performance in areas including computer vision, speech recognition, and natural language processing when combined with smarter training methods and adjustments. Despite recent advancement, an accurate estimation of SoC is essential to accommodate the variety of conditions and electrochemical properties of materials used by classical empirical techniques. Robust algorithms are core for collecting data to be analyzed and improvise SoC under diverse operating scenarios [26]. Most existing models do not accommodate for aging health factors and cross-correlations of SoC with interdependent nonlinearities within collected data [27,28]. Modern SoC estimation is essential to ensure that EVs are equipped with accurate and reliable systems for monitoring and managing their batteries [29].

To estimate the SoC using DNN is a relatively recent concept, nonetheless. There aren't many published papers on SoC estimation

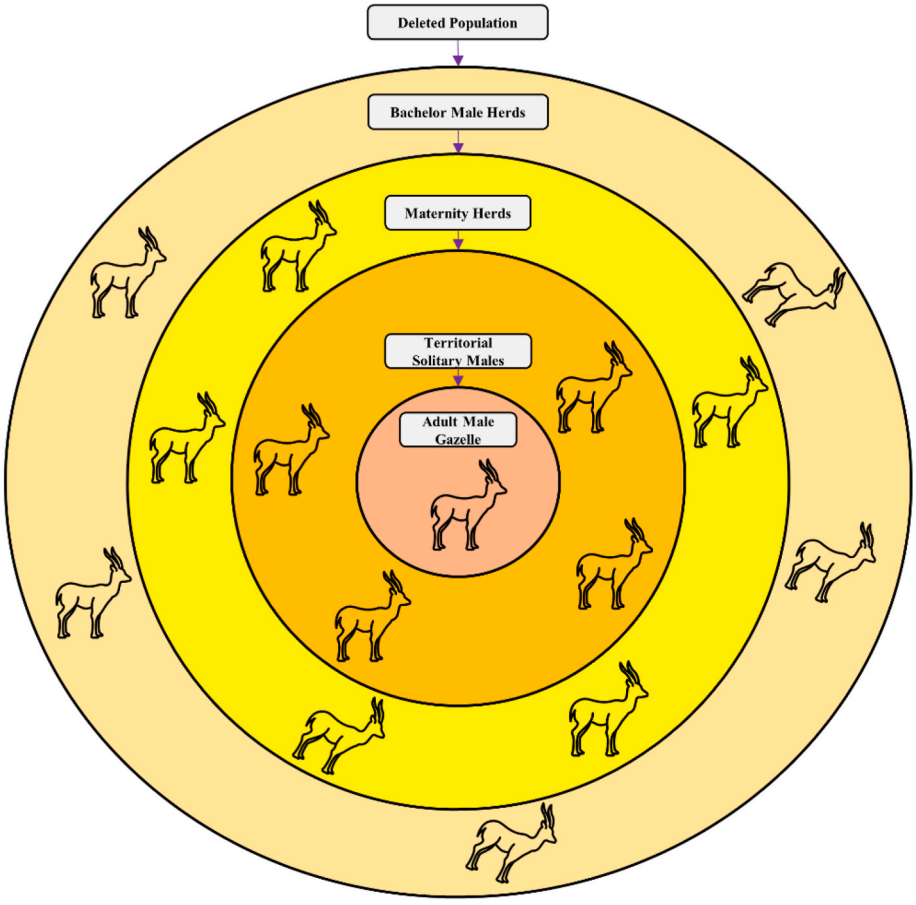

Fig. 2. Optimization procedure of MGO algorithm.

with DNN. In Ref. [30], the SoC of batteries has been correctly predicted by the authors using two distinct DNN variations. LSTM network was trained and tested on certain driving cycles that were adopted. Without the use of any filtering techniques, the data reveal very positive results. These authors suggested utilizing multilayer DNN to forecast the SoC in another paper. However, there are several DNN variations that haven't been studied for the purpose of SoC estimation [31]. Numerous deep learning publications have emerged that offer various types of deep neural networks for SoC estimation [32]. This study is inspired by these most recent developments and trends.

1.2. Contribution and paper organization

In this paper, Hybrid Deep Neural Network structure is proposed for SoC estimation for Lithium-ion batteries in EV. The proposed generalized structure of SoC estimation is presented in Fig. 1. Contributions of this work are.

- Detailed Analysis of multilayer DNN structure for SoC estimation is presented.
- Novel implementation of Mountain Gazelle Optimizer (MGO) based Deep Neural Network architecture is proposed for SoC Estimation.
- MGO updates parameters i.e., weights and biases, of DNN according to the loss function.
- Comparison is made with Mayfly Optimization based DNN (MFADNN), Particle Swarm Optimization based DNN (PSODNN), and Back-Propagation based DNN (BPDNN).
- Four distinct Drive Cycle Tests at different temperatures are employed for Validation.
- Evaluation indexes i.e., Normalized Mean Square Error (NMSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Relative Error (RE) are presented for comparison.
- The proposed MGODNN shows better performance as compared to competing techniques.

The rest of the paper is organized as follows: Section 2 includes the proposed technique in which the Mountain Gazelle Optimizer is explained. The DNN architecture and the training process is also outlined. Section 3 explains the dataset generation of different drive cycles. Section 4 presents the estimation results of the proposed technique and a comparative analysis with competing techniques. Summary of results and conclusions are given in Section 5.

2. Proposed technique

In this section, first, the Mountain Gazelle Optimizer algorithm is presented highlighting the use of this approach for optimization problems. Successively, the architecture of the proposed DNN is explained. An important part of DNN is the hyperparameters, which are explained sub-sequentially. In the last part, DNN training using MGO is outlined.

2.1. Mountain Gazelle Optimizer (MGO)

This section presents an optimization method based on mountain gazelle social behavior and lifestyle [33]. A mathematical representation of the MGO algorithm has been developed using the fundamental ideas underlying the community and grouping behavior of mountain gazelle. The four primary aspects of the mountain gazelle's life—bachelor male herds, maternity herds, solitary, territorial males, and

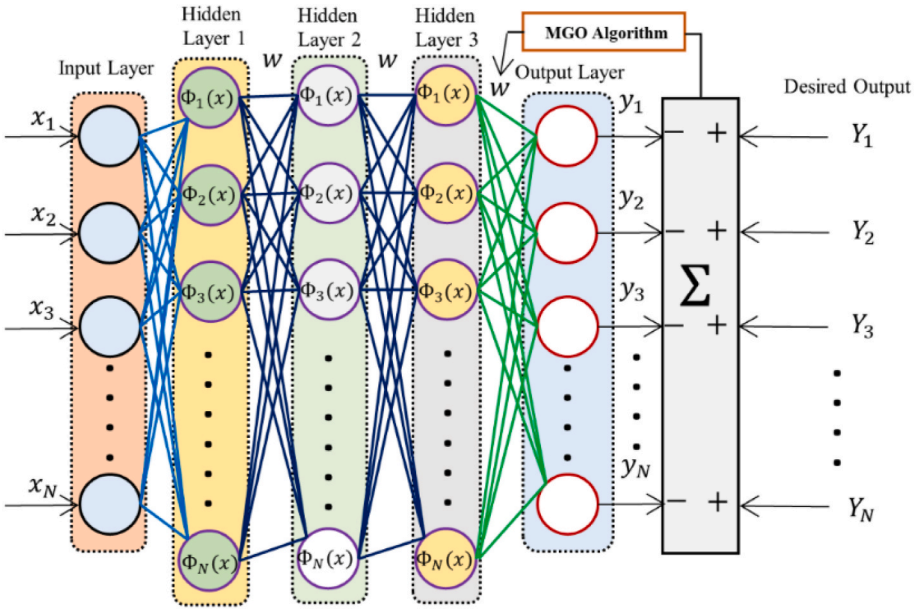

Fig. 3. Proposed 3 Layer DNN structure with MGO as Training Algorithm to update weights and biases.

Table 1
Different hyperparameters of DNN and selected values.

Hyperparameters	Selected Values
No. of Hidden Neurons	10
No. of Hidden Layers	3
Optimization Algorithm	Mountain Gazelle Optimizer (MGO)
Dropouts	No
Activation Function for Hidden Layers	Radial Basis
Activation Function for Output Layer	Sigmoid
No. of Weights and Biases	151

movement in search of food—are used to execute optimization operations.

Each gazelle (Xi) in the MGO algorithm has the potential to join a herd of maternity herds, a herd of solitary males, or a herd of lone, territorial males during the optimization procedure. One of these 3 herds may give birth to a young gazelle. Adult male gazelles in the herd's domain are MGO's greatest option for the entire world. Approximately one-third of the search population, which consists of male bachelor gazelles that are not yet mature or strong enough to reproduce or dominate females, will have the lowest fitness. This is because these gazelles are not yet fully developed or capable of reproducing or gaining control of female gazelles.

Other options that are open to the entire populace are also regarded as gazelles in maternity herds. At the conclusion of each repeat, strong gazelles with viable alternatives are still there. Old and diseased gazelles are deemed to be eliminated from the population, whereas other alternatives that are introduced to the overall population and cost considerably less are thought to be. The techniques used by MGO to carry out optimization operations are developed and described mathematically in the sections that follow.

As compared to other metaheuristic algorithms, the exploitation and exploration stages of the proposed method are carried out simultaneously employing four mechanisms. The exploitation and exploration stages are carried out concurrently. In other words, a solution may advance in the direction of the optimum solution while also carrying out the exploration operation in accordance with the four processes of the suggested model. These methods are presented in Fig. 2.

i. Territorial Solitary Males

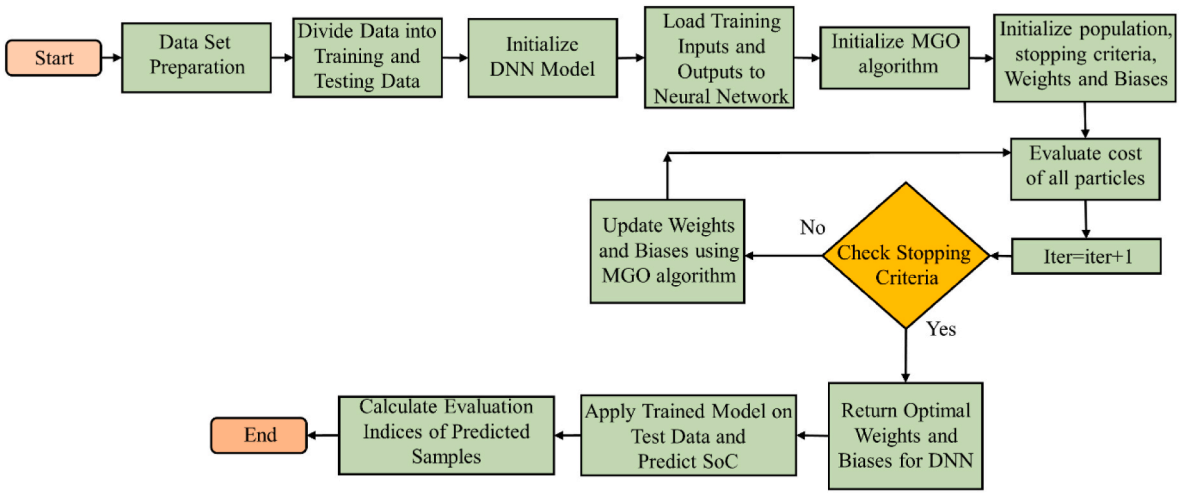

Fig. 4. Flow Chart of Proposed Technique in which MGO uses to update Weights and Biases of DNN.

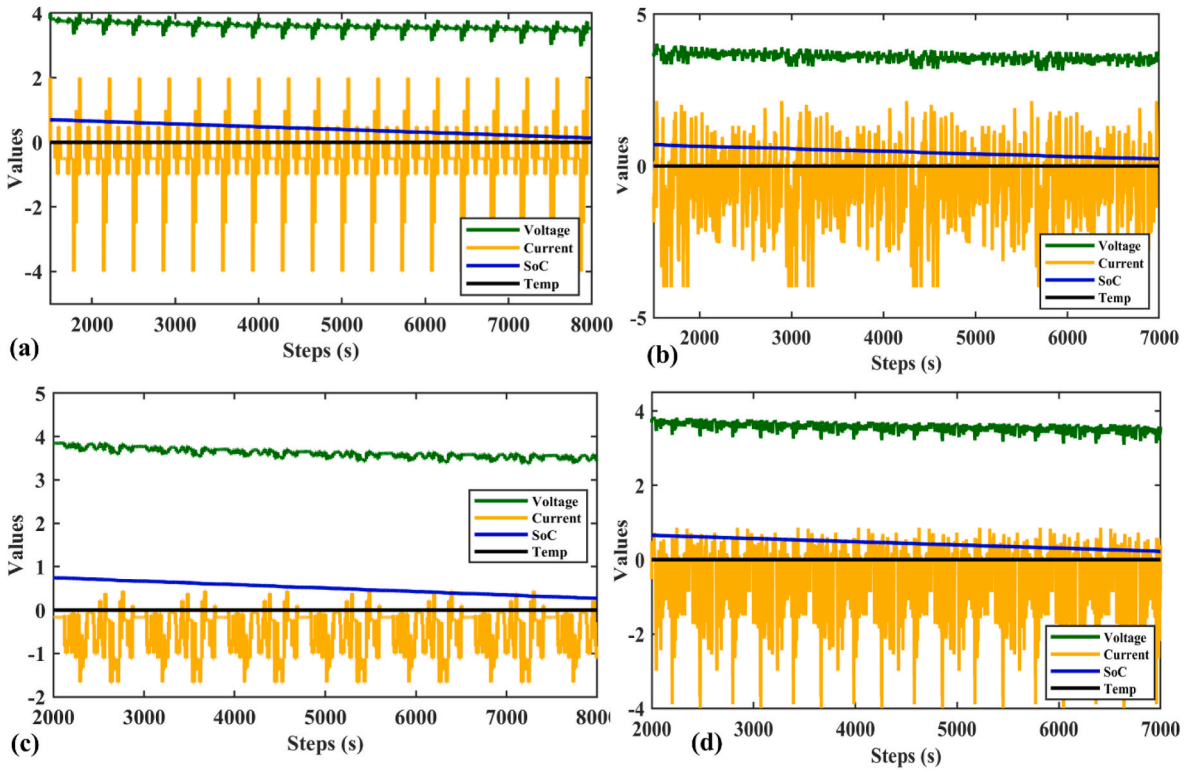

Fig. 5. Voltage, Current, Temperature and SoC of Battery for: (a) DST Dataset with 0 Degree Temp (b) FUDS Dataset with 0 Degree Temp (c) BJDST Dataset with 0 Degree Temp (d) US06 Dataset with 0 Degree Temp.

Table 2
Battery Specs used in Experimental Setup for Generation of Different Datasets.

Type	Capacity (Ah)	Voltage (V)	Cut off Voltage.	Maximum Current (A)	Life Cycle
18,650 NMC	2.0	3.60	2.4/4.2	22	1000–2000

When adult alpine gazelles enter maturity and are physically able to defend themselves, they establish solitary regions, which are far apart. Adult male gazelles fight each other for control of the female's territory or ownership. While the older males attempt to safeguard their surroundings, the adolescent males attempt to seize the territory or the female. This is expressed by the following equation:

$$TSM = male_{gazelle} - |(ri_1 \times BH - ri_2 \times X\backslash(t)) \times F| \times Cof_r \quad (1)$$

where $male_{gazelle}$ is the global best solution in equation (1). Random integers between 0 and 1make up the parameters ri_1 and ri_2. The young male herd coefficient vector, or BH, is determined using Equation (2). Equation (3) is used to calculate F. Additionally, to improve search efficiency, Cof_r is a randomly chosen coefficient vector that is updated after each iteration (4). The value of Cof_r is calculated using Equation (4).

$$BH = X_{ra} \times [ri_1] + M_{pr} \times [ri_2], ra = \left\{ \left[\frac{N}{3}\right] \cdots N \right\} \quad (2)$$

In Equation (2), X_{ra} represents random particle in the range of ra. M_{pr} is average number of particles selected randomly $\left[\frac{N}{3}\right]$ which are randomly selected. N is total population, while ri_1 and ri_2 are random values between 0 and 1.

$$F = N_1(D) \times exp\left(2 - iter \times \left(\frac{2}{Maxiter}\right)\right) \quad (3)$$

where N_1 is random value from normal distribution in Equation (3)'s problem dimensions. *Iter* is the current iteration, *Maxiter* is the total iterations, and *exp* is another name for the exponential function.

$$Cof_r = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times cos((r_4 \times 2) \times N_4(D)), \end{cases} \quad (4)$$

Equation (4) determines the value of "a" using Equation (5). Additionally, r_3," r_4," are random numbers that fall within the range of 0 and 1. " N_2," " N_3," and " N_4" are random numbers that are normally distributed within the dimensions of the problem. The symbol " *cos* " represents the cosine function. The parameter a is a controlling parameter which varies from −1 to −2 over the iterations and it is obtained as follows:

$$a = -1 + iter \times \left(\frac{-1}{Maxiter}\right) \quad (5)$$

ii. Maternity Herds

This stage is crucial to the mountain gazelle's life cycle. Male gazelle can also be involved in the birth of gazelles and the attempts of young males to mate with females. Equation (6) is used to analyze this behavior:

$$MH = \left(BH + Cof_{1,r}\right) + \left(ri_3 \times male_{gazelle} - ri_4 \times X_{rand}\right) \times Cof_{2,r} \quad (6)$$

In Equation (6), "BH" is young male's influence value, as calculated in Equation (2). $Cof_{1,r}$ and $Cof_{2,r}$ are factors computed using Equation (4). ri_3 and ri_4 represents integer value either 1 or 2. $male_{gazelle}$ represents the best solution. Finally, X_{rand} is randomly selected particle.

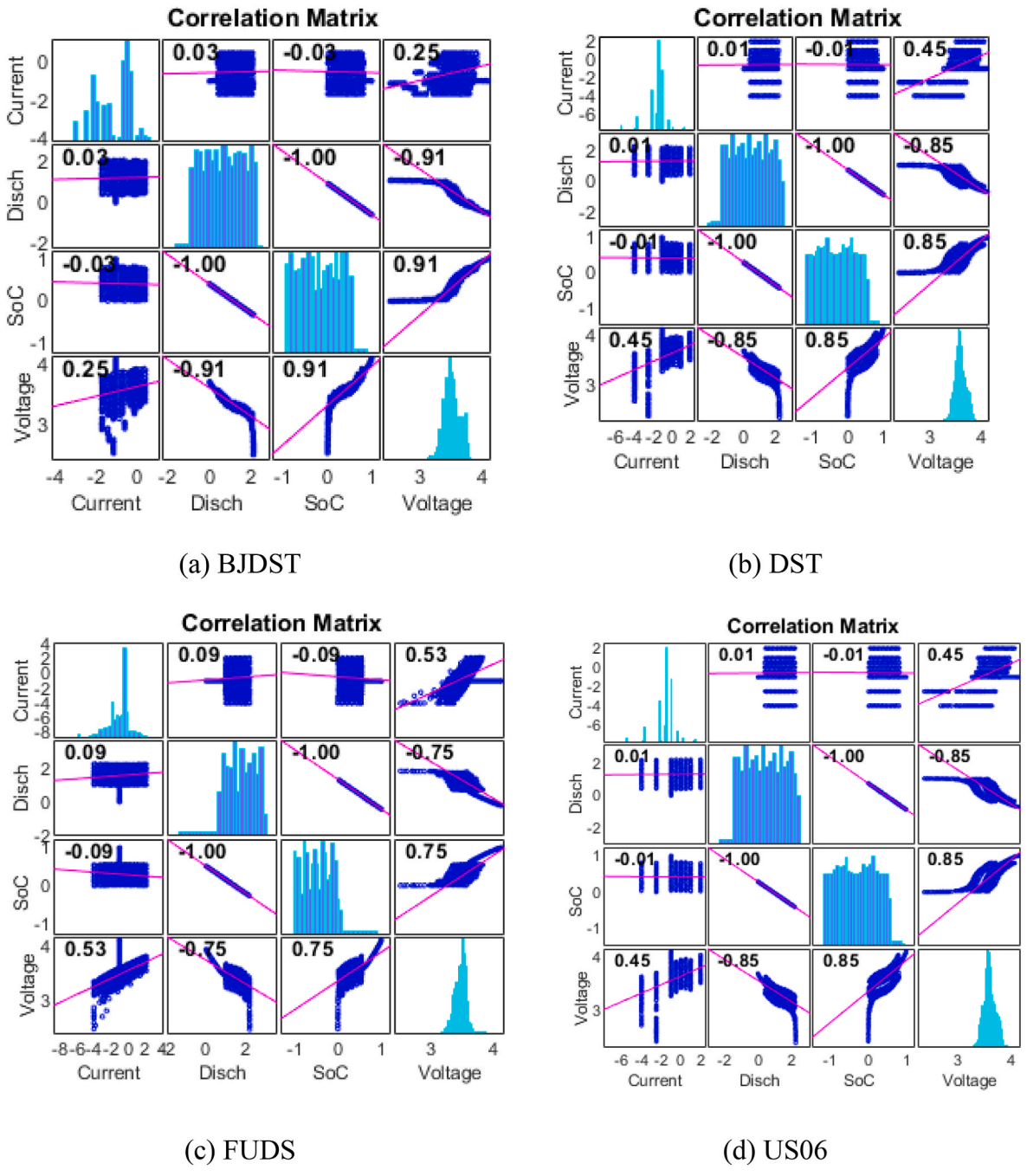

Fig. 6. (a–d) The data correlation Matrix at STC conditions for 4-main data clusters for SoC w.r.t. Voltage, current and Discharge.

Table 3
Evaluation of proposed training algorithm on different hidden layers of DNN (DST drive cycle).

Hidden Layers	NMSE	RMSE	MAE	RE	Time (s)
1	0.0980	0.023	4.27	0.215	**101.1**
2	0.0013	0.009	0.57	0.047	124.5
3	**0.0010**	**0.003**	**0.15**	**0.017**	152.3
4	0.0016	0.007	0.68	0.024	178.6
5	0.0053	0.042	1.15	0.026	213.3

iii. Bachelor Male Herd

Male gazelles tend to establish territories and seize control of female gazelles as they get older. Young male gazelles start fighting older males for dominance of the females at this point, and there may be a lot of violence involved. Equation (7) is utilized to mathematically represent this behavior of gazelles:

$$BMH = (X(t) - D) + \left(ri_5 \times male_{gazelle} - ri_6 \times BH\right) \times Cof_r \quad (7)$$

In Equation (7), $X(t)$ represents current particle. The value of D is determined using Equation (8). ri_5 and ri_6 represents integer value either 1 or 2. Cof_r is selected randomly used using Equation (4).

$$D = \left(|X(t)| + |male_{gazelle}|\right) \times (2 \times r_6 - 1) \quad (8)$$

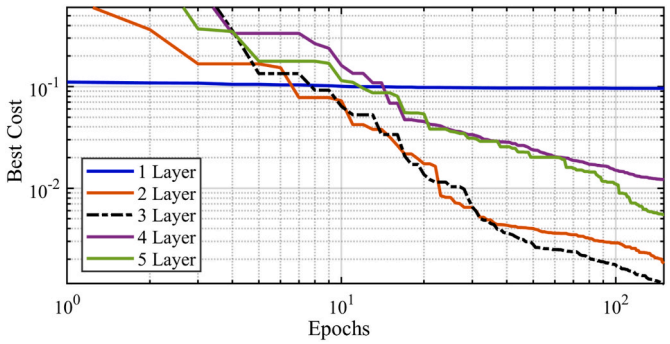

Fig. 7. Comparison of Best Cost vs Epochs for Multilayer DNN Structures on DST Drive Cycle Dataset using MGO as Training Algorithm.

iv. Migration (Movement) in Search of Food

These animals are known for their extreme running pace and strong jumping abilities. Equation (9) has been used to mathematically model these behaviors of mountain gazelles:

$$MSF = (ub - lb) \times r_7 + lb \tag{9}$$

In Equation (9), *ub* and *lb* are upper and lower limits. Finally, r_7 is a number between 0 and 1 chosen randomly. To produce new populations of gazelles, the four above explained mechanisms are applied to all gazelles. Each generation is equivalent to one replication, and each period increases the total population. Additionally, towards the conclusion of each era, all gazelles are sorted in increasing order. The best gazelles are those that offer high-caliber, promising solutions in the population. Older or weaker gazelles are culled from the population. The adult male gazelle who controls the territory is also thought to be the best gazelle.

2.2. DNN architecture

The DNN is a type of neural network that consists of at least four layers: input layer, two hidden layer and one output layer [34]. In this study, we investigate the use of DNNs for estimating SoC of battery. Rather than relying on expert knowledge of battery chemistry, we use a data-driven approach in which the DNN is trained on a dataset of battery parameters measured under controlled laboratory conditions. This dataset includes measurements of voltage, current, and temperature, which are used to estimate the SoC. While this method requires precise sensor calibration and accuracy, it offers an alternative to model-based approaches that rely on mathematical models of battery behavior.

In this case, DNN is trained to model the behavior of battery by mapping inputs, which are a vector of battery parameters (voltage, current, and temperature), to a scalar output representing the calculated SoC. Mathematically, the input vector is defined as $X = [V_k, I_k, T_k]$ and the output as $Y = SoC_k$, where V_k is the instantaneous voltage, I_k is the instantaneous current, T_k is the instantaneous temperature, and SoC_k is the SoC value. The input vector is provided to the input layer of DNN, and the resulting value of the SoC is obtained at the output layer. The proposed three-layer DNN structure is shown in Fig. 3.

To get the output Y_K i.e., SoC from the DNN, input must go through multiplication of matrix in hidden layers. h^k is the activation in hidden layers, w^k are the weights, b^k are the biases. Activation in hidden layer can be calculated as Equation (10):

$$h^k = \Omega\left(\sum\left(w^k h^{k-1} + b^k\right)\right) \tag{10}$$

where Ω is the activation function i.e., Radial Basis Function (RBF) and this RBF Is mathematically represented in Equation (11):

Table 4
Comparative analysis of all competing techniques with proposed technique.

Drive Cycles	Technique	NMSE	RMSE	MAE	Mean RE	Time (s)
DST	MGO-DNN	**0.0010**	**0.0021**	**0.15**	**0.017**	197.3
	MFA-DNN	0.0063	0.005	0.38	0.036	758.4
	PSO-DNN	0.0241	0.011	1.45	0.561	247.1
	BP-DNN	0.0157	0.078	0.79	0.651	**181.2**
BJDST	MGO-DNN	**0.0021**	**0.0024**	**0.24**	**0.052**	220.6
	MFA-DNN	0.0056	0.047	0.53	0.075	876.9
	PSO-DNN	0.0320	0.112	1.03	0.89	261.3
	BP-DNN	0.0345	0.120	1.87	1.67	**198.4**
FUDS	MGO-DNN	**0.0002**	**0.0004**	**0.023**	**0.0021**	138.7
	MFA-DNN	0.0031	0.0035	0.101	0.0065	387.2
	PSO-DNN	0.0254	0.072	1.22	0.0765	256.3
	BP-DNN	0.0165	0.023	0.91	0.019	**118.2**
US06	MGO-DNN	**0.0009**	**0.0019**	**0.13**	**0.006**	132.6
	MFA-DNN	0.0066	0.0530	1.43	0.023	654.9
	PSO-DNN	0.0148	0.811	2.37	0.061	189.3
	BP-DNN	0.0660	0.967	1.87	0.073	**101.3**

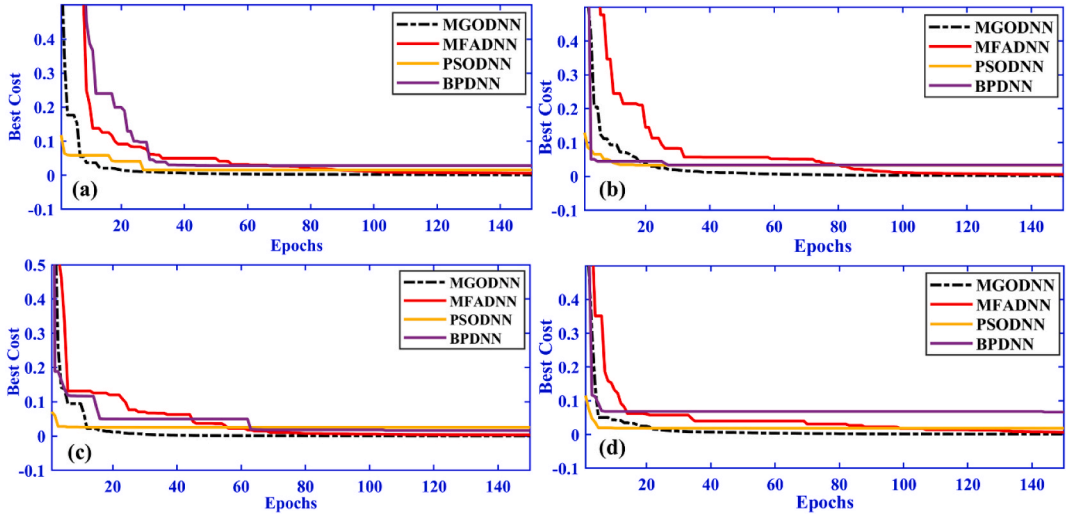

Fig. 8. Cost vs Epochs Comparative Analysis of all Techniques on: (a) DST Drive Cycle (b) BJDST Drive Cycle (c) FUDS Drive Cycle (d) US06 Drive Cycle.

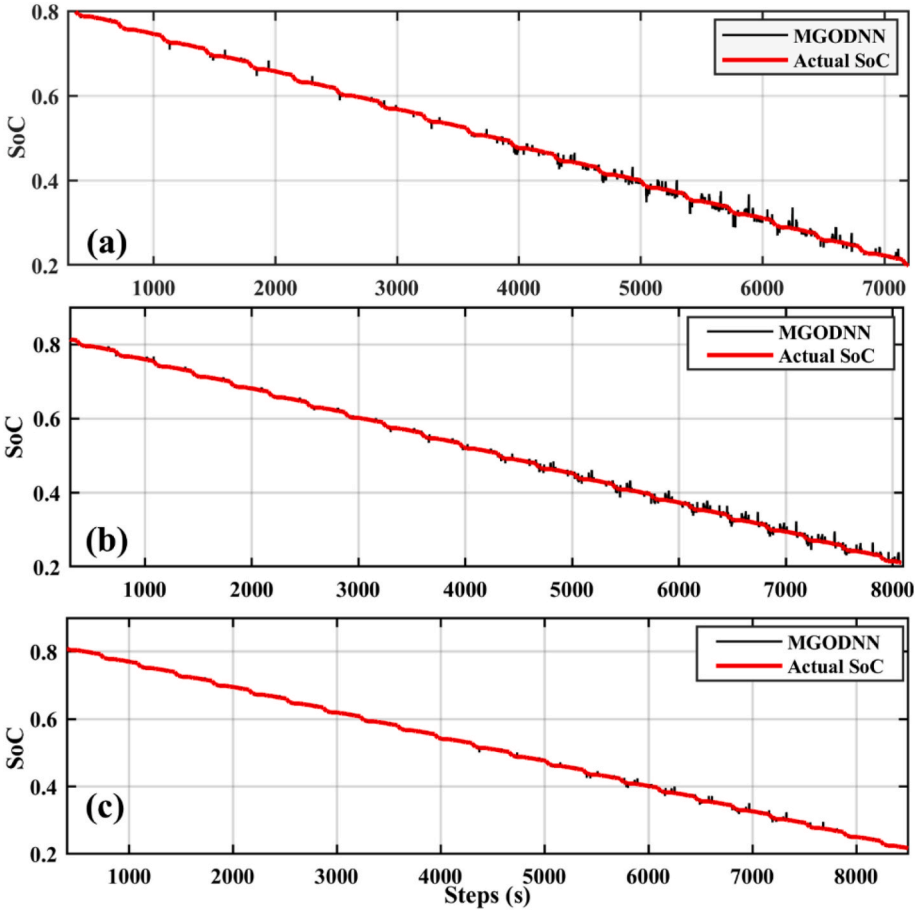

Fig. 9. SoC Estimation of MGODNN for DST Drive Cycle. (a) Drive cycle at 0° Temp (b) Drive cycle at 25° Temp (c) Drive cycle at 45° Temp.

$$\Omega(x) = e^{-\frac{(x-c)^2}{r^2}} \tag{11}$$

where x is the input, c is center, and r is radius of the represented dataset.

The output vector for the output layer neurons is computed as follows:

$$Y_K = SoC_K = \Omega\left(\sum\left(w^K h^{K-1} + b^K\right)\right) \tag{12}$$

where K represents the last hidden layer.

2.3. Hyperparameters of DNN

Deep learning algorithms often have a variety of hyperparameters that can affect their behavior and performance [35]. When using a DNN, it is necessary to choose the values of these hyperparameters. Table 1 lists some of the most important hyperparameters and their selected values for this work. As shown in the table, the number of possible combinations of these values is very large. Search space for hyperparameters rises exponentially with the increase in number of hidden layers and neurons. This makes it difficult and time-consuming to find the optimal combination of hyperparameters for a given task. There are various algorithms that can assist with hyperparameter optimization, but this is an active area of research and is outside the scope of this study [36]. In this study, we will focus on two important hyperparameters that have a significant impact on the behavior of a deep neural network: the number of hidden layers and the weights and biases of the layers.

It is well-known that increasing the number of neurons and hidden layers in a deep neural network model can improve its expressiveness, if overfitting is avoided. This study will examine the effects of increasing the number of hidden layers on the performance of the DNN. However, one major concern with adding more hidden layers and neurons is the risk of overfitting. To limit the danger of overfitting, a batch normalization (BN) layer is introduced after each consecutive hidden layer. By minimizing internal covariance shift, using BN layers has also been proven to drastically cut down on training time. In the following sections, a comparative analysis will be presented, which demonstrates that a model with 3 hidden layers is the best balance between accuracy and time.

2.4. DNN training using MGO

The most important hyperparameters of DNN are the Weights and Biases which need to be updated according to the Dataset for best accuracy. In this work MGO algorithm is used to update the Weights and biases of the DNN.

MGO is a computer technique for minimizing a problem by repeatedly attempting to raise a candidate solution's quality score. In an MGO-based neural network training algorithm, the individual particles in the population represent different potential solutions to the optimization problem, which in this case is the configuration of the network's weights and biases that will result in the best performance on a given dataset. Each particle has a position in the search space that corresponds to a particular set of network weights and biases, as well as a velocity that determines how the particle moves through the space. The MGO algorithm continues iterating until some stopping criteria is met, such as a maximum number of iterations or a satisfactory level of performance on the training dataset. By using the collective intelligence of the particles to search for good solutions, the MGO-based training algorithm finds high-quality network configurations more quickly than other methods. In this work NMSE [37] as a Cost Function is used to train DNN structure

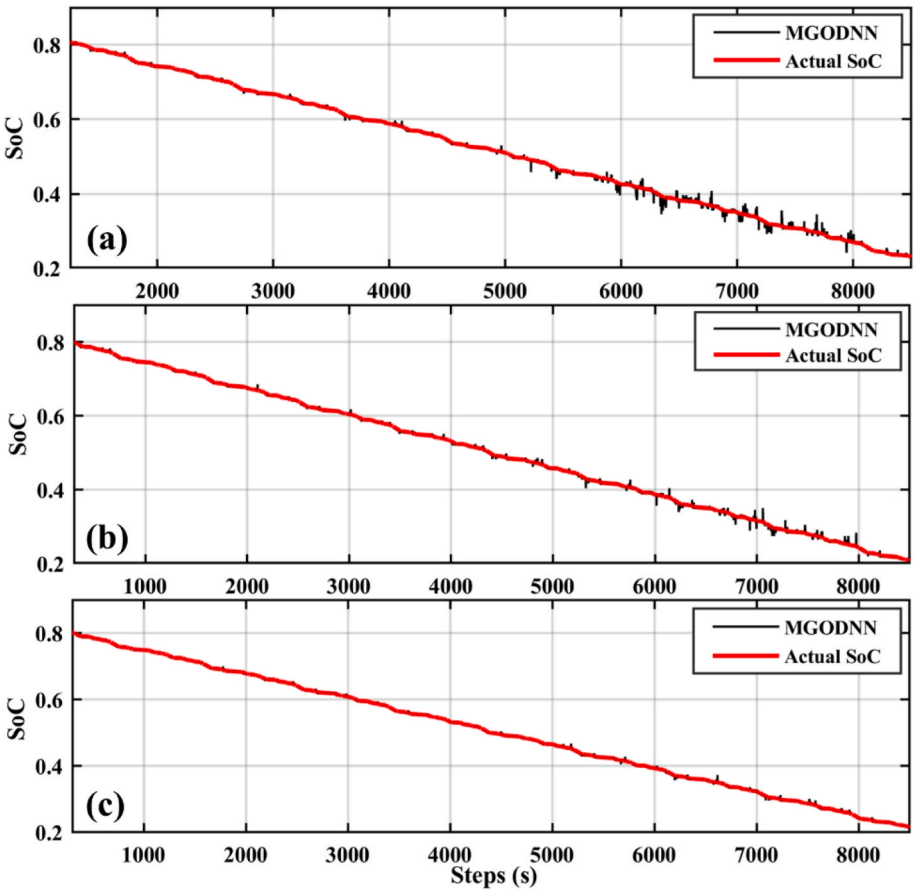

Fig. 10. SoC Estimation of MGODNN for BJDST Drive Cycle. (a) Drive cycle at 0° Temp (b) Drive cycle at 25° Temp (c) Drive cycle at 45° Temp.

using MGO optimization algorithm which is presented in Equation (13):

$$COST = \frac{1}{N}\sum_{i=1}^{N}\frac{(SoC(T)-SoC(P))^2}{(SoC(T))^2} \tag{13}$$

where N is total quantity of samples, SoC(T) is the actual SoC Value, SoC (P) is the predicted SoC by DNN. The flow chart of MGO-based DNN training is presented below in Fig. 4.

Several error metrics can be used to evaluate the performance of deep neural network (DNN) models, which are also called Loos or Cost Function. These include the NMSE, RMSE, and MAE [38]. The equations for these error metrics are as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(SoC(T)-SoC(P))^2 \tag{14}$$

$$NMSE = \frac{1}{N}\sum_{i=1}^{N}\frac{(SoC(T)-SoC(P))^2}{(SoC(T))^2} \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(SoC(T)-SoC(P))^2} \tag{16}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|SoC(T)-SoC(P)| \tag{17}$$

3. Experimental setup

In this work, we used MATLAB 2021a as for implementation of Hybrid Deep Learning Models for SoC estimation and the hardware used is AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz. The depth of a DNN (number of computational layers) can affect its performance in estimating the SoC. To study this, we trained a DNN with different hidden layers using the parameters in Table 1. Then, we evaluated the selected 3-layer DNN's performance in predicting the SoC on different drive cycles using the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) error metrics [39]. We repeated this process with increasing the depth of the DNN.

3.1. Drives cycles of EV

Drive cycles are a standard method of evaluating the characteristics of electric vehicles (EVs), e.g. energy utilization and releases [40]. Some common drive cycles used in EV testing include the FUDS, US06, BJDST, DST [41]. Fig. 5 illustrates sample plots of current, voltage, temperature, and state of charge for the different drive cycles used in this work.

3.2. SoC dataset

In this study, we utilized data from the CALCE Research Group, which was made available to us [41]. The data was collected using various drive cycles on a cylindrical INR 1865020 R LiNiMnCoO2/NMC Li-ion battery cell, using standard charging and discharging protocols. The cell was first charged using a constant current/constant voltage protocol and then discharged at three different temperatures (0 °C, 25 °C, and 45 °C). The specifications of the battery used in the study are included in Table 2.

3.3. Data diversity

To generate a superior training environment, multiple data sets are

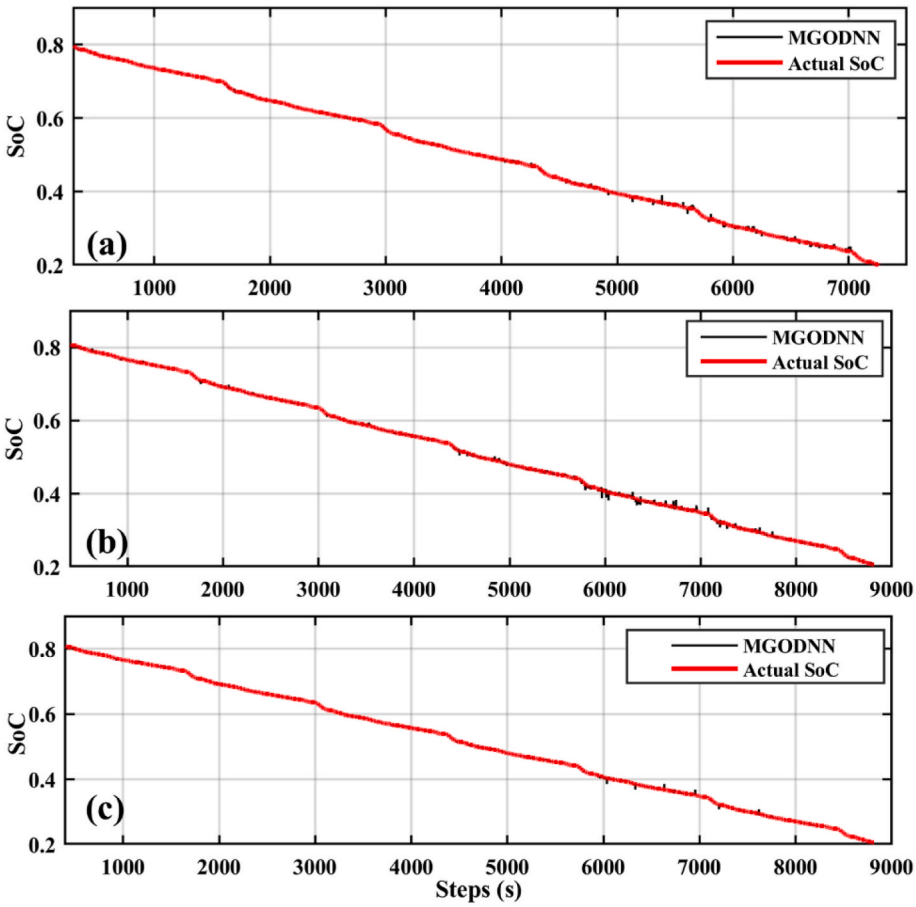

Fig. 11. SoC Estimation of MGODNN for FUDS Drive Cycle. (a) Drive cycle at 0° Temp (b) Drive cycle at 25° Temp (c) Drive cycle at 45° Temp.

highly effective [42]. 16 distinct cases for data are utilized using combination of four drive cycle data sets and training models. The correlation matrix (CM) is employed to show the relationship between several variables to highlight strong association between multiple variables with strength and direction of the relationships. Each cell in the matrix contains a correlation coefficient ranging between −1 and 1. When the correlation coefficient is 1, it means that as one variable rises, the other variable rises as well and for −1 vice versa. There is no connection between the variables when the correlation coefficient is 0. CM advocates for the identification of patterns and trends in the data while identifying multicollinearity in multiple regression models. The analysis shows a strong positive relationship SoC with voltage and discharge capacity (Disch) as compared to current density. Moreover, urban settings (BJDST and US06) impact the SoC more comparatively. Lower temperatures, repeated discharging and charging also impacts the SoC in long run. Fig. 6 below shows the strong diversity of datasets at STC conditions.

4. Results and discussion

In this section first the evaluation of different layer DNN structure is presented on DST dataset. From the comparative analysis, the 3-layer DNN structure is selected for SoC estimation and then tested on all drive cycle tests. In the last, comparative analysis is presented with the competing techniques, which shows that proposed Hybrid model outperforms in SoC estimation.

4.1. Evaluation of DNN architectures

For effective training and testing of DNN models, the number of hidden layers plays an important role. In this work, we investigated DNN models with up to 5 hidden layers to check the performance and select the best performing model. The evaluation indices used are NMSE, RMSE, MAE, RE and Time taken for training the model. Comparative evaluation of different DNN models is presented in Table 3 and cost vs epochs analysis is presented in Fig. 7.

In single layer neural network structure, the training and testing time is very low as compared to other models, but the evaluation indices values are very high which indicates that the single layer neural network doesn't perform well in this estimation problem. 2-layer, 3-layer and 4-layer DNN models perform well on SoC estimation dataset with the cost of higher time. The lowest evaluation indices value is achieved by 3-layer DNN structure which is also a good trade-off between cost value and time. 3-layer DNN achieves NMSE value 0.0010 which is up to 80% less as compared to other structures. Therefore, 3-layer DNN structure is proposed for SoC estimation in this work. All these structures are trained using the proposed MGO algorithm.

After the selection of DNN structure, the proposed MGO-DNN needs to train on dataset to check the training performance. All the 4 datasets are divided into 80-20% training and testing data. Fig. 8 shows the comparison of cost vs epochs by competing techniques with proposed MGO-DNN. In all 4 drive cycle datasets, MGO-DNN effectively reduces the cost function over the iterations and achieves less cost as compared to MFA-DNN, PSO-DNN and BP-DNN. Fast reduction of cost function by MGO-DNN shows that MGO is very effective to update the weights and biases. BP-DNN permanently stuck in local minima while training the DNN and MFA-DNN reduces the cost function effectively. Therefore, MGO as a DNN training algorithm for SoC estimation is validated.

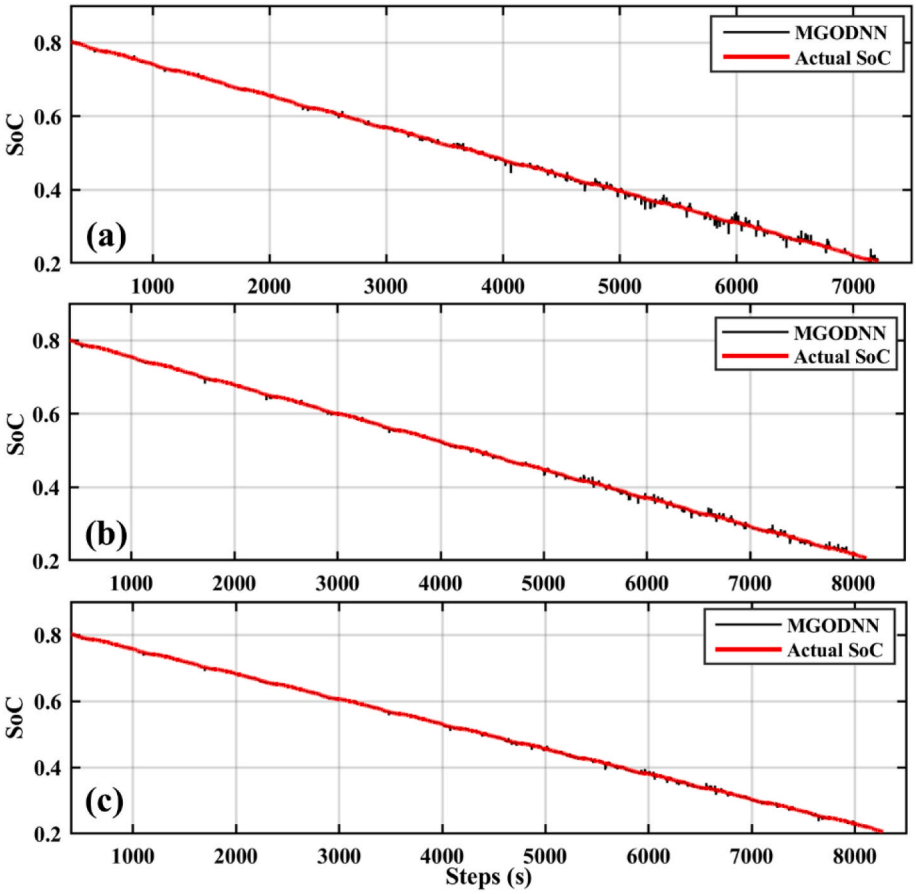

Fig. 12. SoC Estimation of MGODNN for US06 Drive Cycle. (a) Drive cycle at 0° Temp (b) Drive cycle at 25° Temp (c) Drive cycle at 45° Temp.

4.2. Testing evaluation of proposed technique

With MGO as a training algorithm, the proposed DNN structure can perform well under different drive cycle tests for SoC approximation of Li-ion batteries in EV. Evaluation of proposed technique is presented in Table 4. MGO-DNN is compared with other hybrid DNN models i.e., Mayfly-based DNN (MFA-DNN), Particle Swarm Optimization based DNN (PSO-DNN), and classical Back Propagation based DNN (BP-DNN). The competing techniques are tested on 4 drive cycles datasets i.e., DST, BJDST, FUDS, and US06 at different temperatures ($0^oC, 25^oC, 45^oC$). The parameters used to evaluate the testing performance are NMSE, RMSE, MAE, RE and Time.

In case 1, which is SoC estimation for DST drive cycles, the estimated SoC for different temperatures by MGO-DNN is shown in Fig. 9. The NMSE achieved by MGO-DNN, MFA-DNN, PSO-DNN and BP-DNN is 0.0010, 0.0063, 0.0241 and 0.0157 respectively. As shown in Table 4, MGO-DNN achieves low NMSE, RMSE, MAE and RE as compared to MFA-DNN, PSO-DNN and BP-DNN but the time taken by BP-DNN is low as compared to MGO-DNN and other techniques. Time utilized by MGO-DNN, MFA-DNN, BP-DNN and PSO-DNN is 197.3s, 758.4s, 247.1s, and 181.2s respectively. As shown in Fig. 9, MGO-DNN performs very well for DST drive cycle at all temperatures.

In case 2, which is SoC estimation for BJDST drive cycles, the estimated SoC for different temperatures by MGO-DNN is shown in Fig. 10. The NMSE achieved by MGO-DNN, MFA-DNN, PSO-DNN, and BP-DNN is 0.0021, 0.0056, 0.0320, and 0.0345 respectively. As shown in Table 4, MGO-DNN achieves low NMSE, RMSE, MAE, and RE as compared to MFA-DNN, PSO-DNN, and BP-DNN but time taken by BP-DNN is low as compared to MGO-DNN and other techniques. The time utilized by MGO-DNN, MFA-DNN, BP-DNN and PSO-DNN is 220.6s, 876.9s, 261.3s, and 198.4s respectively. As shown in Fig. 9, MGO-DNN performs very well for BJDST drive cycle on all temperatures.

The proposed MGO-DNN is also tested on FUDS and US06 drive cycles dataset which also validates that MGO-DNN achieves less error under all temperatures, but the time performance is second best after BP-DNN. The SoC estimation by MGO-DNN for FUDS and US06 datasets is shown in Figs. 11 and 12.

To check the estimation performance of proposed MGO-DNN, the relative error is also calculated and shown in Fig. 13. This shows that the MGO-DNN performs well when the SoC value is high but estimated value starts deviating as discharging occurs.

4.3. Comparative analysis

In this work MGO based DNN model is presented for SoC estimation. Proposed work is compared with more estimation techniques. To validate the performance of proposed technique, it is also compared with state-of-the-art estimation techniques presented in literature. Table 5 shows that multiple efforts are made by authors for accurate SoC estimation using long-short term memory recurrent neural network (LSTM-RNN) and Back Propagation based DNN. All the comparative techniques are tested on same drive cycle tests, and it is clear that MGO-DNN achieves less NMSE and RMSE errors (up to 66% less) as compared to LSTM-RNN architecture.

4.4. Gragner causality test

The Granger causality test (GCT) is a statistical hypothesis test that assesses whether past and present values of a set of m1 time series variables, called the “cause” variables, affect the predictive distribution of a distinct set of m2 time series variables, called the “effect” variables [44]. A cyclic process is modelled as a series of events. The estimation

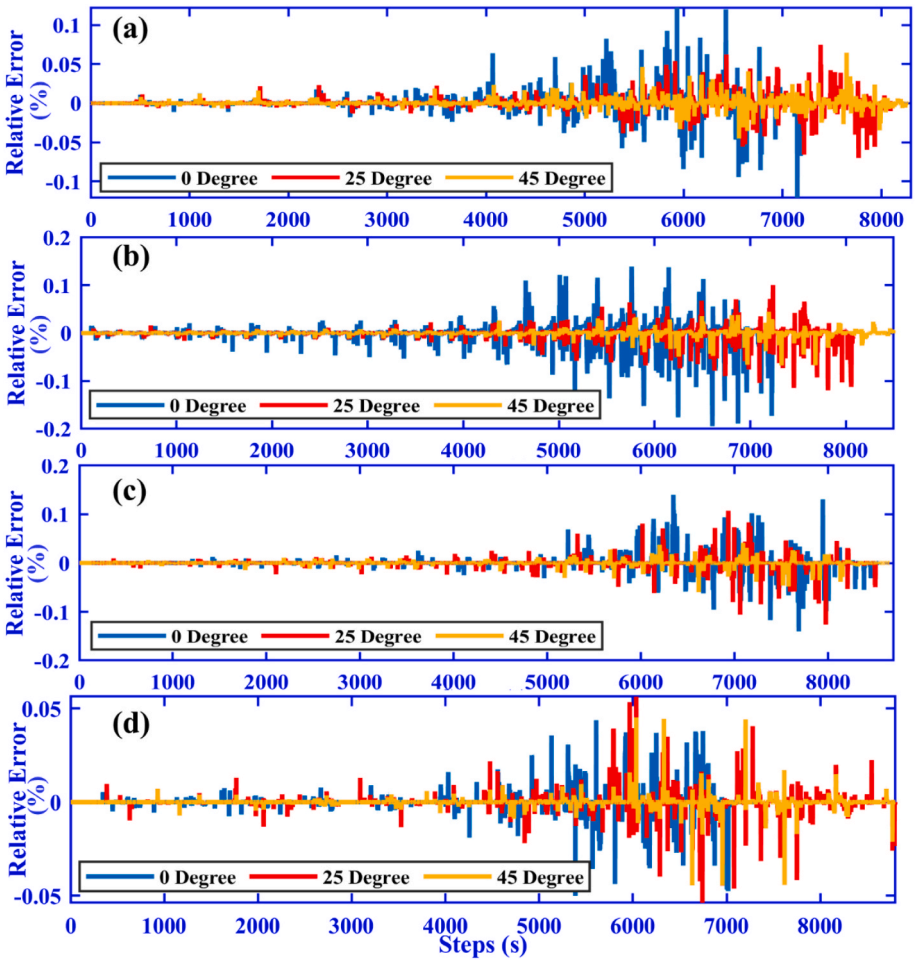

Fig. 13. Relative Error comparison of MGODNN on different Drive cycle tests. (a) US06 Drive Cycle (b) (b) DST Drive Cycle (c) BJDST Drive Cycle (d) FUDS Drive Cycle.

Table 5
Comparative Analysis of MGO-DNN with Recently Proposed Techniques for SoC estimation.

Technique	Drive Cycles	Temperature	Error %
Proposed (MGO-DNN)	DST, BJDST, FUDS, US06	$0^oC, 25^oC, 45^oC$	NMSE, 0.1% RMSE, 0.2%
LSTM-ACKF [43]	FUDS, US06	$0^oC, 25^oC, 40^oC$	RMSE, 0.9%
BP-DNN [41]	DST, BJDST, FUDS, US06	$0^oC, 25^oC, 45^oC$	NMSE, 0.49%
LSTM-RNN [30]	US06, FUDS	$0^oC, 25^oC, 40^oC$	RMSE, 0.4%

Table 6
Granger causality test comparison.

Step #	Explanation
Step 1:	State the null hypothesis and alternate hypothesis i.e. y(t) confines Granger-cause x(t)
Step 2:	Choose the lags: The results should not be sensitive to lags. It is convenient to pick several values and run the Granger test several times to see if the results are the same for different lag levels.
Step 3:	Calculate the f-value using Eq. (20) and Eq. (21),
Step 4:	Calculate f-statistic using Eq. (22)
Hypothesis:	(for X and Y data arrays)
	H0: lagged values of X do not Granger-cause Y
	Ha: lagged values of X do Granger-cause Y

Table 7
Granger causality test comparison of RE.

Drive Cycle	Technique	F-Value	SE	T statistic	P = value
DST	**MGO-DNN**	0.6328	0.150	0.0103	0.1039
	MFA-DNN	0.8258	0.380	0.0344	0.1007
	PSO-DNN	1.2014	1.450	0.0469	0.0950
	BP-DNN	1.2837	0.790	0.0234	0.1272
BJDST	**MGO-DNN**	0.7662	0.240	0.0103	0.1039
	MFA-DNN	0.8938	0.530	0.0344	0.1007
	PSO-DNN	1.2183	1.030	0.0469	0.0950
	BP-DNN	1.2837	1.870	0.0234	0.1272
FUDS	**MGO-DNN**	0.6328	0.020	0.0103	0.1039
	MFA-DNN	0.8258	0.110	0.0344	0.1007
	PSO-DNN	1.3092	1.220	0.0469	0.0950
	BP-DNN	1.2837	0.910	0.0234	0.1272
US06	**MGO-DNN**	0.6332	0.150	0.0103	0.1039
	MFA-DNN	0.8258	1.430	0.0344	0.1007
	PSO-DNN	1.3092	1.220	0.0469	0.0950
	BP-DNN	1.2837	1.870	0.0234	0.1272

solidifies the time series behavior in SoC prediction. Data should be transformed to eliminate the possibility of autocorrelation and unit roots should be removed as these roots skew the test results. The core steps of Granger causality are given in Table 6. The GCT results are summarized in Table 7. The value of $\beta_j = 0$ is calculated for all lags (j) using Equation (18) and Equation (19)

$$y(t) = \sum_{i=1}^{\infty} \alpha_i y(t-i) + c_1 + v_1(t) \tag{18}$$

$$y(t) = \sum_{i=1}^{\infty} \alpha_i y(t-i) + \sum_{i=1}^{\infty} \beta_i y(t-j) + c_2 + v_2(t) \quad (19)$$

With Equation (20) and Equation (21) check y(t) Granger-causes x(t) as:

$$y(t) = \sum_{i=1}^{\infty} \alpha_i y(t-i) + c_1 + u_1(t) \quad (20)$$

$$y(t) = \sum_{i=1}^{\infty} \alpha_i y(t-i) + \sum_{i=1}^{\infty} \beta_i y(t-j) + c_2 + u_2(t) \quad (21)$$

Equation (22) provides the f-statistic as:

$$F = \frac{(ESS_R - ESS_{UR})/q}{ESS_{UR}/(n-k)} \quad (22)$$

The results are summarized in Table 7 below. The proposed MGO-DNN closely associated SoC with an estimated value over all drive cycles as depicted by GCT.

5. Conclusions

In this work, the Mountain Gazelle Optimizer was used to train Deep Neural Networks for SoC estimation of Lithium-ion Battery for Electric vehicles. For the training and testing of the proposed model, MGO-DNN, four publicly available drive cycle datasets were used i.e., DST, BJDST, FUDS, and US06. The proposed model using MGO-trained DNN was able to accurately estimate the SoC of the lithium-ion battery in the electric vehicle. The use of MGO to train the DNN resulted in improved performance compared to other training methods, as demonstrated by the higher accuracy and lower error rates obtained. The proposed model was evaluated using several performance metrics, including NMSE, RE, EMSE, MAE and GCT. The results of the evaluation showed that the proposed model achieved a low NMSE, RMSE, MAE, and RE, indicating a high level of accuracy in the SoC estimation. For DST the NMSE, RMSE, MAE and RE are 0.0010, 0.002, 0.15, and 0.017. Achieving 12–170% lower error margins. Same performance is achieved in BJDST, FUDS and US06 drive cycles. Quantitative results demonstrate the effectiveness of the proposed model in estimating the SoC of a lithium-ion battery for an electric vehicle using INR 1865020 R LiNiMnCoO2/NMC Li-ion battery cell. Granger causality test also shows the minimum differential between actual and predicted SoC. The use of multiple performance metrics helps to provide a more comprehensive evaluation of the model's performance. The proposed model has the potential to be applied in real-time monitoring and control systems for electric vehicles, enabling more efficient and reliable operation.

Further research could be conducted to optimize the performance of the proposed model and explore its potential for use in other applications.

Credit author statement

Muhammad Hamza Zafar and Majad Mansoor: Conceptualization, Methodology, Resources, and Project Administration. **Mohamad Abou Houran:** Validation, Data Curation. **Noman Mujeeb Khan:** Conceptualization, Formal Analysis, Investigation. **Kamran Khan:** Visualization, Data Curation. **Syed Kumayl Raza Moosavi:** Validation, Formal Analysis. **Filippo Sanfilippo:** Supervision, Funding Acquisition, Investigation.

All authors were involved in the preparation and review of the original manuscript.

Declaration of competing interest

None.

All authors claim that there is not any conflict of interest regarding the above submission. The work of this submission has not been published previously. It is not under consideration for publication elsewhere. Its publication is approved by all authors and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder

Data availability

The reference is provided for the data used in this study.

References

[1] Wei X, Amin MA, Xu Y, Jing T, Yi Z, Wang X, Xie Y, Li D, Wang S, Zhai Y. Two-stage cooperative intelligent home energy management system for optimal scheduling. IEEE Trans Ind Appl 2022;58(4):5423–37.

[2] Xiong R, Cao J, Yu Q, He H, Sun F. Critical review on the battery state of charge estimation methods for electric vehicles. IEEE Access 2017;6:1832–43.

[3] Lin C, Mu H, Xiong R, Shen W. A novel multi-model probability battery state of charge estimation approach for electric vehicles using H-infinity algorithm. Appl Energy 2016;166:76–83.

[4] Sun L, Li G, You F. Combined internal resistance and state-of-charge estimation of lithium-ion battery based on extended state observer. Renew Sustain Energy Rev 2020;131:109994.

[5] Daina N, Sivakumar A, Polak JW. Modelling electric vehicles use: a survey on the methods. Renew Sustain Energy Rev 2017;68:447–60.

[6] Jeong Y-M, Cho Y-K, Ahn J-H, Ryu S-H, Lee B-K. Enhanced Coulomb counting method with adaptive SOC reset time for estimating OCV. In: 2014 IEEE energy conversion congress and exposition (ECCE). IEEE; 2014.

[7] Wang Q, Wang J, Zhao P, Kang J, Yan F, Du C. Correlation between the model accuracy and model-based SOC estimation. Electrochim Acta 2017;228:146–59.

[8] Sun F, Hu X, Zou Y, Li S. Adaptive unscented Kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles. Energy 2011;36(5):3531–40.

[9] Tang X, Liu B, Lv Z, Gao F. Observer based battery SOC estimation: using multi-gain-switching approach. Appl Energy 2017;204:1275–83.

[10] Xu J, Mi CC, Cao B, Deng J, Chen Z, Li S. The state of charge estimation of lithium-ion batteries based on a proportional-integral observer. IEEE Trans Veh Technol 2013;63(4):1614–21.

[11] Dai K, Wang J, He H. An improved SOC estimator using time-varying discrete sliding mode observer. IEEE Access 2019;7:115463–72.

[12] Takyi-Aninakwa P, Wang S, Zhang H, Li H, Xu W, Fernandez C. An optimized relevant long short-term memory-squared gain extended Kalman filter for the state of charge estimation of lithium-ion batteries. Energy 2022;260:125093.

[13] Lin X, Tang Y, Ren J, Wei Y. State of charge estimation with the adaptive unscented Kalman filter based on an accurate equivalent circuit model. J Energy Storage 2021;41:102840.

[14] Xu Y, Zhang H, Yang F, Tong L, Yan D, Yang Y, Ren J, Ma L, Wang Y. State of charge estimation of supercapacitors based on multi-innovation unscented Kalman filter under a wide temperature range. Int J Energy Res 2022;46(12):16716–35.

[15] Singh KV, Bansal HO, Singh D. Fuzzy logic and Elman neural network tuned energy management strategies for a power-split HEVs. Energy 2021;225:120152.

[16] Tan X, Zhan D, Lyu P, Rao J, Fan Y. Online state-of-health estimation of lithium-ion battery based on dynamic parameter identification at multi timescale and support vector regression. J Power Sources 2021;484:229233.

[17] Chen X, Chen X, Chen X. A novel framework for lithium-ion battery state of charge estimation based on Kalman filter Gaussian process regression. Int J Energy Res 2021;45(9):13238–49.

[18] Tian J, Wang Y, Liu C, Chen Z. Consistency evaluation and cluster analysis for lithium-ion battery pack in electric vehicles. Energy 2020;194:116944.

[19] Movassagh K, Raihan A, Balasingam B, Pattipati K. A critical look at coulomb counting approach for state of charge estimation in batteries. Energies 2021;14(14):4074.

[20] Hogrefe C, Waldmann T, Hölzle M, Wohlfahrt-Mehrens M. Direct observation of internal short circuits by lithium dendrites in cross-sectional lithium-ion in situ full cells. J Power Sources 2023;556:232391.

[21] Wang Y, Liu C, Pan R, Chen Z. Modeling and state-of-charge prediction of lithium-ion battery and ultracapacitor hybrids with a co-estimator. Energy 2017;121:739–50.

[22] Cui Z, Wang L, Li Q, Wang K. A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network. Int J Energy Res 2022;46(5):5423–40.

[23] Priya RP, Mishra S, Priyadarshi A. State-of-Charge estimation in lithium-ion battery for electric vehicle applications: a comparative review. Recent Advances in Power Electronics and Drives: Select Proceedings of EPREC 2022;2023:93–108.

[24] Jin Y, Yang L, Du M, Qiang J, Li J, Chen Y, Tu J. Two-scale based energy management for connected plug-in hybrid electric vehicles with global optimal energy consumption and state-of-charge trajectory prediction. Energy 2023;267:126498.

[25] Fan X, Zhang W, Zhang C, Chen A, An F. SOC estimation of Li-ion battery using convolutional neural network with U-Net architecture. Energy 2022;256:124612.

[26] Hu C, Ma L, Guo S, Guo G, Han Z. Deep learning enabled state-of-charge estimation of LiFePO4 batteries: a systematic validation on state-of-the-art charging protocols. Energy 2022;246:123404.

[27] Muduli UR, Beig AR, Al Jaafari K, Al Hosani K, Al-Sumaiti AS, Behera RK. Dual motor power sharing control for electric vehicles with battery power management. IEEE Transactions on Industrial Electronics; 2023.
[28] Kannan M, Sundareswaran K, Nayak PSR, Simon SP. A combined DNN-NBEATS architecture for state of charge estimation of lithium-ion batteries in electric vehicles. IEEE Transactions on Vehicular Technology; 2023.
[29] Semenoglou A-A, Spiliotis E, Assimakopoulos V. Data augmentation for univariate time series forecasting with neural networks. Pattern Recogn 2023;134:109132.
[30] Chen J, Zhang Y, Wu J, Cheng W, Zhu Q. SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output. Energy 2023; 262:125375.
[31] Liu Y, He Y, Bian H, Guo W, Zhang X. A review of lithium-ion battery state of charge estimation based on deep learning: directions for improvement and future trends. J Energy Storage 2022;52:104664.
[32] 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.
[33] Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S. Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv Eng Software 2022;174:103282.
[34] Patre PM, MacKunis W, Kaiser K, Dixon WE. Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure. IEEE Trans Automat Control 2008;53(9):2180–5.
[35] Cho H, Kim Y, Lee E, Choi D, Lee Y, Rhee W. Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks. IEEE Access 2020;8:52588–608.
[36] Liao L, Li H, Shang W, Ma L. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. ACM Trans Software Eng Methodol 2022;31(3):1–40.
[37] Jafarian A, Nia SM, Golmankhaneh AK, Baleanu D. On artificial neural networks approach with new cost functions. Appl Math Comput 2018;339:546–55.
[38] Zhang N, Shen S-L, Zhou A, Xu Y-S. Investigation on performance of neural networks using quadratic relative error cost function. IEEE Access 2019;7: 106642–52.
[39] Khan MK, Zafar MH, Rashid S, Mansoor M, Moosavi SKR, Sanfilippo F. Improved reptile search optimization algorithm: application on regression and classification problems. Appl Sci 2023;13(2):945.
[40] 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;162:871–82.
[41] How DN, Hannan MA, Lipu MSH, Sahari KS, Ker PJ, Muttaqi KM. State-of-charge estimation of li-ion battery in electric vehicles: a deep neural network approach. IEEE Trans Ind Appl 2020;56(5):5565–74.
[42] Shin H-C, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers RM. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imag 2016;35(5):1285–98.
[43] Tian Y, Lai R, Li X, Xiang L, Tian J. A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter. Appl Energy 2020;265:114789.
[44] Sun W, Liu M, Liang Y. Wind speed forecasting based on FEEMD and LSSVM optimized by the bat algorithm. Energies 2015;8(7):6585–607.