

Research Article

Accelerating AI-Based Battery Management System's SOC and SOH on FPGA

Satyashil D. Nagarale ¹ and B. P. Patil ²

¹Department of Electronics and Telecommunication Engineering, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India

²Department of Electronics and Telecommunication Engineering, Army Institute of Technology, Savitribai Phule Pune University, Pune, India

Correspondence should be addressed to Satyashil D. Nagarale; satyashil.nagrare@pccoepune.org

Received 5 December 2022; Revised 16 March 2023; Accepted 26 May 2023; Published 5 June 2023

Academic Editor: Yiming Tang

Copyright © 2023 Satyashil D. Nagarale and B. P. Patil. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Lithium battery-based electric vehicles (EVs) are gaining global popularity as an alternative to combat the adverse environmental impacts caused by the utilization of fossil fuels. State of charge (SOC) and state of health (SOH) are vital parameters that assess the battery's remaining charge and overall health. Precise monitoring of SOC and SOH is critical for effectively operating the battery management system (BMS) in a lithium battery. This article presents an experimental study for the artificial intelligence (AI)-based data-driven prediction of lithium battery parameters SOC and SOH with the help of deep learning algorithms such as Long Short-Term Memory (LSTM) and bidirectional LSTM (BiLSTM). We utilized various gradient descent optimization algorithms with adaptive and constant learning rates with other default parameters. Compared between various gradient descent algorithms, the selection of the optimal one depends on mean absolute error (MAE) and root mean squared error (RMSE) accuracy. We developed an LSTM and BiLSTM model with four hidden layers with 128 LSTM or BiLSTM units per hidden layer that use Panasonic 18650PF Li-ion dataset released by NASA to predict SOC and SOH. Our experimental results advise that the selection of the optimal gradient descent algorithm impacts the model's accuracy. The article also addresses the problem of overfitting in the LSTM/BiLSTM model. BiLSTM is the best choice to improve the model's performance but increase the cost. We trained the model with various combinations of parameters and tabulated the accuracies in terms of MAE and RMSE. This optimal LSTM model can predict the SOC of the lithium battery with MAE more minor than 0.0179%, RMSE 0.0227% in the training phase, MAE smaller than 0.695%, and RMSE 0.947% in the testing phase over a 25°C dataset. The BiLSTM can predict the SOC of the 18650PF lithium battery cell with MAE smaller than 0.012% for training and 0.016% for testing. Similarly, using the Adam optimization algorithm, RMSE for training and testing is 0.326% and 0.454% over a 25°C dataset, respectively. BiLSTM with an adaptive learning rate can improve performance. To provide an alternative solution to high power consuming processors such as central processing unit (CPU) and graphics processing unit (GPU), we implemented the model on field programmable gate array (FPGA) PYNQ Z2 hardware device. The LSTM model using FPGA performs better.

1. Introduction

Using fossil fuels has resulted in adverse environmental impacts such as air pollution and global warming, leading to increased health issues and other socioeconomic impacts worldwide [1]. Most countries are signing international agreements and implementing national policies to combat this environmental impact. Recently, there has been a significant focus on EVs powered by lithium batteries, owing to

the constraints associated with fossil fuels [2, 3]. To encourage the acceptance of EVs in the country, the central government of India announced several promotional measures in the previous ten years, including tax incentives for EV owners and public EV charging infrastructure development [4, 5].

Real-time monitoring of lithium battery parameters is crucial for the safety and optimum performance of the battery. This can be performed by accurately estimating SOC

and SOH [6]. However, this is challenging due to the nonlinear dynamics and electrochemical properties of the lithium batteries. Many technologies attempt to solve this challenge. As battery degradation begins immediately after manufacturing, to ensure the safe functioning of batteries, it is advisable to replace them once they reach approximately 70–80% of their original capacity [7].

Within this article, we are implementing data-driven AI-based predictive BMS to estimate SOC and SOH using deep learning algorithms such as LSTM and BiLSTM. We have implemented various stochastic gradient optimizers used in LSTM and compared various optimizers with adaptive learning rates and constant learning rates. The data-driven approach bypasses the need for knowledge about internal battery parameters and functioning [8]. Also, information on the physical model, chemical properties or reactions, filters, etc., of the battery is not required. It requires a considerable quantity of real-time battery data, which is readily available nowadays. We can extract battery features from historical data using advanced deep learning algorithms. Our research objective is to implement an FPGA-based AI-driven predictive BMS for electric mobility (E-mobility). We plan to predict a lithium battery SOC and SOH using LSTM and BiLSTM [9]. We have analyzed various gradient descent optimization algorithms and BiLSTM to improve the model performance. We also implemented the model on the FPGA PYNQ Z2 device with a limited dataset to measure SOH. The challenges and drawbacks associated with the PYNQ Z2 board are also discussed [10].

1.1. BMS SOC and SOH Parameters. A BMS is an electronic circuit that manages battery conditions for increased lifespan and safety. It controls and monitors the battery at the cell, module, and pack levels, while a lithium battery consists of cells arranged to meet voltage and capacity requirements. The BMS balances each cell unit to avoid degradation in EV performance. The BMS performs various tasks, including measuring system voltage, current, temperature, SOC, SOH, and RUL; minimizing charging time; maximizing battery life; and cell balancing. Thus, it ensures the safety and optimum performance of the battery [11].

An advanced BMS plays a pivotal role in driving the expansion of the EV industry. Innovating a novel battery technology that provides higher energy and power density and reduces cost is essential. An effective BMS with algorithms that can control and monitor real-time data of the battery and ensure the safety and reliability of the energy storage devices is required. The BMS monitors the battery pack that powers your EV and estimates the range for us [12]. Additionally, the BMS monitors the battery pack's health and safety during use. Lithium batteries operate under appropriate temperatures, SOC, SOH, and RUL conditions. The BMS estimates the energy stored, such as a SOC, SOH, and RUL, in real-time [13].

“SOC refers to the amount of charge remaining in the battery and is given by the residual capacity of the battery divided by its nominal capacity.” SOC can be mathematically represented by the following equation:

$$\text{SOC} = \frac{Qa}{Qn} * 100\%, \quad (1)$$

where Qa is the battery charge available at present and Qn is the nominal capacity which is constant throughout the battery life (both in Ah).

$$\text{SOC} = \text{SOC}_0 - \frac{\int_0^t I_{\text{bat}} dt}{Q_n} * 100\%, \quad (2)$$

where SOC_0 is the initial value of SOC and I_{bat} is the battery current. Accurate SOC estimation is critical in effectively controlling battery charge and discharge and extending battery lifespan. However, SOC relies on several factors, such as electrochemical reactions, temperature, material degradation, and ageing cycles, as it represents the internal state of a battery. Hence, in the past few decades, research has been concentrated on devising a professional approach for estimating SOC [14].

The SOH represents the difference between the health of a battery being used and a new battery. It is the ratio of the maximum battery charge to the rated capacity:

$$\text{SOH} = \frac{Q_{\text{max}}}{Q_n} * 100\%, \quad (3)$$

where Q_{max} represents the battery's maximum charging capacity and Q_n represents nominal capacity* RUL of lithium battery = total number of charging/discharging life cycles* – actual number of charging/discharging life cycles (*nominal capacity and total number of lifecycles are given by manufacturers and are fixed throughout battery life. SOH degrades as the number of charging cycles increase).

Data-driven algorithms can estimate battery states using historical data without the physical model of the battery. This helps to save considerable time and effort. Optimized AI-based methods are widely used in natural language processing (NLP), computer vision, image processing, video processing, speech recognition, etc., and are areas of study within the field of artificial intelligence [15, 16].

Optimizing neural networks in AI is a challenge. Overfitting, characterized by a large gap between training and test errors, can be mitigated using regularization techniques and adaptive learning rate algorithms such as Adadelata, SGD, Adam, Adamax, AdaGrad, FTrl, and RMSProp [17].

1.1.1. Contribution

- (i) Developed LSTM and BiLSTM models to predict SOC and SOH
- (ii) Implemented the model on field programmable gate array (FPGA) PYNQ Z2 hardware device to provide an alternate solution to high power consuming processors such as CPU and GPU
- (iii) Compared FPGA time and space complexity with CPU and GPU.

In our experiments, we have correctly tuned LSTM/BiLSTM model hyperparameters to overcome the issue of overfitting. In addition, we investigated the effect of selecting the gradient optimization algorithm on the model's

accuracy, especially regarding the SOC/SOH estimation of a lithium battery. We deployed a popular LSTM/BiLSTM model strategy, effectively using early stopping, dropout, adaptive learning rates, splitting testing, and training data.

1.2. Related Work in AI-Based Algorithms, SOC and SOH Estimation, and FPGA. Almaita et al. [18] implemented a data-driven BiLSTM NN method wherein MAE is evaluated at less than 0.62%, demonstrating both robustness and accuracy of SOC prediction. Sun et al. [19] proposed an ML-based LSTM-RNN with extended input (EI) and constrained output (CO) utilized for SOC estimation in lithium batteries, ensuring both accuracy and robustness, named EI-LSTM-CO wherein RMSE and MAE are estimated at 1.3% and 3.2%, respectively, on unknown data. To improve the SOH prediction accuracy of the lithium batteries, Sun et al. [19] implemented ICA-BiLSTM to predict lithium battery SOH. Mean square error (MSE) was used as a performance metric and compared with GRU, LSTM, and BPNN. Stighezza et al. [20] SVM was employed as a regression method for SOC estimation. The model was simulated in MATLAB Simulink and converted to HDL to implement on FPGA Board with RMSE at 1.4%. Chemali et al. [21] presented an alternative solution for low-cost hardware to improve the throughput and resource usage of a SOC estimator for lithium-ion batteries which attains a low MAE of 0.573% at a constant ambient temperature and an MAE of 1.606% on a dataset with temperature ranging from 10 to 25 degrees Celsius. Kim [22] proposed an AI-based SOH estimation method that achieves high accuracy, with approximately 98.4% prediction accuracy for rule-based operation profiles and around 99.5% for dynamic driving profiles. A similar work by Luciani et al. [23] focuses on designing and validating a data-driven SOC estimation method using IoT-based HIL experiments. The method achieves an impressive 98% accuracy in real-time hardware testing. Khumprom and Nita concluded in their article [24] that data-driven methods using advanced AI-based algorithms achieve higher performance and accuracy with the drawback of higher computational time. Li et al. [25] proposed that GRU RNN requires no physical model. Observable variables such as voltage, current, and temperature are directly linked to SOC through mapping. The method is evaluated on two public datasets, yielding MAEs of 0.86%, 1.75%, and 1.05%. Li proposes the BMS hardware prototype as part of future work. Jemmali et al. [26] proposed an FPGA implementation of the EKF algorithm with low power consumption and high-speed advantages. He and He [27] implemented FPGA-based deep neural network to provide an alternative solution for CPU and GPU. Experimental results compared with CPU and GPU achieves improved performance in terms of speed and energy. Bobulski and Kubanek [28] applied deep learning technique for plastic waste classification system which would help to solve the plastic waste problem. Zhenhua Cui et al. proposed hybrid model CNN-BWGRU (CNN and bidirectional weighted gated recurrent unit) to enhance the performance of SOC prediction, which is able to predict MAE and RMSE 0.0127 and 0.0171 with 300 BWGRU units for 1000 iterations.

Based on the above literature review, data-driven approaches are accurate and robust. This implementation does not require knowledge or modelling of the battery's internal parameters but requires a substantial volume of data. Researchers suggest that deep learning algorithms such as RNN, LSTM, and BiLSTM algorithms offer more accuracy and advantages than ML model-based algorithms. DL algorithms are ML subsets consisting of three or more network layers. Each layer of a deep learning NN is a series of complex mathematical operations such as multiplication and accumulation between the input data and some constraints such as activation functions, weights, and bias. So, we may require on-chip memory and a reliable processor to store data and real-time data processing.

Currently, CPUs and GPUs are used to process the data, but are slower than GPUs [29]. Therefore, DL algorithms are implemented on high-speed processors such as GPUs [30]. But GPUs consume high power and are expensive, which is a challenge in real-time applications. FPGAs are scalable and configurable and use low power. FPGA is also a good choice considering GPU and CPU. FPGA provides low-cost hardware and improves throughput [31]. Based on the summary, findings, and current research development, we proposed AI-based data-driven SOC and SOH prediction algorithms using a low-cost hardware FPGA to improve the performance.

2. Proposed Data-Driven AI-Based BMS SOC and SOH

The proposed method started with data preparation or data preprocessing of the lithium battery. Data preprocessing includes data cleaning, which helps improve data quality, transform data into a valuable and efficient format, and enhance the model's performance. Data preprocessing is essential to avoid duplication and eliminate missing data fields. Furthermore, it carried out outlier detection to identify any data points that lie outside the anticipated range. Additionally, it implemented data normalization techniques to guarantee consistency and coherence across the entire dataset. After that, a dataset is split into training and testing sets. The model is trained and tested on different datasets. After an experimental result analysis, the LSTM and BiLSTM model is built with the correct number of LSTM units. Various gradient descent techniques are implemented and examined based on performance metrics. An optimized gradient descent technique model is then ready to implement on an FPGA board. All these steps are shown in Figure 1.

The proposed method is an AI-based data-driven approach, eliminating the need to manually construct a physical or mathematical model for varying temperatures and other unstable on-road real-time conditions. The proposed method for estimating SOC and SOH can be applied to different types of batteries, enabling accurate prediction of SOC and SOH. We have employed two datasets of lithium-ion batteries. In the dataset split step, the dataset is split into training, testing, and validation sets and adapted to suit the LSTM model. We trained the models on the first 60% of the dataset and tested and validated it on the remaining 40%. Data is split to keep overfitting in balance. The splitting of data in training and testing affects the LSTM model. Data quality is vital in a data-driven approach to obtain good

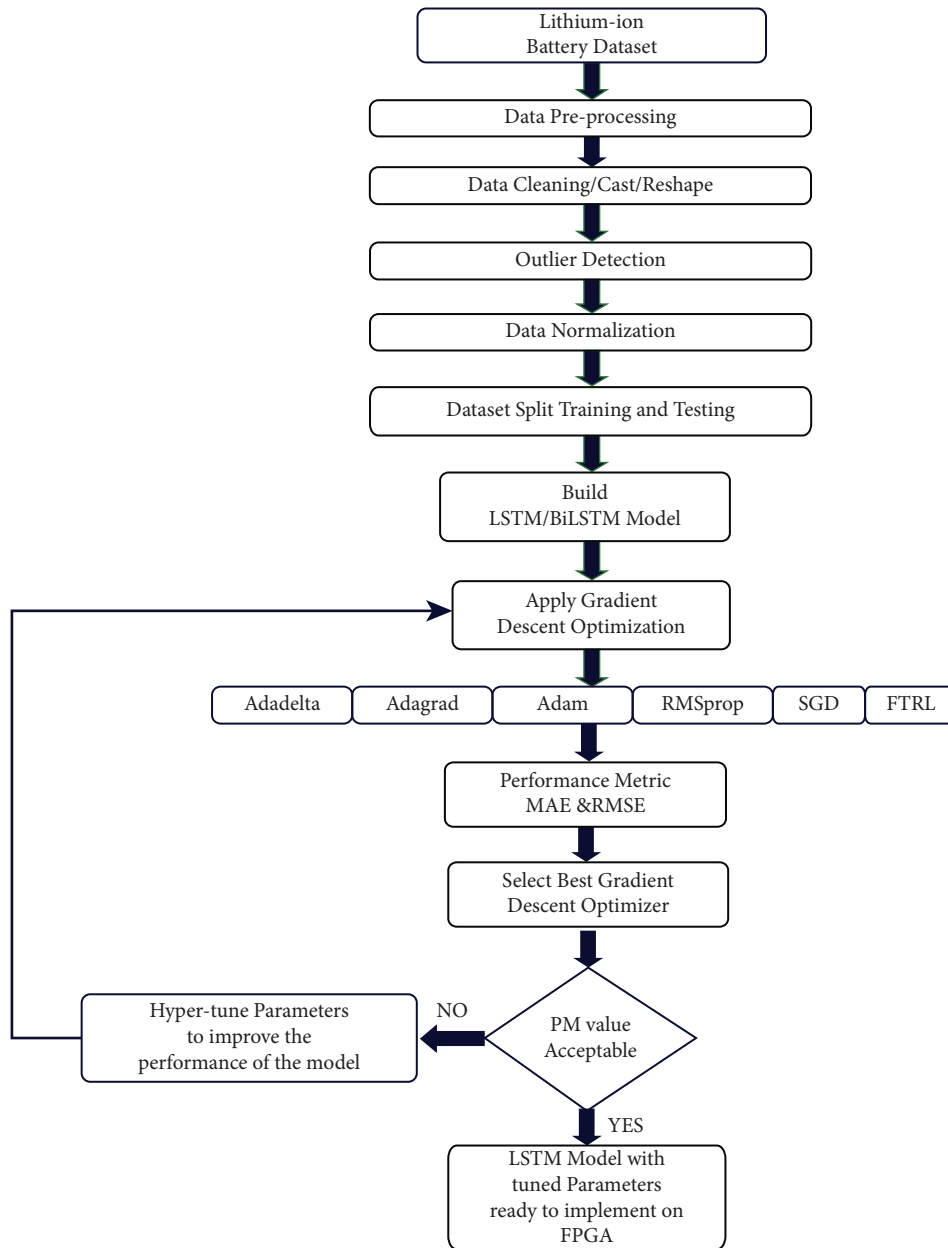


FIGURE 1: Proposed AI-based BMS SOC and SOH on an FPGA.

results. Various gradient optimization algorithms are implemented, and Adam gradient optimization is selected based on their performance. Here, we have tried constant learning rates as well as adaptive learning rates. Adaptive learning rates provide better results compared to constant learning rates. While comparing gradient descent algorithms, we have selected default parameters other than adaptive learning rates. The data-driven prediction techniques can accurately predict lithium battery SOC and SOH by computing a battery's current, voltage, temperature, and other parameters. This lets us avoid the physical battery model and mathematical model-based approaches [32].

The LSTM deep learning networks were constructed with four layers: a sequence input layer (with 1 feature), Uni-LSTM/BiLSTM Layers (with 128 hidden units), and a fully

connected layer. Additionally, we experimented with LSTM layers featuring 256 hidden units and 100 units or fewer in each hidden layer. However, the LSTM model with 128 units in each hidden layer demonstrated superior performance while optimizing hardware cost. Various gradient descent algorithms are implemented with default parameters, and their performance is measured using MAE and RMSE. The model's performance was evaluated, and the best optimizer was selected based on the results. This selection prepares the model for implementation on an FPGA. To implement the model on the FPFA board, we have chosen the PYNQ Z2 board. To develop LSTM/BiLSTM model, we have used LSTM from the latest TensorFlow and keras libraries. PYNQ Z2 is an open-source project to enhance Python productivity. The latest TensorFlow and keras library is not supported by the

PYNQ Z2 platform, so we need to design or develop LSTM from scratch [33]. We have developed the LSTM model from scratch without taking inbuilt support. The model performance is tested on limited data and compared with CPU and GPU. The proposed methodology is discussed in detail in the outcome analysis and discussion section.

In this article, we have effectively tuned the hyper-parameters of the LSTM/BiLSTM model to address the challenge of overfitting. Furthermore, we investigated the impact of selecting different gradient optimization algorithms on the model's performance, with a particular focus on SOC/SOH estimation for lithium batteries. We deployed a popular strategy in the LSTM/BiLSTM model of effectively using early stopping, dropout, adaptive learning rates, splitting testing, and training data. This has two critical benefits as compared to model-driven approaches.

- (i) AI-based data-driven algorithms accurately predict the relationship between observable quantities (voltage, current, and temperature) and unobservable quantities (SOC and SOH)
- (ii) The need to identify electrochemical machining (ECM) parameters is circumvented.

Therefore, we intend to conduct in-depth research and propose a solution using the AI algorithm as the next step.

Multiple complex factors impact battery SOC and SOH. None of the methods proposed by specialists can guarantee the accuracy and practicability of estimate SOC and SOH estimation. Our current research outlays the following technical challenges in the process of SOC and SOH prediction:

- (1) Enhancing the accuracy, robustness, and effectiveness of SOC and SOH estimation without increasing model complexity
- (2) To simplify the estimation models and enable their implementation on cost-effective hardware such as FPGA rather than relying on GPU/CPU.

The solutions to the above are correlated. Therefore, the aim is to find an acceptable trade-off between accuracy and computational resources. We accomplished this by proposing an approach to curtail the sources of error in the SOC and SOH prediction. This article attempts to predict SOC and SOH using deep learning algorithms LSTM and BiLSTM. We implemented this model on an FPGA device. Our research aims to develop reconfigurable hardware for an advanced AI-based data-driven model.

3. Data-Driven AI-Based Algorithms and PYNQ Z2 FPGA Device

This section introduces the fundamental theories of RNN, LSTM, BiLSTM, lithium battery dataset, and the PYNQ Z2 device.

3.1. Recurrent Neural Networks (RNNs). RNNs are deep learning models with cyclic connections that can store information for a long time. Unlike feed-forward networks,

RNNs utilize input from previous neurons, making them suitable for sequential time series prediction. Figure 2 illustrates an unfolded RNN architecture for predicting SOC and SOH which is specifically designed for this purpose.

An RNN has a feedback loop, as shown in the above figure. This feedback loop can be unfolded in time steps. Input and output at time step t is denoted as INPUT and OUTPUT. The INPUT at the time sequence t consists of temperature, voltage, current, and other battery parameters. These parameters are mapped to SOC and SOH. We can provide SOC and SOH as input at time step t and estimate output the error between actual and predicted SOC and SOH. The hidden state at time step t is represented by h_t , while the output SOC $_{t+1}$ and SOH $_{t+1}$ value at time step $t + 1$.

3.2. LSTM. LSTM, a more advanced variant of RNN, incorporates time-cyclic neural networks and memory mechanisms to capture long-term dependencies and mitigate the vanishing gradient problem effectively. Unlike RNNs, LSTM is free from the issue of long-term dependencies as there are four interacting cells against a single neuron and a unique storage unit structure. These characteristics of the LSTM help in effectively forecasting battery states. In addition, the accuracies are visualized through a plot of training and testing battery states values. LSTM employs three gates to control data flow, allowing information retention over time. The cell stores information, while the gates manage memory. This architecture enables LSTM to classify, analyze, and predict time series data with varying durations. The forward pass of an LSTM cell architecture [34] is represented as shown in Figure 3.

At time step " t ," the variables F_t , I_t , O_t , C_t , and h_t represent the components of the LSTM architecture. Specifically, F_t denotes the forget gate, I_t represents the input gate, O_t represents the output gate, C_t represents the cell state, and h_t represents the hidden state. The sigmoid activation function is denoted by the symbol σ . The symbol \odot represents the Hadamard product, which signifies the element-wise multiplication of vectors. The output of an LSTM cell in time series mostly depends on three gates: (1) cell state gate (C_t) long-term memory of LSTM cell, (2) hidden state (h_t) which stores the previous output, and (3) the input gate (I_t) stores the input data at the present step. These gates decide what information to be entered in the LSTM network and out of the LSTM network [35]. The initial step in the LSTM architecture cell involves the forget gate (F_t), responsible for determining which information from the previous cell state C_{t-1} will be discarded at time step t and what information should be stored to pass in the LSTM network. Utilizing a sigmoid activation function, the forget gate determines whether to discard or keep information from the previous cell state. 0 signifies low weightage and forgetting, while 1 indicates retaining everything. Thus, the decisions of gates are based on the current input I_t , cell state C_t , hidden state h_t , and the LSTM network's weights and biases [36].

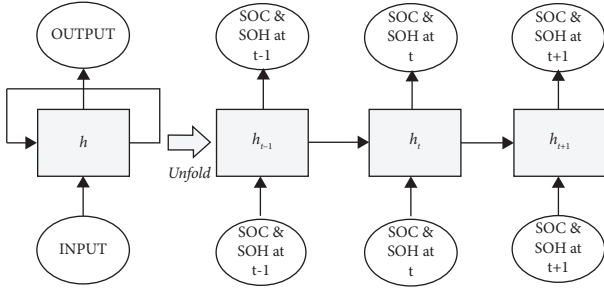


FIGURE 2: RNN architecture for SOC and SOH.

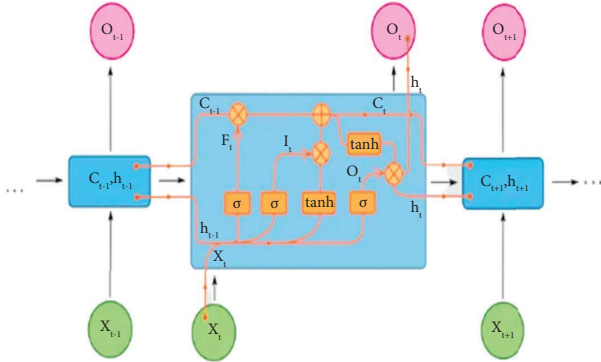


FIGURE 3: LSTM architecture.

3.2.1. *Forget Gate.* It conditionally decides what to forget from the network.

$$F = \sigma(W_f x_t + W_f h_{t-1} + b_f), \quad (4)$$

where “ x_t ” represents the current input at time step t and “ b_f ” denotes the bias term. “ W_f ” denotes the weight matrix associated with forget gate “ F_t .”

3.2.2. *Input Gate.* The sigmoid function determines the binary values (0 or 1) that impact memory, while the tanh function assigns weights to data on a scale of -1 to 1 .

$$I_t = \sigma(W_i x_t + W_i h_{t-1} + b_i). \quad (5)$$

The tanh layer produces the new value C_t as shown in the following equation:

$$C_t = \tanh(W_c x_t + W_c h_{t-1} + b_c). \quad (6)$$

The cell state C_t is updated by combining it with the previous cell state, as mentioned in equation (6). The forget gate (F_t) and input gate (I_t) collectively determine whether the values from the previous LSTM cell should be stored or forgotten. The cell state (C_t) at time t is represented by the following equation:

$$C_t = F_t \odot C_{t-1} + I_t \odot C_t. \quad (7)$$

The final step involves the output gate (O_t) using a sigmoid activation function to determine which portion of the cell state is transferred to the hidden state (h_t). Within the hidden state, the cell state (C_t) is passed through the tanh

function and then multiplied by the output gate (O_t) to retain only the desired output. The output gate (O_t) and hidden state (h_t) are shown in equations (8) and (9), respectively.

3.2.3. *Output Gate.* The LSTM architecture, as depicted in Figure 3, makes conditional decisions regarding what output to produce based on the block’s input and memory.

$$O_t = \sigma(W_o x_t + W_o h_{t-1} + b_o), \quad (8)$$

$$h = O_t \odot \tanh(C_t). \quad (9)$$

3.3. *Bidirectional LSTM (BiLSTM).* Bi-LSTM enhances the capabilities of LSTM by training the input data in both the forward and backward directions, thereby expanding its potential for capturing contextual dependencies. Bi-LSTMs capture information that the one-way network may ignore. To enhance the performance of the LSTM model, the BiLSTM model can be applied [33].

3.4. *PYNQ Z2 (Python Productivity for Zynq FPGA).* PYNQ Z2 is a Xilinx-supported open-source project designed to facilitate ZYNQ device usage. Python makes programmable logic available to designers. PYNQ-enabled boards are easily programmable using Python in PyCharm or Jupyter Notebook. While developing the model in PyCharm or Jupyter Notebook, the latest Tensorflow or Python libraries are not supported by the PYNQ environment. So, we need to create LSTM from scratch, followed by implementation on the PYNQ Z2 device [37].

3.5. *Lithium Battery Dataset.* In this article, we selected two lithium battery datasets with different features.

- (1) We utilize the Panasonic 18650PF Li-ion battery data, provided by Dr Phillip Kollmeyer from the University of Wisconsin-Madison, to estimate SOC.
- (2) Li-ion battery dataset released by NASA to estimate SOH.
- (3) The Panasonic 18650PF Li-ion battery dataset comprises a collection of ten drive cycles; only the discharge Cycle 1 at temperature 25 C is used in our experiments [38]. Table 1 displays a summary of the Panasonic 18650PF Li-ion battery.

3.5.1. *The Li-Ion Battery Dataset of Four Batteries Released by NASA.* At room temperature, numbers 5, 6, 7, and 18 underwent three distinct operational profiles, including charge, discharge, and impedance. This dataset includes information on lithium batteries over several charge and discharge cycles at room temperature and was used for SOH prediction [39]. Within the NASA battery dataset, each of the four battery groups provides a singular maximum capacity value for every charge-discharge cycle. As a result, we

TABLE 1: Summary of the panasonic 18650PF Li-ion battery.

Attributes	Description
Capacity	Min. 2750 mAh Typ. 2900 mAh
Nominal voltage	3.6 V
Min/Max voltage	2.5 V/4.2 V
Charging	CC-CV, std. 1375 mA, 4.20 V, 4.0 h
Temperature	Storage: -20°C to 50°C Discharge: -20°C to 60°C Charge and discharge: 0°C to 45°C
Energy density	Gravimetric: 207 Wh/kg Volumetric: 577 Wh/l

have extracted the health features on a per-cycle basis. The battery's maximum capacity decreases over time as it ages.

Note that this article aims to assess the performance of LSTM and BiLSTM with adaptive learning rates and constant learning rates with different gradient descent optimizations. So, we focused on developing an advanced deep learning model with minimum error and hardware realization. The benefits of FPGA can be leveraged as real-time estimations through AI-based algorithms which are bound to require excessive computational power.

4. Outcome Analysis and Discussion

4.1. Performance Metrics RMSE and MAE. The AI-based LSTM/BiLSTM models for predicting SOC and SOH were trained and tested using two distinct datasets. The performance of these models is assessed using various evaluation metrics, such as RMSE and MAE metrics, providing insights into their effectiveness [38].

$$\begin{aligned}
 \text{MAE} &= \frac{1}{m} \sum_i (|\text{SOC}_i^* - \text{SOC}_i|), \\
 \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^m (\text{SOC}_i^* - \text{SOC}_i)^2}{m}}.
 \end{aligned}
 \tag{10}$$

Battery datasets used for experimental purposes: to measure SOC, we utilized a 2.9 Ah Panasonic 18650PF Li-ion cell, while for measuring SOH, we relied on the Li-ion battery dataset provided by NASA. The input characteristics of lithium batteries are plotted in Figures 4 and 5, showing the relationship between the battery temperature vs. SOC and battery voltage vs. SOC. Temperature and voltage rate strongly influence battery performance. The battery should be operated at a proper temperature range for optimum performance. Figure 4 shows temperature versus SOC and indicates that the battery operates well at ambient temperature [40]. Figure 5 illustrates the correlation between battery voltage and SOC.

The terminal voltage of the lithium battery exhibits a nearly proportional relationship with the SOC. Hence, understanding the relationship between voltage and SOC is crucial for accurately estimating and effectively managing the BMS [41].

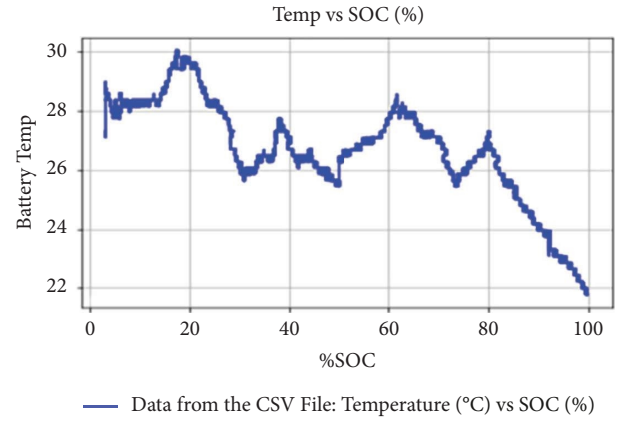


FIGURE 4: Temperature (°C) vs. % SOC.

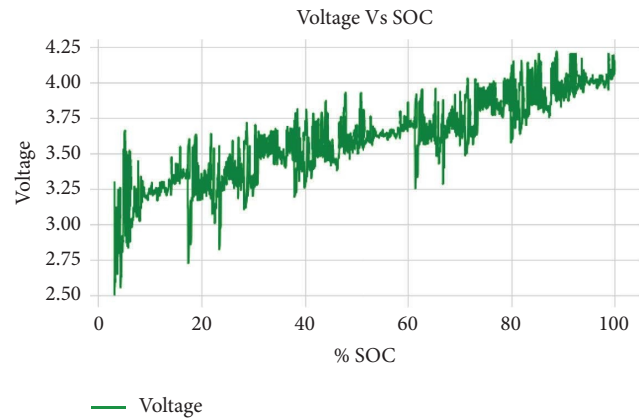


FIGURE 5: Voltage vs. SOC (%).

4.2. Dataset Released by NASA. Figures 6 and 7 display characteristic curves illustrating the degradation of SOH to the number of charging/discharging cycles. The figures demonstrate that the battery's health degrades as cycles increase. The SOH degradation starts at about 120–130 cycles or when SOH is around 0.70%. Therefore, we can conclude that the battery's maximum and energy storage capacity is reducing slowly [42].

4.3. Data-Driven LSTM/BiLSTM Model Development. Initially, we built the LSTM model of four hidden layers with 150 units in each layer, variable dropout in various layers, constant learning rate = 0.001, and other default parameters. The LSTM model architecture depicted in Figure 8 consists of four hidden layers, with each layer comprising 150 units. Various gradient descent optimizer algorithms were taken from the TensorFlow and keras packages.

Next, actual SOC at sequential time steps is input to LSTM. The dataset is partitioned into separate training and testing sets, and the testing data is used for validation. We have incorporated several gradient descent algorithms in our implementation, including Adadelta, Adagrad, SGD, and RMSprop. Follow the regularized leader (FTRL) and Adam. We trained and validated the LSTM model during

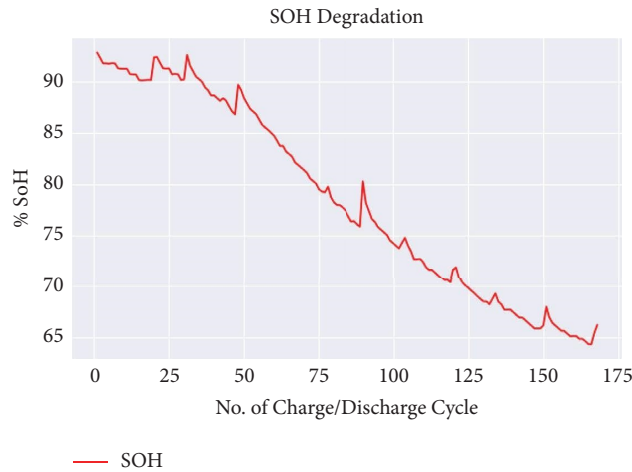


FIGURE 6: SOH degradation vs. number of charge and discharge cycle.

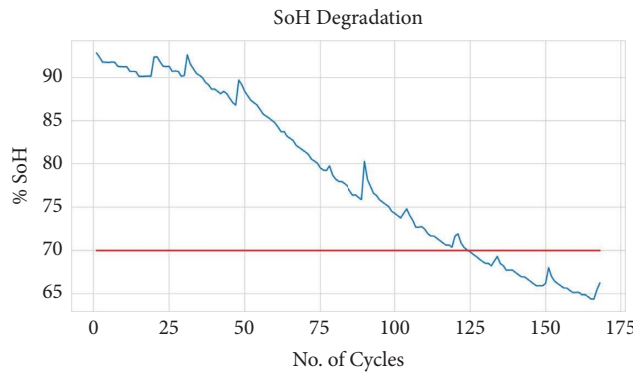


FIGURE 7: SOH degradation vs. number of cycles with threshold line.

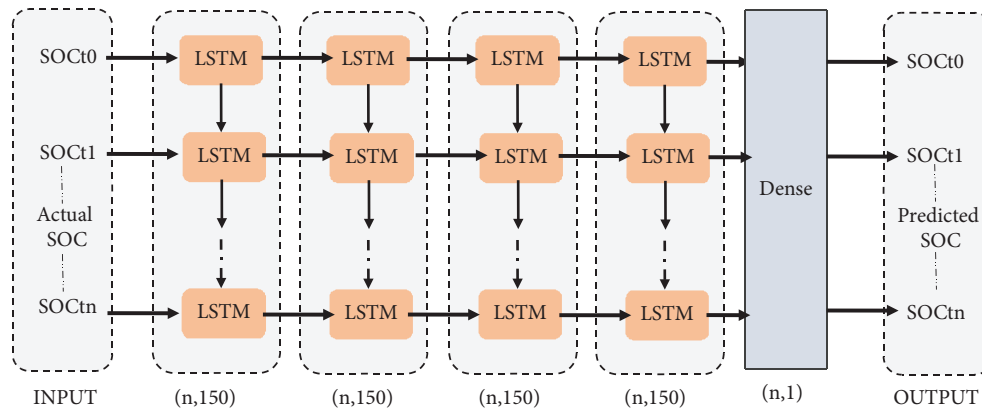


FIGURE 8: Architecture of the first LSTM model.

implementation for 100, 300, and 500 epochs. With constant learning rate = 0.001 and other default parameters. The dataset is divided at 80% in testing and 20% in training. The forecasting performance metrics are measured using MAE and RMSE. Comparisons between them are given in Table 2.

Based on the results presented in Table 2, it can be observed that the Adam optimizer outperforms other optimizers, yielding superior results. RMSprop also produces

a better result, but it has an underfitting problem. Another observation is that the performance metric RMSE of a test is more significant than the RMSE of a train. This indicates that the LSTM model has a problem with overfitting. To overcome this, we have tried tuning the hyperparameter of the LSTM model by simplifying the network, reducing the number of LSTM units, early stopping, and reducing dropout regularization.

TABLE 2: Comparison of various gradient descent optimization algorithms.

Gradient descent optimizing algorithms	Number of iteration	Training (80%)		Testing (20%)	
		MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
Adadelta	100	53.430	56.180	12.350	13.711
	300	32.331	36.693	15.386	19.796
	500	16.431	18.260	41.835	43.647
Adagrad	100	22.0153	26.612	33.643	34.862
	300	19.069	21.0565	48.851	49.698
	500	19.053	21.035	49.504	50.340
SGD	100	17.211	17.211	39.223	41.162
	300	17.148	19.268	39.422	41.350
	500	17.148	19.269	39.423	41.352
RMSprop	100	7.936	10.093	10.205	11.445
	300	8.659	9.549	4.494	6.211
	500	4.001	4.4998	13.063	16.603
FTRL	100	45.781	48.966	11.339	12.640
	300	15.344	17.386	47.081	48.708
	500	15.307	17.380	47.271	48.892
Adam	100	1.059	1.7103	3.702	5.306
	300	0.996	1.453	5.693	8.124
	500	0.894	0.998	11.561	14.988

4.3.1. Early Stopping. This is a critical feature available in keras callback. Stop training when a performance metric has stopped improving [43], as illustrated in Figure 9.

The dropout stochastic regularization technique prevents neural networks from overfitting. During training, specific neurons are randomly excluded or “dropped out.” This dropout technique introduces noise to the hidden state, enhancing the model’s robustness and preventing overfitting [44]. Our results suggest that reducing dropout improves performance and overcomes the problem of overfitting. In general, if we cared about the LSTM deep learning model performance on the training dataset, we expect a model to have perfect performance on the testing dataset. Splitting the dataset into separate training and testing sets holds significant importance. Training and testing the model on distinct datasets is crucial to ensure accurate evaluation and robustness. To overcome the problem of overfitting, the testing data should be almost equal to the training dataset.

Based on the findings presented in Table 2, we have chosen Adam as a gradient descent optimizer and varying dropout regularization with different numbers of LSTM units but without early stopping. So, LSTM network is run for 500 iterations with 64 LSTM units in each hidden layer with varying dropout regularization, and the dataset is divided into 65% in testing and 35% training to reduce the problem of overfitting as shown in Figure 10, and the results are shown in Table 3.

Table 3 shows exciting results. We have taken only 64 LSTM units in this model in each hidden layer to simplify the network. As we reduce the dropout regularization, LSTM network performance improves regarding MAE and RMSE. By reducing dropout regularization, the performance of the LSTM network improves. However, still model is overfitting. To address the overfitting issue, the number of LSTM units in each hidden layer was

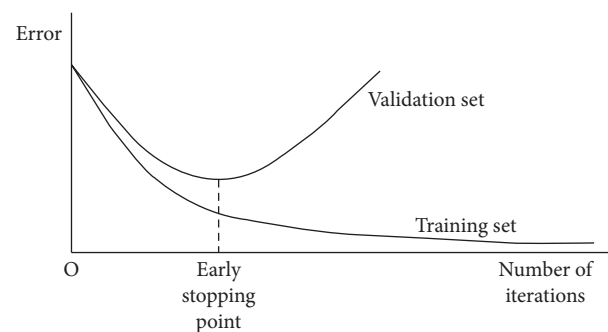


FIGURE 9: Early stopping.

increased to 100 from the previous 64 LSTM units. Additionally, dropout was reduced. The results of these adjustments are displayed in Table 4.

Following are the key observations based on results shown in Tables 3 and 4:

- (1) As we reduced dropout, MAE and RMSE improved
- (2) As we decreased the number of LSTM below 100 units, MAE and RMSE increased, which degraded the performance of the LSTM model.

So, we have concluded from Table 4 that model must have LSTM units of more than 100 with less dropout in each layer. However, the model still faced issues of overfitting. We then used 128 LSTM units in each layer and compared it with 256 units and 64 LSTM units with optimizing gradient descent. We used “Adam” with and without early stopping, as shown in Tables 5 and 6.

In Table 5, we have run the LSTM model for 200 iterations without early stopping. LSTM model with 256 units performs better. Increasing the number of LSTM units in each hidden layer leads to improved performance; however, it also results in increased costs.

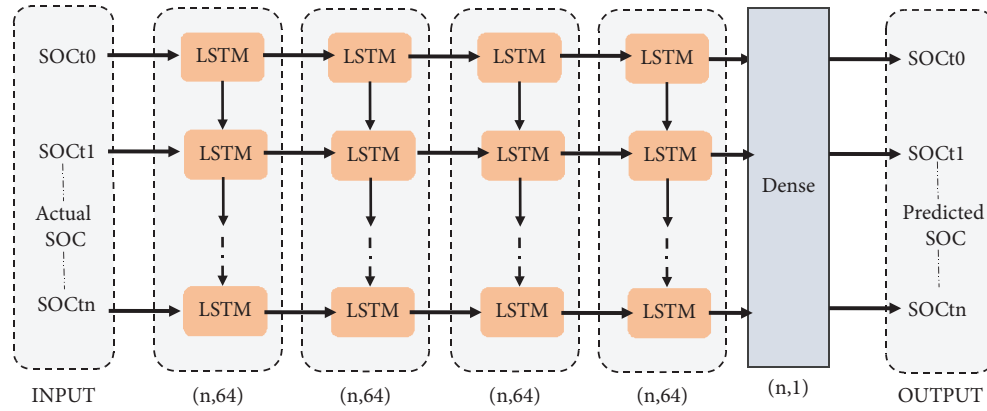


FIGURE 10: LSTM model architecture.

TABLE 3: Adam optimizer varying dropout regularization and without early stopping.

Number of iterations (without early stopping)	#No. of LSTM in each hidden layer	Dropout	Train (65%)		Test (35%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
500	64	0.7	6.217	8.050	19.153	24.112
		0.3	3.418	4.720	15.820	19.010
		0.2	1.889	2.833	13.779	18.007

TABLE 4: Comparison with 64 LSTM units with 100 units with decreasing dropout.

Number of iteration (without early stopping)	#No. of LSTM in each hidden layer	Dropout	Train (65%)		Test (35%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
100	64	0.2	1.259	1.524	4.141	5.520
	100	0.1	0.647	0.732	4.811	6.624

TABLE 5: Without early stopping.

Number of iteration (with early stopping)	#No. of LSTM in each layer	Dropout	Train (60%)		Test (40%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
200	64	NIL	0.067	0.070	0.824	1.196
	128		0.066	0.071	0.772	1.020
	256		0.0190	0.0224	0.336	0.462

TABLE 6: With early stopping.

Number of iteration	#No. of LSTM in each layer	Dropout	Train (60%)		Test (40%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
200	64	NIL	0.067	0.070	0.824	1.196
188	128		0.0179	0.0227	0.695	0.947
99	256		0.0135	0.0165	0.210	0.303

In Table 6, we have run the LSTM model for 200 iterations with early stopping. It is observed that with early stopping, the model performed very well compared to those without early stopping. A model with zero dropouts produced better results. This means early stopping affects the model performance.

The results in Table 6 show that the LSTM model with 256 units in each hidden layer performs well compared to 64

and 128 units in each layer. Though a model with 256 units in each layer performs exceptionally well, it will increase the cost of the LSTM network. To balance all these parameters and save cost, a model with zero dropout and early stopping, the LSTM model with 128 units, seems to be the best choice. For further investigation, a model with 128 LSTM units in each hidden layer, as illustrated in Figure 11. The LSTM model is built with the following specification:

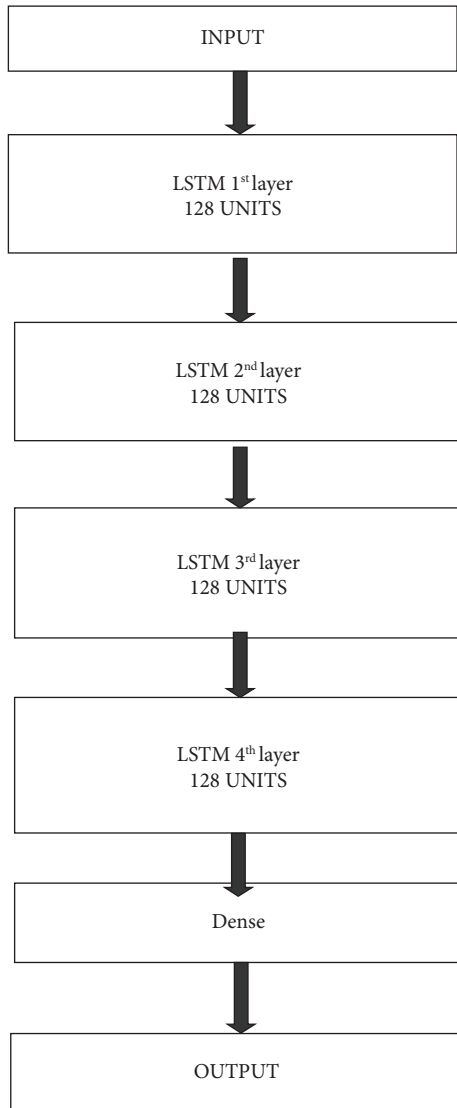


FIGURE 11: LSTM model with 128 LSTM units in each hidden layer.

- (1) Simplify the network
- (2) Without dropout
- (3) With early stopping
- (4) Dataset split (60 : 40).

Figure 12 shows LSTM Model loss plot in training and testing (validation) phase.

Figure 13 shows the SOC prediction of LSTM

Also, we have developed the BiLSTM model as shown in Figure 14. Gradient descent is optimized using “Adam” for the BiLSTM Model. Results are shown in Table 7.

The BiLSTM model produces better results with 128 BiLSTM cells in each layer without dropout with early stopping. Results are shown in Table 7.

Table 7 results show that the BiLSTM model without dropout performs well. For further investigation, we have developed a BiLSTM model without dropout. The BiLSTM model is developed for 32, 48, 64, and 128 LSTM units in each hidden layer, and results are shown in Table 8.

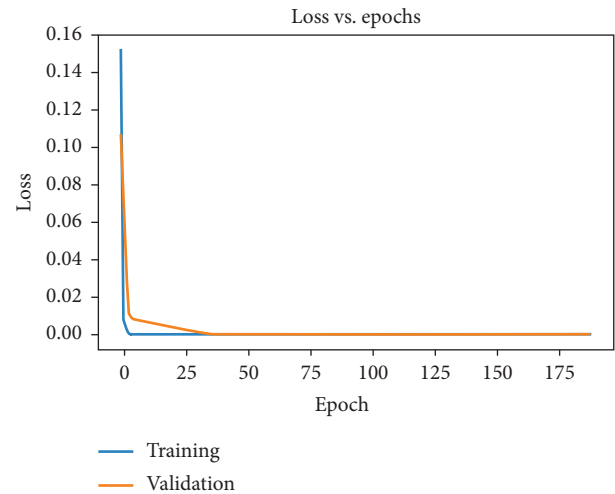


FIGURE 12: Model loss.

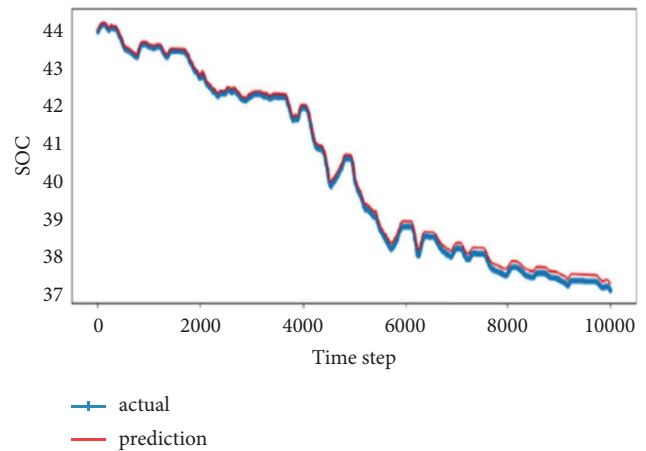


FIGURE 13: SOC prediction of LSTM model.

As the number of LSTM units in each hidden layer increases, the model performance improves with fewer iterations required. The model performs far better with 128 units. Table 8 analyses the BiLSTM model with an adaptive learning rate using different configurations of hidden units in each layer: 32, 64, and 128. As can be observed, the BiLSTM model with 128 units in each hidden layer without dropout performs better than with dropout. Datasets splitting of training and testing is performed with an 80 : 20 ratio, shown in Table 9 and compared with the BiLSTM model with datasets splitting 60 : 40 ratio in Table 8.

Datasets splitting 60 : 40 ratio performs well compared to 80 : 20. Datasets splitting also affects the model’s performance. Also, we can select an adaptive learning rate instead of a constant learning rate. BiLSTM is the best choice, but it will increase the network cost.

Table 10 shows that the BiLSTM model performs slightly better than LSTM with early stopping. Based on the results shown in Table 10, model loss and SOC prediction are plotted in Figures 15 and 16, respectively.

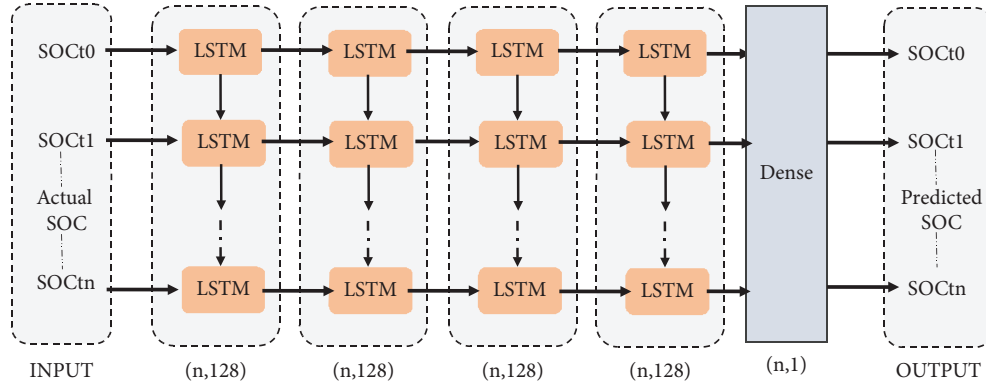


FIGURE 14: BiLSTM model architecture (128 LSTM units in each layer).

TABLE 7: BiLSTM model with adaptive learning rate with and without dropout.

Number of iterations (ES)	No. of BiLSTM in each layer	Dropout	Train (60%)		Test (40%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
122	64	0.3	0.163	0.224	3.90	4.958
197		NIL	0.015	0.018	1.263	1.848

TABLE 8: Analysis of BiLSTM model with adaptive learning rate.

Epocs (with ES)	No. of BiLSTM in each layer	Dropout	Train (60%)		Test (40%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
231	32	NIL	0.022	0.0276	1.677	2.389
209	48		0.020	0.023	1.134	1.670
197	64		0.015	0.018	1.263	1.848
170	128		0.012	0.016	0.326	0.454

TABLE 9: Datasets splitting of training and testing (80 : 20).

Epocs (with ES)	No. of BiLSTM in each layer	Dropout	Train (80%)		Test (20%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
128	128	NIL	0.036	0.049	0.453	0.531

TABLE 10: Comparison of LSTM vs. BiLSTM without dropout with early stopping.

LSTM type	Epochs (with ES)	No. of units	Train (60%)		Test (40%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
BiLSTM	170	128	0.0120	0.0160	0.326	0.454
LSTM	188		0.0179	0.0227	0.695	0.947

As we plan to implement the LSTM model on the FPGA board, we must consider the resources available on-chip. LSTM, with early stopping and an adaptive learning rate, is the best choice, requiring fewer resources than BiLSTM. This developed model is used for predicting time steps t and $t + 1$. As we increased the look-back time steps, the model performed poorly.

BiLSTM model is developed with 128 LSTM units in each layer with look-back 5, 10, and 15. As we increased the look back, the model's performance is degrading, i.e., MAE and RMSE are increasing, as seen in Table 11. Based on the previous results, we used the LSTM model with 128 units in each hidden layer to estimate SOH on NASA battery sets. Figure 17 shows the architecture of the LSTM model SOH prediction.

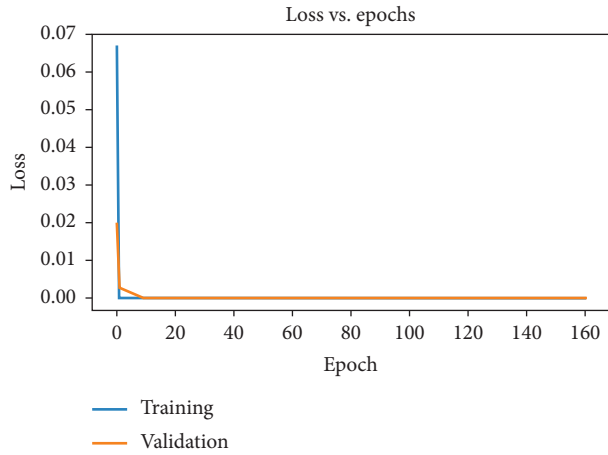


FIGURE 15: Model loss.

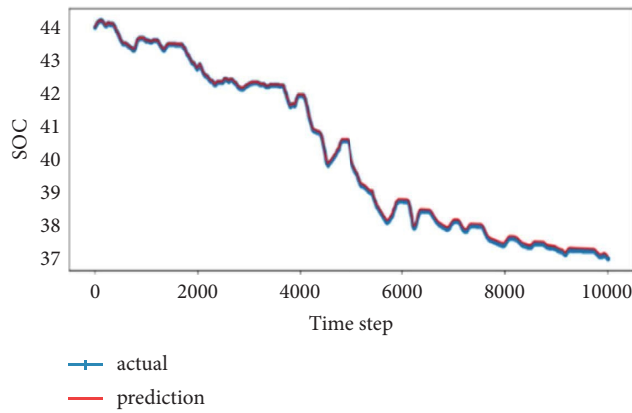


FIGURE 16: SOC prediction.

TABLE 11: Optimizing gradient descent using “Adam” for BiLSTM model with look back.

Epochs	BiLSTM	Look back	Train (60%)		Test (40%)	
			MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
170		00	0.012	0.016	0.326	0.454
285	128	5	0.011	0.016	0.329	0.400
161		10	0.072	0.095	2.036	2.753
90		15	0.443	0.461	2.315	2.988

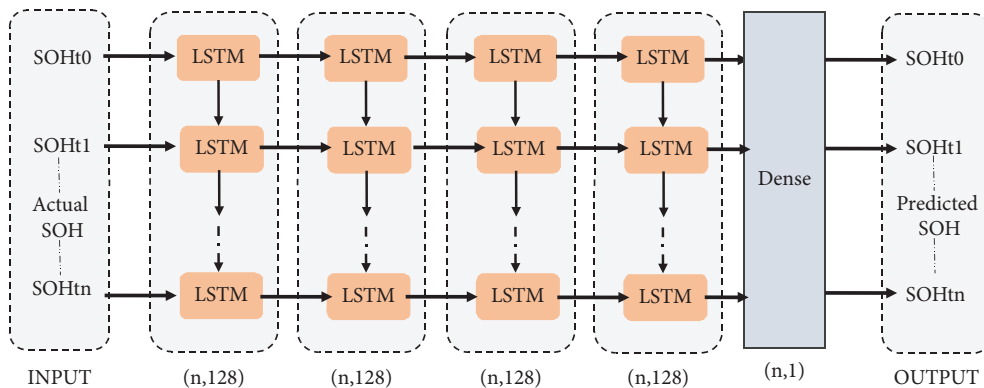


FIGURE 17: Architecture LSTM model SOH prediction.

TABLE 12: The LSTM model accurately predicts the SOH of lithium-ion batteries.

Epochs (with ES)	No. of units in each layer	Train (60%)		Test (40%)	
		MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
50	128	0.000133	0.000462	0.002224	0.00256
100	128	0.000339	0.000557	0.001991	0.00236
200	128	0.000299	0.000551	0.000628	0.00085

TABLE 13: LSTM model with look back.

Epochs (with ES)	Look back	Train (60%)		Test (40%)	
		MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
100	5	0.0006664	0.000811	0.0011000	0.001446
	10	0.0003096	0.000606	0.000324	0.000463
	100	0.0001639	0.000778	0.012962	0.019902

Using the same LSTM model, we predicted the SOH of lithium-ion batteries on various datasets and displayed the results in Table 12.

Table 12 shows the results for the LSTM model that are run for 50, 100, and 200 iterations, and the model is free from overfitting. RMSE in training and testing is almost equal. The LSTM model is also run for look back 5, 10, and 100. Results are shown in Table 13.

The difference between RMSE values under train and test is very minimal. The model SOH dataset released by NASA does not have a problem with overfitting or underfitting. Table 14 presents a comparison of the results obtained from the LSTM and BiLSTM models with the findings from existing studies.

4.4. Monitoring Real-Time SOC and SOH Poses Hardware Development Challenges That Need to Be Addressed. We are implementing our module on a System on Chip (SOC) based PYNQ Z2 board, a programmable device. The PYNQ Z2 development platform combines an ARM Cortex-A9 processor with FPGA fabric. It allows for control of FPGA overlays through a Python interface, enabling seamless software and hardware components integration. The drawback of PYNQ Z2 is that it does not support the latest keras and Tensorflow libraries. We cannot use predefined LSTM modules from these libraries. So, we have developed an LSTM model without using Tensorflow or keras. For our results on PYNQ Z2, we have tried many alternatives to get the highest possible accuracy. We collected approximately 650 data samples to predict a lithium battery's SOH. Due to significant discrepancies between the actual and predicted data, we experimented with various methods and approaches to reduce errors. Eventually, we chose to train the model using 1000 cycles, a learning rate of 0.5, and 128 LSTM units. The following tables show the parameters we experimented with and the corresponding training and test accuracies.

Table 15 shows that the model is run for 1000 iterations with varying learning rates. As we decreased the learning rate, the CPU performed slower. The selection of the

learning rate must be in the proper range. As we decreased the learning rate, model performance improved. We observed that at a lower learning rate, the model takes more CPU time and moves to overfitting. We have tested our model for a learning rate of 0.01, and the results are shown in Table 16.

We have run a model for 100, 500, and 1000 cycles with a learning rate of 0.01, but it takes more CPU time than 0.5. So, to keep the balance between CPU performance and accuracy, we have run the model with a learning rate of 0.5 for 100, 500, and 1000 iterations implemented on the PYNQ Z2 board. We compared the speed and resource utilization between CPU and GPU. Table 17 shows the system specification on which the LSTM model is implemented. In this experiment, CPU/GPU and PYNQ Z2 are utilized.

The LSTM model with a limited dataset runs for 100, 500, 1000, 1500, and 2500 iterations. Performance is measured based on speed and memory usage. Comparison results are shown in Table 18.

Results for the LSTM model with a constant learning rate of 0.5 and 128 LSTM units are in Table 18. CPU and GPU performance is almost identical for small data volumes and fewer iterations. It is not recommended to use GPU for small applications. GPU is faster for large applications, as seen when the model runs for 2500 iterations. PYNQ Z2 board is much slower because specifications are low-end. The LSTM model is implemented on PYNQ Z2 device. The results are presented in Figure 18, which shows the relationship between the SOH and the number of data samples.

The design and development cycle is significantly faster with Python than with C/C++ and hardware description languages (HDLs). However, we cannot use Python to enhance the model's performance on FPGA. To achieve this, we must develop our model using C/C++ or HDLs. Code quality is better with Python, but most inbuilt advanced libraries are not supported by PYNQ, which can estimate the resource utilization of hardware on FPGA. To improve the model's performance in terms of speed, power, and area, PYNQ with Python may not be the most suitable option.

TABLE 14: Comparison of the results of the LSTM and BiLSTM models with existing findings.

Ref. no./year	Methodology	Performance metrics MAE/RMSE (error estimation)	Observations
[45] 1 Jan 2023	EI-LSTM-CO is an advanced LSTM-RNN model designed for battery SOC estimation. It utilizes extended input and constrained output techniques to enhance accuracy	RMSE and MAE estimates on unknown data are less than RMSE 1.3% and MAE 3.2%	An improved machine learning model-based method is proposed for SOC estimation
[18] 1 Aug 2022	LSTM-based models are utilized for accurate state of charge (SOC) prediction	The proposed model achieves a maximum standard error (MSE) of less than 0.62%, indicating its high estimation accuracy	The accuracy of SOC estimation using LSTM surpasses that of other methods
[19] 28 Mar 2022	Adaptive weighting factors ICA-Bi-LSTM is used to predict lithium battery SOH	MSE is used as a performance metric. Compared with GRU, LSTM, and BPNN, MSE with regularization is 0.45%	To improve the SOH prediction accuracy of lithium batteries
[46] 1 Feb 2021	RoLSTM is a robust adaptive online long short-term memory (LSTM) model proposed for accurately estimating state of charge (SOC)	MAE and RMSE panasonic 18650PF battery at 25°C is 0.40% and 0.79%, respectively	RoLSTM requires a large amount of data which requires more amount of time. It needs to reduce computational complexity
[20] 28 October 2021	SVM is applied as a regression technique to estimate the SOC	RMSE and MSE are utilized as performance metrics, with the RMSE measuring approximately 1.4%	The model is simulated in MATLAB simulink and converted to HDL to implement on FPGA board
[14] March 2020	RNARX is a proposed algorithm for SOC prediction that utilizes RNN principles, nonlinear autoregressive modelling, and exogenous inputs	Results: RMSE 0.4174~0.5637% MAE 0.3164~0.3847%	An optimized ML technique for SOC evaluation is developed to enhance the accuracy of SOC estimation in the context of sustainable EVs
[41] 23 April 2019	Proposed stacked LSTM network with multiple hidden layers to estimate SOC	Results: RMSE 2% and MAE within 1%	The average evaluation time is submillisecond suitable for real-time applications of BMS
[21] Aug 2018	The constructed networks utilized TensorFlow's primary LSTM cell, composed of LSTM units, as their core component	The LSTM-RNN model trained on data from a panasonic 18650PF cell achieved an MAE of 0.6%, RMS of 0.7%, and MAX of 2.6% at 25°C	LSTM-RNN can be trained on various types of battery cells for SOC estimation
The proposed model in this article	In summary, a developed LSTM and BiLSTM model with 128 units in each hidden layer utilizing adaptive gradient descent optimization and early stopping	Results are shown in Table 10, which are superior to other models	The LSTM models are implemented on an FPGA PYNQ Z2 board with a limited dataset to improve computational complexity

TABLE 15: CPU performance with varying learning rate (# of LSTM 128 units and # of cycles 1000).

Learning rate	CPU time in sec	Train		Test	
		MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
0.5	263.005	0.0094	0.0116	0.0016	0.0020
0.1	335.856	0.0094	0.0117	0.0020	0.0023
0.01	354.533	0.0095	0.0118	0.0019	0.0023
0.001	344.493	0.0084	0.0101	0.0059	0.0061
0.0001	444.230	0.0076	0.0084	0.0171	0.0172

TABLE 16: CPU performance with learning rate = 0.01 and # of LSTM 128 units.

Epochs (with ES)	CPU time in sec	Train		Test	
		MAE (%)	RMSE (%)	MAE (%)	RMSE (%)
100	32.028	0.0083	0.0100	0.0062	0.0064
500	163.56	0.0094	0.0117	0.0022	0.0026
1000	330.36	0.0095	0.0118	0.0020	0.0024

TABLE 17: System specification.

CPU	GPU	PYNQ z2
Intel(®) core (™) i7-10750H CPU @2.60 GHz, RAM 16.0 GB	Nvidia GeForce GTX 1660 Ti GPU clock speed: 1770 MHzGraphics RAM size: 6 GB	650 MHz Arm® Cortex®-A9 dual-core processor 512 MB DDR3 with 16 bit bus @1050 Mbps

TABLE 18: LSTM model constant learning rate 0.5 and # 128 LSTM units.

Number of training cycles	Time in seconds			RAM usage in %		
	CPU	GPU	PYNQ z2	CPU	GPU	PYNQ z2
100	35.73	27.72	878.87 s	75.9	81.2	PYNQ Z2
500	190.97	120.14	4466.5 s	77.6	80.8	does not support
1000	243.94	246.85	2 h 25 min 44 s	79.6	80.1	commands in Jupyter
1500	381.26	374.37	3 h 40 min 12 s	79.6	76.6	Notebook to get RAM
2500	628.63	591.06	6 h 7 min 26 s	78.8	77.0	resource utilization

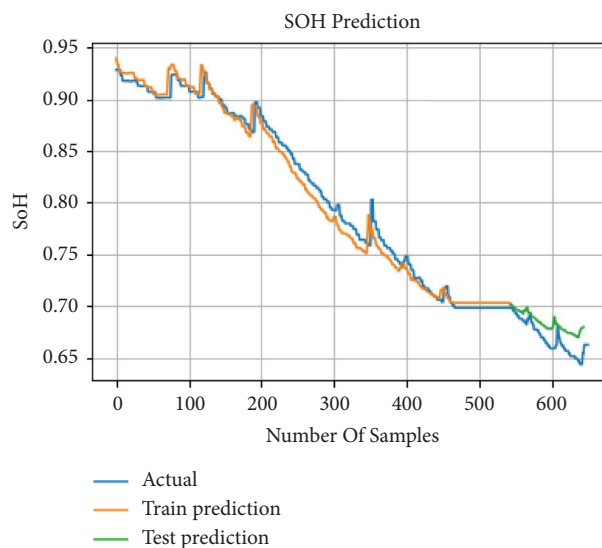


FIGURE 18: SOH vs. number of data samples.

5. Conclusion

Prediction of lithium battery's SOC and SOH is crucial for the enhanced performance of the BMS of EVs. The proposed AI-based SOC and SOH prediction is validated through experiments and testing using diverse EV drive cycles at room temperature, showcasing its adaptability and generalization capability. Our experimental results indicate that the performance of the LSTM model is influenced by the selection of the gradient descent optimization algorithm with an adaptive learning rate. This BiLSTM can predict the SOC of the 18650PF lithium battery cell with MAE smaller than for training and 0.012% and 0.016% for testing. Similarly, using the Adam optimization algorithm, RMSE for training and testing is 0.326% and 0.454% over a 25 C dataset. BiLSTM with an adaptive learning rate can improve performance. As we increased the look back number, the model performance degraded. We must improve the performance LSTM/BiLSTM model with the long look back number. The proposed AI-based approach for estimating state variables of lithium batteries offers advantages such as low cost, low power consumption, high speed, reprogrammable logic, and ample on-chip memory storage, making it a superior choice over existing techniques. To improve the overall performance of the LSTM model, we need to implement the LSTM model using HDLs or C/C++. The findings of this study will aid in developing an efficient algorithm for estimating state variables of lithium batteries such as SOC and SOH, thereby contributing to the future of E-Mobility.

Data Availability

The data supporting this study are from previously reported studies and datasets, which have been cited. We have used third-party open-source data for experimental purpose. Our experimental source code is available from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors have contributed to determining the approach, methodologies, structuring, and revising the paper.

Acknowledgments

The authors express their gratitude to the E&TC Department at Pimpri Chinchwad College of Engineering, Pune, for their support in publishing this work in this journal.

References

- [1] R. Xiong, Y. Zhang, J. Wang, H. He, S. Peng, and M. Pecht, "Lithium-ion battery health prognosis based on a real battery management system used in electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4110–4121, 2019.
- [2] IEA, *Energy and Air Pollution*, <https://www.iea.org/reports/energy-and-air-pollution>, IEA, Paris, 2016, <https://www.iea.org/reports/energy-and-air-pollution>.
- [3] D. Ouyang, M. Chen, J. Liu, R. Wei, J. Weng, and J. Wang, "Investigation of a commercial lithium-ion battery under overcharge/over-discharge failure conditions," *RSC Advances*, vol. 8, pp. 33414–33424, 2018.
- [4] Bureau of Energy Efficiency, "Bureau of energy efficiency," 2022, <https://beeindia.gov.in/content/e-mobility>.
- [5] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring lithium-ion batteries in electric and hybrid vehicles," *Journal of Power Sources*, vol. 258, pp. 321–339, 2014.
- [6] M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: challenges and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 834–854, 2017.
- [7] B. Yang, J. Wang, P. Cao et al., "Classification, summarization and perspectives on the state-of-charge estimation of lithium-ion batteries used in electric vehicles: a critical comprehensive survey," *Journal of Energy Storage*, vol. 39, Article ID 10257, 2021.
- [8] S. Yan, "Understanding lstm and its diagrams," 2022, <https://medium.com/mlreview/>.
- [9] R. L. Abduljabbar, H. Dia, P. W. Tsai, and S. Liyanage, "Short-term traffic forecasting: an LSTM network for spatial-temporal speed prediction," *Future Transp*, vol. 1, no. 1, pp. 21–37, 2021.
- [10] A. Putnam, A. M. Caulfield, E. S. Chung et al., "A reconfigurable fabric for accelerating large scale datacenter services," *ACM SIGARCH - Computer Architecture News*, vol. 42, no. 3, pp. 13–24, 2014.
- [11] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery management system: an overview of its application in the smart grid and electric vehicles," *in IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 4–16, 2013.
- [12] R. S. Kumar, U. R. Geethashree, P. Kirthi, and B. C. Ranjeeta, "Battery and its management for E-rickshaw," in *ICDSMLA 2020. Lecture Notes in Electrical Engineering*, A. Kumar, S. Senatore, and V. K. Gunjan, Eds., Springer, Singapore, 2022.
- [13] K. V. Raj, K. Rayudu, and G. Battapothula, "Critical Review on Battery Management Systems," in *Proceedings of the 2022 International Conference on Applied Artificial Intelligence and Computing (ICAIC)*, Salem, India, May 2022.
- [14] M. A. Hannan, "Toward enhanced state of charge estimation of lithium-ion batteries using optimized machine learning techniques," scientific reports," *Nature*, vol. 10, no. 4687, p. 2020.
- [15] M. S. Hossain Lipu, M. A. Hannan, A. Hussain et al., "How, Data-driven state of charge estimation of lithium-ion batteries: algorithms, implementation factors, limitations and future trends," *Journal of Cleaner Production*, vol. 277, Article ID 124110, 2020.
- [16] Y. Cai, Q. Wang, and W. Qi, "D-ukf based state of health estimation for 18650 type lithium battery," in *Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation*, pp. 754–758, Harbin, China, August 2016.
- [17] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural networks approach to a speaker and language recognition,"

- IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.
- [18] E. Almaita, S. Alshkoor, E. Abdelsalam, and F. Almomani, “State of charge estimation for a group of lithium-ion batteries using long short-term memory neural network,” *Journal of Energy Storage*, vol. 52, Article ID 104761, 2022.
- [19] H. Sun, J. Sun, K. Zhao, L. Wang, and K. Wang, “Data-driven ICA-Bi-LSTM-Combined lithium battery SOH estimation,” *Mathematical Problems in Engineering*, vol. 2022, Article ID 9645892, 8 pages, 2022.
- [20] M. Stighezza, V. Bianchi, and I. De Munari, “FPGA implementation of an ant colony optimization based SVM algorithm for state of charge estimation in Li-ion batteries,” *Energies*, vol. 14, p. 7064, 2021.
- [21] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, “Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2018.
- [22] J. Kim, “Data-driven state of health estimation of Li-ion batteries with RPT-reduced experimental data,” *IEEE Access*, vol. 7, pp. 106987–106997, 2019.
- [23] S. Luciani, S. Feraco, A. Bonfitto, and A. Tonoli, “Hardware-in-the-Loop assessment of a data-driven state of charge estimation method for lithium-ion batteries in hybrid vehicles,” *Electronics*, vol. 10, p. 2828, 2021.
- [24] P. Khumprom and N. Yodo, “A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm,” *Energies*, vol. 12, p. 660, 2019.
- [25] C. Li, F. Xiao, and Y. Fan, “An approach to state of charge estimation of lithium-ion batteries based on recurrent neural networks with gated recurrent unit,” *Energies*, vol. 12, p. 1592, 2019.
- [26] S. Jemmali, B. Manai, and M. Hamouda, “Pure hardware design and implementation on FPGA of an EKF based accelerated SoC estimator for a lithium-ion battery in electric vehicles,” *IET Power Electronics*, vol. 15, pp. 1004–1015, 2022.
- [27] D. He and J. He, “An FPGA-Based LSTM Acceleration Engine for Deep Learning Frameworks,” 2021, <https://www.mdpi.com/2079-9292/10/6/681>.
- [28] J. Bobulski and M. Kubanek, “Deep learning for plastic waste classification system,” *Appl. Comput. Intell. Soft Comput.*-vol. 2021, Article ID 6626948, 12 pages, 2021.
- [29] GPU vs. CPU performance in deep learning models, “GPU vs. CPU performance in deep learning models,” 2022, <https://deci.ai/blog/close-gap-cpu-performance-gpu-deep-learning-models/>.
- [30] C. Ashish, “Why gpu can process image much faster than cpu?,” 2021, <https://www.e2enetworks.com/blog/why-gpu-can-process-image-much-faster-than-cpu>.
- [31] W. Alexander, *Chapter 8 - Digital Signal Processing Systems Design Digital Signal Processing*, W. Alexander and C. Williams, Eds., Academic Press, Boston, 2017.
- [32] X. Tian, B. Jeppesen, T. Ikushima, F. Baronti, and R. Morello, “Accelerating state-of-charge estimation in FPGA-based battery management systems,” in *Proceedings of the 6th Hybrid Electr. Vehicles Conferences. (HEVC)*, London, UK, November 2016.
- [33] M. Hamayel and A. Owda, “A novel cryptocurrency price prediction model using GRU, LSTM and bi-LSTM machine learning algorithms,” *AIDS*, vol. 2, pp. 477–496, 2021.
- [34] LSTM, “The architecture of LSTM,” 2021, <https://www.intechopen.com/chapters/75265>.
- [35] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Networks*, vol. 18, pp. 602–610, 2015.
- [36] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [37] PYNQ-Z2, “PYNQ-Z2 setup guide,” 2022, https://pynq.readthedocs.io/en/v3.0.0/getting_started/pynq_z2_setup.html.
- [38] P. Kollmeyer, “Panasonic 18650PF Li-ion battery data,” *Mendeley Data*, vol. 1, 2018.
- [39] NASA Prognostic Center, “Experiments on Li-ion batteries,” 2022, <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>.
- [40] R. Zhang, B. Xia, B. Li et al., “A study on the open circuit voltage and state of charge characterization of high capacity lithium-ion battery under different temperature,” *Energies*, vol. 11, p. 2408, 2018.
- [41] F. Yang, X. Song, F. Xu, and K.-L. Tsui, “State-of-charge estimation of lithium-ion batteries via long short-term memory network,” *IEEE Access*, vol. 7, pp. 53792–53799, 2019.
- [42] Z. Deng, X. Lin, J. Cai, and X. Hu, “Battery health estimation with degradation pattern recognition and transfer learning,” *Journal of Power Sources*, vol. 525, Article ID 231027, 2022.
- [43] Early stopping, https://keras.io/api/callbacks/early_stopping/, 2022.
- [44] “A simple way to prevent neural networks from overfitting nitish srivastava,” *Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [45] J. Chen, Y. Zhang, J. Wu, W. Cheng, and Q. Zhu, “SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output,” *Energy*, vol. 262, Article ID 125375, 2023.
- [46] G. Javid, D. Ould Abdeslam, and M. Basset, “Adaptive online state of charge estimation of EVs lithium-ion batteries with deep recurrent neural networks,” *Energies*, vol. 14, pp. 1–14, 2021.
- [47] B. Pang, L. Chen, and Z. Dong, “Data-driven degradation modeling and SOH prediction of Li-ion batteries,” *Energies*, vol. 15, p. 5580, 2022.