



Deep-learning framework for state of charge (SoC) estimation of electric vehicle batteries using a Pynq board

V. S. Vijaya Krishna V.^{a*} • N. Pappa^b • S. P. Joy Vasantha Rani^c

^aDepartment of Electronics and Communication Engineering, Presidency University, Bangalore, India

^bDepartment of Instrumentation Engineering, Anna University, Chennai, India

^cDepartment of Electronics Engineering, Anna University, Chennai, India

Received 05 24 2024; accepted 08 18 2024

Available 12 31 2024

Abstract: In recent years, carbon emissions have increased worldwide due to the excessive use of fossil fuels. To overcome these emissions, lithium-ion (Li-ion) batteries have become a more prominent alternative. Li-ion batteries are used as the primary component of energy storage systems for sustainable energy in response to rising global carbon gases. The battery management system (BMS) in electric vehicles (EVs) is an important aspect and is indicated by two parameters called state of charge (SoC) and state of health (SoH). Of these two, the SoC value is related to energy distribution, charging and discharging of batteries. Hence, estimating the SoC value is of high importance in BMS for the optimum use of batteries. Recent trends in artificial intelligence and deep learning provides a way for new developments in algorithms for estimating the SoC. At the same time, the use of programmable devices like field programmable gate arrays (FPGAs) for data processing applications provides acceleration in time and an optimal use of hardware. Pynq boards which are Zynq dependent, and one variant of FPGAs, are capable of executing python programs directly on hardware. This paper focuses on developing different DNN architectures for estimating the SoC of a Li-ion battery of an EV with a Pynq Z2 board. From the results, it can be concluded that a hybrid model when using a PYNQ board has less MAE and RMSE with 0.52% and 0.63%, and execution time with 0.28 ms which are optimum values when compared with other models.

Keywords: Electric vehicle, state of charge (SoC), deep learning, battery management system, Pynq

*Corresponding author.

E-mail address: v.vijaykrishna@gmail.com (V. S. Vijaya Krishna V.).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

It has become a global consensus to reduce and control the production of fossil fuel-based fuel vehicles, making the accelerated transformation of vehicle electrification an important trend in the current development (Hassan et al., 2022). Therefore, the requirement for clean energy development has become more important. As a result, battery performance and energy management requirements (He et al., 2021) have drawn more attention. A battery management system (BMS) is required as a primary component in all battery applications with numerous cells to ensure a dependable operation throughout consumption in electric vehicles (EVs). The majority of electric vehicles' current power source is lithium batteries.

The state of charge (SoC) and state of health (SoH) are the two most important functions of the BMS for the assessment of battery states. These functions allow users to assess the battery pack's remaining charge, forecast its level of aging, and determine how much power it can deliver at any given time. Since the SoC estimation directly gauges a vehicle's possible mileage and is necessary for the battery balancing system, it is also essential to maintain proper operation of the (Hu et al., 2018) EV drive systems. Direct measurement of SoC is challenging because it is not a directly observable quantity like voltage or current. SoC is an internal state of the battery, representing the remaining charge relative to the battery's capacity, which can only be inferred through indirect methods. Variability in battery characteristics, environmental conditions, and aging further complicate direct measurement, necessitating sophisticated inference techniques to accurately estimate SoC. A careful calibration of the model parameters is necessary and can be achieved by empirical models (Hu et al., 2022), which are frequently used in the BMS or data driven models which particularly depend on battery data. Machine learning and deep-learning approaches, on the other hand, need a lot of data for training and testing the battery condition.

The amount of remaining energy must be indicated by a state parameter in lithium-ion (Li-ion) batteries (Chatterjee et al., 2021). The state of charge (SoC) is therefore suggested. The ratio of the battery's present available capacity to its maximal capacity is known as SoC. In order to calculate the SoC, a precise battery capacity measurement is essential. Clarifying the change in SoC allows for the determination of the battery's anomalous charging capacity, resulting in the safe operation of the electric vehicles (Guo & Peng, 2019). Traditional methods, such as Coulomb counting and model-based approaches, often require precise models and are susceptible to inaccuracies due to parameter variations and measurement noise. Deep-learning approaches, on the other hand, are data-driven and can learn complex relationships

from large datasets, allowing for better handling of nonlinearities and variations in battery behaviour.

Several Artificial Intelligence (AI) techniques for the SoC estimate have been proposed in recent years. These tactics have the potential to perform better than the conventional ones. The techniques make use of a special learning feature of the AI model for training (Ahmed et al., 2021), which may, with the aid of a vast amount of data, link the relationships and patterns between the cell assessment parameters (i.e., voltage, current, resistance, and temperature) and the SoC (Cui et al., 2022). Recently, the field of EVs and associated infrastructure, including EV battery design and management, charging stations, and even the smart grid, have seen industrial and academic applications of deep learning, which is defined as algorithms supporting deep neural networks aimed at mimicking natural thinking, perception, and action.

Deep-learning algorithms are gaining ground as key elements in a wide range of industries, from banking and biotechnology to computer vision and robotics. CNNs are chosen for their ability to capture spatial dependencies in data, which is useful in identifying patterns in the input features relevant to SoC estimation. RNNs are selected for their strength in handling sequential data and capturing temporal dependencies, which is crucial for understanding the time-series nature of battery charge-discharge cycles. Combining these architectures leverages their complementary strengths in processing different aspects of the data. DNNs have only recently achieved their current level of success due to the enormous amount of data that is currently available (Kalapothas et al., 2022), coupled with high computer power and advancements in algorithmic methodologies. In this regard, numerous hardware platforms have been targeted in an effort to increase the energy efficiency and throughput of Deep Neural Networks (DNNs). Field Programmable Gate Arrays (FPGAs) (Messaoud et al., 2022) are being used more frequently for data-intensive applications due to the ability to customise hardware accelerators and obtain high-performance implementations with little power consumption. In embedded systems applications, where the advantages of the re-configurability qualities make the system more durable, capable of facing system failures, and respecting the restrictions of the embedded devices, FPGAs (Saady & Essai, 2022) have also shown themselves to be a competitive alternative to GPUs. System designers can also utilise the Python language and its libraries to create applications for Zynq-7000-based devices developed by Xilinx's open-source project Pynq (Python Productivity for Zynq) (Mándi et al., 2021). The Zynq-7000 is made up of a programmable logic FPGA and a dual-core ARM A9 processor known as the processing system (Krishna 2014). Within the Pynq, Xilinx offers tools for interacting with Hardware modules and downloading complete bit streams

that set up the Zynq device's whole programmable logic. Additionally, Xilinx offers an overlay bit stream that highlights the key aspects of the Hardware and software. Overlays offer their functionality to Python through methods that are built into classes (Saidi et al., 2021).

To estimate the SoC of li-ion battery of EV and explain the neural network with understandable AI, the work in this study aims to provide an optimised deep-learning (DL) approach. The work includes two parts, one is executing deep network architectures for selected dataset of li-ion battery which is used in EVs and next part is to run the same architectures on reconfigurable hardware boards like Pynq board (Huynh, 2021). The first step in the process is to gather and prepare the data for the model training. Then the SoC of the EV battery is estimated using three DL network architectures based on Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). These three models are subsequently trained using training data and tested using test data. An optimal solution can be concluded after comparison of the architectures. Later the same architectures are executed on hardware to compare the speed of execution. Section 2 describes SoC and its characteristics and different ways to estimate it based on data driven concept. Section 3 discusses the various types of deep networks which are used in this paper for estimation. Section 4 describes Pynq platform and has information about hardware and process of execution. Section 5 discusses the results and evaluation of architectures; a comparison is provided followed by conclusions in Section 6.

2. State of charge (SoC) estimation

The SoC is a vital parameter in battery management system of electric vehicles. It will give estimates about charge present in the battery and hence the battery backup can be configured. SoC can be defined as the ratio of present available charge to normal full charge (Zhang et al., 2021). Full charge data can be obtained from the data sheet of the manufacturer. The simple mathematical representation of SoC is given as in Equation 1.

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

Where Q(t) represents the available charge
 Q_n represents the full charge

Another mathematical expression representing SoC(%) in terms of battery capacity is given in Equation 2.

$$SoC(\%) = \frac{C_{Present}}{C_{total}} \times 100 \tag{2}$$

Where $C_{present}$ represents the available battery capacity
 C_{total} represents the full capacity

To estimate the SoC, there are different methods and architectures available (Zhang et al., 2022). Estimation can be done statistically by having model parameters of the battery or by processing the historic data of the battery.

Figure 1 illustrates some of the available methods used for estimating the SoC of an electric vehicle. Filter algorithms like Kalman filter (KF) and other variants of KF can be used for estimation. They rely on data of input parameters like voltage, current, temperature etc. These data are processed using battery model and iterated certain number of times based on availability of number of input samples (Lian et al., 2020). Basic KF can be used for accurate estimation provided the battery model and parameters follow a linear system. Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are nonlinear variants, where the model and parameters need not be linear (Vidal et al., 2020). The work in this paper uses dataset of Li-ion battery which is modelled as a nonlinear system. Though KF algorithms are highly resistant to interferences in systems and highly robust, it requires accurate battery model and parameter identification. Even EKF and UKF provides a way for estimation of nonlinear systems, they suffer from computational overload if the system becomes more complex. In summary KF algorithms are based on battery model and data of input parameters and hence statistical information of battery is essential.

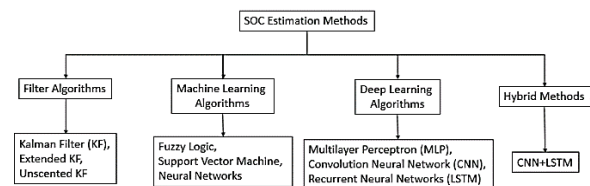


Figure 1. Summary of algorithms used for SoC estimation of Li-ion battery.

Another set of algorithms based on machine learning and deep learning depends solely on battery data battery. Traditional machine learning algorithms like Fuzzy Logic, Support Vector Machine (SVM) (Wu et al., 2021), Neural Networks etc. are used for SoC estimation. SVM can be used to solve logistic/classification problems. Generally, SoC estimation is based on regression method, where the algorithm minimizes error function step by step. This method is used to solve data that is not linearly separable. However, a vital feature of SVM is to employ simplified optimization techniques such as programming with linear constraints of the SVM parameters. Additionally, the feature of error tolerance margin is used in such a way that no cost function penalty is applied to the fitting error, only if it is within some defined error band and thereby it will stabilize the estimation. Fuzzy logic is based on fuzzy techniques which operate on information processing where classical set theory and binary

logic becomes hard to employ. The main feature of fuzzy logic is the use of symbolic knowledge representation in a fuzzy-based (if, then) rule. Neural networks deal with classifying large-scale networks into various categories and identifying them with their fine structures (How et al., 2020).

These algorithms require high computing process and hence used selectively based on the battery data and input parameters. A solution for the above problem can be handled by deep-learning algorithms. Deep learning can generate new features from a limited dataset used for training. The accuracy depends on how the network is trained and the type of dataset used for training. These algorithms can be used with unstructured data and hence the appropriate data format can be obtained (Pisal & Vidyarthi, 2022). Detailed descriptions of these algorithms are presented in the next section. Traditional machine learning algorithms and deep networks can be combined to generate a hybrid model, where the accuracy of the estimation increases. The advantage of these algorithms is that it does not require the full knowledge of the battery model and its design. The flipside of these algorithms is related to the accuracy of estimation using these models which is dependent on the amount of data provided and the way the useful data is extracted from the available bulk data. The processing time is also a matter of concern with data.

3. Deep neural network (DNN) framework

This section describes the steps in choosing the DNN architecture, data set and the process to estimate the SoC of battery in EVs.

3.1. Process flow

Process flow describes the various steps performed for estimation. Figure 2 shows the steps used in deep learning for estimation of SoC of Li-ion battery used in EV. Initially the data required for the estimation must be acquired. Data like temperature, voltage, current, among others, are acquired. The acquired data has redundant information, and entire data is not necessary for the further process. Hence data has to be processed and the required data for training and testing has to be generated.

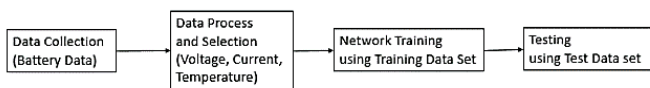


Figure 2. Steps in estimating the SoC of a Li-ion battery using the deep-learning framework.

The training data set is used for training the network related to estimation of SoC of battery. The data set of battery is used to verify whether the parameters of the training data set are

reasonable to change the model parameters, and the test set is used to test the functional capability of the trained network and thus verification can be done. The trained network is trained with chosen neural network with the training data set, and the model trained by the training data set is verified in the validation data set to see that if the accuracy is attaining maximum value (Shrivastava et al., 2021). If the required accuracy is not reached, then the parameters of the neural network are to be adjusted, and then the network can be trained again. After completing the training the network, testing the network is carried out. This step is used for evaluation of the network for the required estimation application. Evaluation can be done by comparing the estimated values with the actual values of the concerned parameter. Here, two parameters are considered for comparison; one is the mean absolute error (MSE), and the other one is the root mean square error (RMSE). The equations for MAE and RMSE are given in Equations 3 and 4.

$$AE = \frac{1}{N} \sum_{K=1}^N (SoC_{est} - SoC_{act}) \tag{3}$$

$$MSE = \sqrt{\frac{1}{N} \sum_{K=1}^N (SoC_{est} - SoC_{act})^2} \tag{4}$$

Where SoC_{est} is the estimated value
 SoC_{act} is the actual value
 N is the number of variables

The smaller the error, the higher the accuracy of the estimation.

3.2. DataSet

The Li-ion battery data is the most vital parameter of the entire process flow of SoC estimation because, high-quality Li-ion battery data (Kollmeyer 2018) can give clear information about the relationship among the lithium battery electrochemistry parameters such as conditions and design. Since various types of lithium batteries have different parameter attributes based on the life cycle of lithium batteries, failure condition, complexity of design, life cycle of batteries, the battery condition becomes a prominent condition for the battery management system. Hence, more testing institutes have disclosed test data of Li-ion batteries (dos Reis et al., 2021). In this paper the data required for the estimation are obtained from the Panasonic 18650PF Li-ion battery dataset. This dataset is used for lithium battery SoC estimation, lithium battery performance measurement, and dynamic characteristic analysis of the battery operation. Example of dataset used is Panasonic 18650PF Li-ion battery dataset with temperature at 25 °C and Data obtained at 25-amp, 18-volt firing circuit universal battery tester channel in an 8-cubic-foot

thermal chamber. The battery, charged after each test at a 1 C rate to 4.2 V, 50 mA cut off, with different battery temperatures.

3.3. DNN architectures

The SoC estimation of a Li-ion battery in the deep-learning framework uses deep-learning algorithms based on input data (voltage, current, temperature, power, capacity, etc.) and output data (SoC) from the dataset described in previous sections. There are different kinds of deep networks in the literature. In this paper the work is based on three architectures which are single structure DNN with hidden layers, convolution neural networks (CNN), long short-term memories (LSTM) and hybrid network which means combination of single structure and CNN.

3.3.1. Single structure DNN with hidden layers

Single structure forms a part of Multi-Layer Perceptron (MLP). When compared to traditional machine learning algorithms, these MLPs have improved arithmetic complexity and training parameters. Main advantage of this model is that it does not limit the dimensionality of the input, highly adaptable to the data, and theoretically, a simple network with a smaller number of hidden layers can fit any nonlinear function (Shahriar et al., 2022). The main disadvantage is it suffers from easy over fitting when inputs are increased beyond certain limit. Figure 3 displays the structure of a deep neural network with three hidden layers and each hidden layer contains eight neurons. The inputs are taken from voltage, current and temperature.

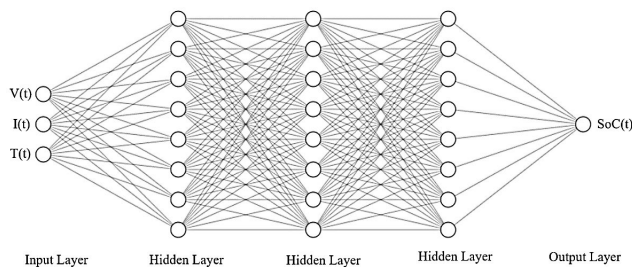


Figure 3. Deep neural network architecture for estimating the SoC of a Li-ion battery.

In this paper for estimation of SoC, a Single structure DNN with 3 hidden layers and one input layer with three input variables and one output layer with one output variable (SoC) is used. Activation functions like hyperbolic tangent, sigmoid (tansig) and rectified liner unit (ReLu) are used. He and Xavier weight initialization techniques are used. There should be a limit in the number of hidden layers as more layers may result in over fitting problem.

3.3.2. Convolution neural network

Convolution type deep networks for SoC estimation of Li-ion batteries are mainly dependent on 1-dimensional convolutional neural networks (1D-CNN). The major benefit of a 1-dimensional convolutional neural network is that it can extract and separate one-dimensional data from the available different kinds of dataset and with less computational capacity. These CNNs are used in defect prediction, image segregation, computer vision, etc.

The SoC estimation of a Li-ion battery is considered a regression problem, but networks in 1D-CNNs are not as accurate in terms of regression estimation problems as in classification problems. Hence, they are configured as a feature extraction layer in combination with other networks. Figure 4 shows a typical CNN for estimation of SoC of battery in EV. The convolution kernel and pooling layer provides an estimate which is given to the fully connected deep network. This is done as one dimension CNN cannot handle the regression concept effectively. Hence, a single structure is used at the end. This conjunction of two networks increases accuracy (Tian et al., 2021).

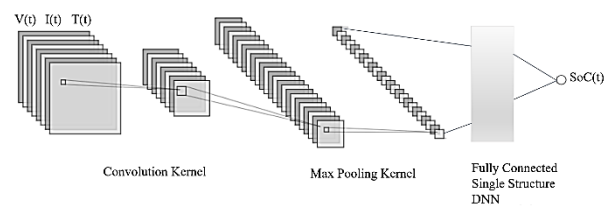


Figure 4. 1D-CNN architecture for estimating the SoC of a Li-ion battery.

3.3.3. LSTM

Long short-term memory (LSTM) is an improved version of recurrent neural network (RNN) which uses temporal information of available input data. It is configured to design contextual information of chronological sequence with feedback loops (Zhang et al., 2016). Figure 5 shows the general architecture of LSTM, where the memory cells are represented with feedback loop. It contains cells that feed the network from previous time stamp to change the estimations at the present time stamp. Each cell contributes for the updating of the data and hence the entire network will be updated at given time stamp. In this way the past data can be transmitted recursively in the complete loop of LSTM. With these features, this algorithm can effectively acquire complicated features within time series with respect to both long and short term (Tian et al., 2022). This algorithm can effectively process the time-series learning problem, but it has a gradient vanishing or explosion problem.

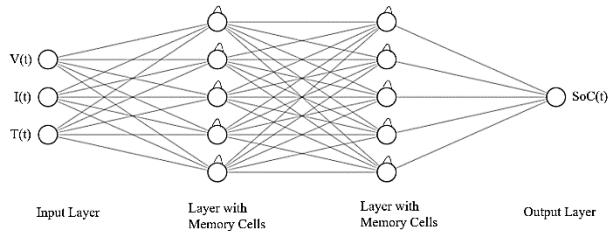


Figure 5. Recurrent LSTM architecture for estimating the SoC of a Li-ion battery.

3.3.4. Hybrid method

The main concept of the hybrid deep neural network for estimating the SoC of a li-ion battery of EV is to improve the estimated accuracy of the network by combining the features of different kinds of neural networks. Figure 6 illustrates a typical hybrid architecture for li-ion battery SoC estimation problem where a 1D-CNN is used as a feature extraction layer to extract deep components of the input data, and a recurrent neural network (LSTM) is used as a network building layer to design a network between the SoC and the input variables. Accuracy can vary even if fully connected layers can be connected before the final output layer to improve the accuracy of the network. Data segregation is done by 1D-CNN and network processing is done by LSTM.



Figure 6. Hybrid method using a combination of 1D-CNN and LSTM architectures for estimating the SoC of Li-ion battery blocks.

4. PYNQ (Python Productivity for Zynq) platform

The architectures realized in previous sections are edited on Jupyter Notebook. The same is tried to execute on the portable hardware called Pynq board. The main advantage of using a special hardware board is its parallel execution and usage of PYNQ overlays and easy interface the board with Jupyter Notebook (Corradi, 2018). This section describes the hardware characteristics and how it can be modelled for the battery estimation.

Pynq can be represented as a platform that is built on Zynq of Xilinx. System on Chip technology is employed in Zynq, where the processor and programmable logic (PL) can be embedded on a single chip. PL can be used to realize functions which can execute in parallel and hence execution time can be decreased. The processor along with PL can be considered a heterogeneous hardware where the processor can be used for complex calculations (Krishna et al., 2018). The combined usage of the processor and the PL can achieve optimised execution time and hardware resources. Pynq board is one special category which is based on Xilinx Zynq 7000 system on

chip which has dual core ARM cortex – A9 processor and Field programmable Gate Array (FPGA) PL along with Python libraries. The Pynq platform provides a flexible way to execute the Python program directly on processor and design stored in PL. The Intellectual property (IP) core design in PL can be used in other applications along with the processor capacity. Initially Pynq-Z1 board is available and later Z2 board with more features is released. Figure 7 displays the simplified block level structure of Zynq system on chip used in Pynq board. Processing system (PS) and Programmable logic (PL) (Huynh, 2021) are main blocks for storing the design and executing. An advanced extensible interface (AXI) is a special type of interconnection between PS and PL. This board has provision to boot the processor from a micro-SD (Scan Disk) card which contains PYNQ Z2 boot image.

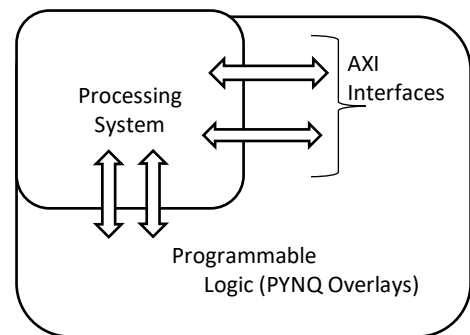


Figure 7. Simplified model of Zynq system on chip on a PYNQ board.

This board has PMOD (peripheral module interface) ports, on board I/O, HDMI, USB and Ethernet connectivity. Because of these features, this board can be used for applications based on system on chip like image processing, industrial control, Internet of Things (IoT), embedded processing, etc. Pynq overlays are customized functions required for the particular application written on PL which can be used with Python packages. AXI interfaces can be various types based on the data type like sample data, streamed data (Wang et al., 2018).

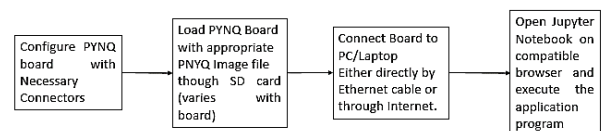


Figure 8. Steps involved in executing DNN architectures for SoC estimation on a PYNQ board.

Figure 8 shows the process flow of executing the estimation algorithm on this board. Initially the PYNQ board must be configured as per the requirement. It includes change of jumper pins, universal serial bus (USB) port settings, Ethernet wire connections. Once the board settings are done then it must be booted with Pynq image through SD card. This SD card

is preloaded with an image file from an authorized Pynq website or that is available with the board accessories (Yamamoto & Kawahara, 2021). Power to the board can be done from the micro USB connected from a PC/laptop. The next step is to connect the board to the PC/laptop. This can be done in two ways: one way is direct connection using an Ethernet cable. This cable can be connected to an Ethernet port of board to the Ethernet port of PC. The second way is to connect both the board and PC to the Internet, available both through the same router network. While connecting with the first method, the port address (Internet Protocol version 4/ IPv4) should be changed as per the board parameters. This can be done through the Internet properties settings tab. Once the board is connected and turned on, the board will boot from the SD card and all Python related libraries and Zynq system on chip will be ready to use. Now, with the appropriate browser, open Jupyter Notebook from the Pynq board with the following address:

192.168.2.99:9090

Now Jupyter Notebook along with Pynq overlays can be accessed. The application program can be written and executed. The associated libraries and packages can be accessed through the folders present in the system. The present application related to SoC estimation is written using Python on this Jupyter Notebook and functions related to estimation are written on PL of Zynq system on chip.

5. Results and evaluation

The algorithms/deep networks discussed in previous sections are used to estimate the SoC of Li-ion battery of EV. Initially the algorithms are edited in Jupyter Notebook using Python and associated libraries are simulated. Later the same algorithms are executed on the Pynq board and tabulated the values.

Different functions are used to get the values of MAE and RMSE. In addition, the execution speed is also monitored for performance evaluation. Voltage, Current, Temperature are considered as inputs and SoC is considered as output. Panasonic 18650PF Li-ion battery dataset is used for estimation. After execution, the values are tabulated as per the table given below. Table 1 gives the values of MAE, RMSE and Execution time executed on PYNQ board.

Table 1 represents the error values of SoC using different architectures obtained through software simulation. From the table, it is evident that hybrid architecture provides lesser error at expense of execution time.

From Table 2, it can be concluded that DNN can take Li-ion battery data without taking the dimensions of input variables into account. However, when it employs many layers, it is simple to exacerbate the issue of overfitting and local optimum in SoC calculation. When CNN is solely utilized in the 1D-CNN structure, it has a lesser precision of SoC estimate than other neural network structures, but it can still effectively extract the data features of Li-ion battery data. Although LSTM provides better precision it has several calculation parameters for SoC estimation and requires a large storage capacity to process Li-ion battery data, it has a long training time. Nevertheless, it can process long-term Li-ion battery data for SoC estimation, and it mitigates the problem of gradient disappearance and explosion. When the right neural network parameters are used, CNN + LSTM can further improve the accuracy of SoC estimation. Due to its relatively complex model structure, which is more intricate than a simple neural network's single structure, a longer execution time is typically observed.

Figures 9 to 11 show the comparison of SoC parameters like error and speed with different SoC architecture models which are realized using PYNQ Platform.

Table 1. Summary table of SoC performance with different DNN architectures obtained using software simulation

Model	Input variables	SoC Error using software simulation		
		MAE	RMSE	Execution
		(%)	(%)	Time (mS)
DNN	V(t), I(t), T(t)	0.64	0.795	0.22
CNN	V(t), I(t), T(t)	0.64	0.795	0.22
LSTM	V(t), I(t), T(t)	0.61	0.72	0,26
CNN+	V(t), I(t), T(t)	0.53	0.65	0.284
LSTM				

Table 2. Summary table of SoC performance with different DNN architectures done using a PYNQ platform.

Model	Input variables	SoC Error using software simulation		
		MAE	RMSE	Execution
		(%)	(%)	Time (mS)
DNN	V(t), I(t), T(t)	0.62	0.79	0.2
CNN	V(t), I(t), T(t)	0.62	0.79	0.2
LSTM	V(t), I(t), T(t)	0.6	0.7	0,25
CNN+	V(t), I(t), T(t)	0.52	0.63	0.28
LSTM				

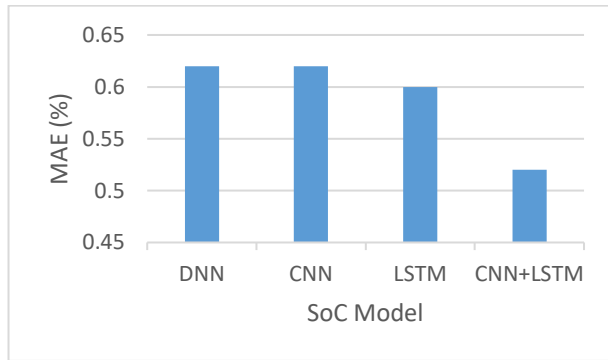


Figure 9. Graph representing the SoC MAE error with different SoC models.

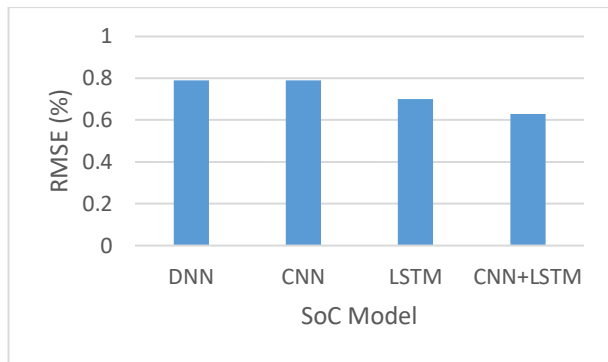


Figure 10. Graph representing the SoC RMSE error with different SoC models.

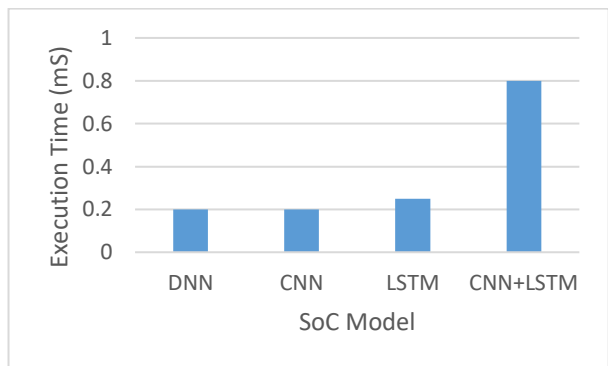


Figure 11. Graph representing the execution time with different SoC models.

Figure 12 shows the Experimental setup with Pynq board along with laptop used for estimation of SoC of Li-ion battery in EV. In the hardware setup Pynq board is connected to laptop using USB mode and for executing code an Ethernet connection is required. Jupyter Notebook can be accessed only through this connection. All the settings are made as per the specifications of the board.

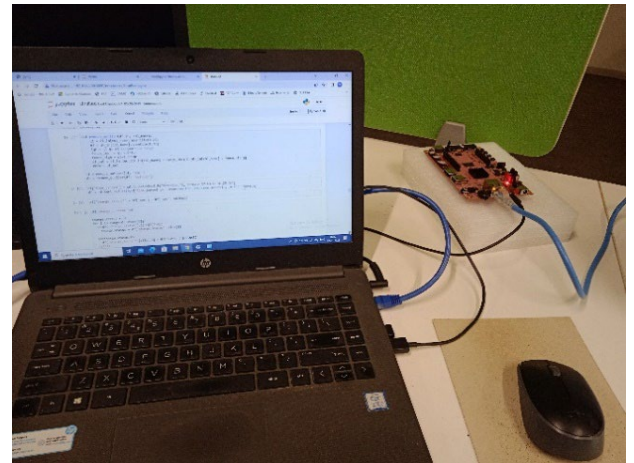


Figure 12. PYNQ board used for estimating the SoC of a Li-ion battery of an electric vehicle.

6. Conclusions

In this paper different deep architectures are used to estimate the SoC of a Li-ion battery of an electric vehicle. The widespread use of li-ion batteries of EV can support energy efficiency, the reduction of carbon dioxide emissions, and environmental protection, helping to achieve carbon neutrality and peak emission goals. Different Deep architectures like DNN, CNN, LSTM, hybrid methods are used to estimate the SoC. Experimental results indicate that different architectures when executed on PYNQ board have optimal values in terms of accuracy and speed. Combination of different architectures or hybrid architectures can provide higher accuracy with more execution time. By providing more accurate and reliable SoC estimates, the framework can improve the overall performance and reliability of EVs. This can enhance user confidence in EV range and battery longevity, promoting greater adoption. Additionally, better SoC estimation can lead to more efficient energy management and longer battery life, reducing operational costs and environmental impact. From the results it is evident that hybrid model which is the combination of CNN and LSTM when realized on a PYNQ board has less MAE and RMSE with 0.52% and 0.63% and execution time with 0.28 ms which are optimum values when compared with other models.

In future, integrating additional data sources, such as environmental factors or vehicle usage patterns can be used to enhance model robustness. Developing more lightweight models to reduce computational demands and facilitate deployment for even more constrained hardware.

Conflict of interest

The authors have no conflict of interest to declare.

Funding

The authors received no specific funding for this work.

References

- Ahmed, M., Y. Zheng, A. Amine, H. Fathiannasab, & Z. Chen. (2021). The role of artificial intelligence in the mass adoption of electric vehicles. *Joule* 5(9): 2296-2322. <https://doi.org/10.1016/j.joule.2021.07.012>
- Chatterjee, P., Singh, J., Singh, R., Avadh, Y. A. R., & Kanchan, S. (2021). Electric Vehicle Modeling in MATLAB and Simulink with SoC & SoE Estimation of a Lithium-ion Battery. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1116, No. 1, p. 012103). IOP Publishing. <https://doi.org/10.1088/1757-899X/1116/1/012103>
- Corradi, G. (2018). *The value of python productivity: Extreme edge analytics on xilinx zynq portfolio. PYNQ Framework and the Zynq Portfolio. Xilinx.*
- Cui, Z., Dai, J., Sun, J., Li, D., Wang, L., & Wang, K. (2022). Hybrid methods using neural network and Kalman filter for the state of charge estimation of lithium-ion battery. *Mathematical Problems in Engineering*, 2022(1), 9616124. <https://doi.org/10.1155/2022/9616124>
- Guo, J., H. He, & J. Peng. (2019). Real-time energy management for plug-in hybrid electric vehicle based on economy driving pro system. *Energy Procedia*, 158, 2689-2694. <https://doi.org/10.1016/j.egypro.2019.02.023>
- Hassan, M. U., Saha, S., Haque, M. E., Islam, S., Mahmud, A., & Mendis, N. (2022). A comprehensive review of battery state of charge estimation techniques. *Sustainable Energy Technologies and Assessments*, 54, 102801. <https://doi.org/10.1016/j.seta.2022.102801>
- He, H., Wang, Y., Li, J., Dou, J., Lian, R., & Li, Y. (2021). An improved energy management strategy for hybrid electric vehicles integrating multistates of vehicle-traffic information. *IEEE Transactions on Transportation Electrification*, 7(3), 1161-1172. <https://doi.org/10.1109/TTE.2021.3054896>
- How, D. N., Hannan, M. A., Lipu, M. S. H., Sahari, K. S., Ker, P. J., & Muttaqi, K. M. (2020). State-of-charge estimation of li-ion battery in electric vehicles: A deep neural network approach. *IEEE Transactions on Industry Applications*, 56(5), 5565-5574. <https://doi.org/10.1109/TIA.2020.3004294>
- Hu, Y., Li, W., Xu, K., Zahid, T., Qin, F., & Li, C. (2018). Energy management strategy for a hybrid electric vehicle based on deep reinforcement learning. *Applied Sciences*, 8(2), 187. <https://doi.org/10.3390/app8020187>
- Hu, C., Li, B., Ma, L., & Cheng, F. (2022). State-of-charge estimation for lithium-ion batteries of electric vehicle based on sensor random error compensation. *Journal of Energy Storage*, 55, 105537. <https://doi.org/10.1016/j.est.2022.105537>
- Kalapothis, S., Flamis, G., & Kitsos, P. (2022). Efficient edge-AI application deployment for FPGAs. *Information*, 13(6), 279. <https://doi.org/10.3390/info13060279>
- Kollmeyer, P. (2018). Panasonic 18650pf li-ion battery data. *Mendeley Data*, 1(2018), 1-15. <https://data.mendeley.com/datasets/wykht8y7tg/1>
- Krishna, V. V., Pappa, N., & Rani, S. J. V. (2018). Implementation of embedded soft sensor for bioreactor on Zynq processing system. In *2018 International conference on recent trends in electrical, control and communication (RTECC)* (pp. 297-301). IEEE. <https://doi.org/10.1109/RTECC.2018.8625644>
- Krishna, V. V. (2014). Performance Optimization of CNFET-based Domino Logic circuits. *International Journal of Advanced Research in Electronics and Communication Engineering*, 3(9), 975-978.
- Lian, R., Tan, H., Peng, J., Li, Q., & Wu, Y. (2020). Cross-type transfer for deep reinforcement learning based hybrid electric vehicle energy management. *IEEE Transactions on Vehicular Technology*, 69(8), 8367-8380. <https://doi.org/10.1109/TVT.2020.2999263>
- Mándi, Á., Máté, J., Rózsa, D., & Oniga, S. (2021). Hardware accelerated image processing on FPGA based PYNQ-Z2 board. *Carpathian journal of electronic and computer engineering*, 14(1), 20-23.

- Messaoud, S., Bouaafia, S., Maraoui, A., Ammari, A. C., Khriji, L., & Machhout, M. (2022). Deep convolutional neural networks-based Hardware–Software on-chip system for computer vision application. *Computers & Electrical Engineering*, 98, 107671. <https://doi.org/10.1016/j.compeleceng.2021.107671>
- Pisal, P. S., & Vidyarthi, A. (2022). An optimal control for power management in super capacitors/battery of electric vehicles using Deep Neural Network. *Journal of Power Sources*, 542, 231696. <https://doi.org/10.1016/j.jpowsour.2022.231696>
- dos Reis, G., Strange, C., Yadav, M., & Li, S. (2021). *Lithium-Ion Battery Data and Where to Find It. Energy and AI*, 5. <https://doi.org/10.1016/j.egyai.2021.100081>
- Saad, M. M., & Essai, M. H. (2022). Hardware implementation of neural network-based engine model using FPGA. *Alexandria Engineering Journal*, 61(12), 12039-12050. <https://doi.org/10.1016/j.aej.2022.05.035>
- Saidi, A., Othman, S. B., Dhouibi, M., & Saoud, S. B. (2021). FPGA-based implementation of classification techniques: A survey. *Integration*, 81, 280-299. <https://doi.org/10.1016/j.vlsi.2021.08.004>
- Shahriar, S. M., Bhuiyan, E. A., Nahiduzzaman, M., Ahsan, M., & Haider, J. (2022). State of charge estimation for electric vehicle battery management systems using the hybrid recurrent learning approach with explainable artificial intelligence. *Energies*, 15(21), 8003. <https://doi.org/10.3390/en15218003>
- Shrivastava, P., Soon, T. K., Idris, M. Y. B., Mekhilef, S., & Adnan, S. B. R. S. (2021). Lithium-ion battery state of energy estimation using deep neural network and support vector regression. In *2021 IEEE 12th Energy Conversion Congress & Exposition-Asia (ECCE-Asia)* (pp. 2175-2180). IEEE. <https://doi.org/10.1109/ECCE-Asia49820.2021.9479413>
- Tian, J., Xiong, R., Shen, W., & Lu, J. (2021). State-of-charge estimation of LiFePO₄ batteries in electric vehicles: A deep-learning enabled approach. *Applied Energy*, 291, 116812. <https://doi.org/10.1016/j.apenergy.2021.116812>
- Tian, J., Xiong, R., Lu, J., Chen, C., & Shen, W. (2022). Battery state-of-charge estimation amid dynamic usage with physics-informed deep learning. *Energy Storage Materials*, 50, 718-729. <https://doi.org/10.1016/j.ensm.2022.06.007>
- Vidal, C., Malysz, P., Kollmeyer, P., & Emadi, A. (2020). Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art. *IEEE Access*, 8, 52796-52814. <https://doi.org/10.1109/ACCESS.2020.2980961>
- Huynh, T. V. (2022). FPGA-based Acceleration for Convolutional Neural Networks on PYNQ-Z2. *International Journal of Computing and Digital Systems*, 11(1), 441-450.
- Wang, E., Davis, J. J., & Cheung, P. Y. (2018). A PYNQ-based framework for rapid CNN prototyping. In *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)* (pp. 223-223). IEEE. <https://doi.org/10.1109/FCCM.2018.00057>
- Wu, C., Fresse, V., Suffran, B., & Konik, H. (2021). Accelerating DNNs from local to virtualized FPGA in the Cloud: A survey of trends. *Journal of Systems Architecture*, 119, 102257. <https://doi.org/10.1016/j.sysarc.2021.102257>
- Yamamoto, K., & Kawahara, T. (2022). Scalable fully coupled annealing processing system and multi-chip FPGA implementation. *Microprocessors and Microsystems*, 95, 104674. <https://doi.org/10.1016/j.micpro.2022.104674>
- Zhang, J., Jiao, X., & Yang, C. (2021). A double-deep Q-network-based energy management strategy for hybrid electric vehicles under variable driving cycles. *Energy Technology*, 9(2), 2000770. <https://doi.org/10.1002/ente.202000770>
- Zhang, X., Wang, Y., Yang, D., & Chen, Z. (2016). Behavior data of battery and battery pack SOC estimation under different working conditions. *Data in brief*, 9, 737-740. <https://doi.org/10.1016/j.dib.2016.10.012>
- Zhang, D., Zhong, C., Xu, P., & Tian, Y. (2022). Deep learning in the state of charge estimation for li-ion batteries of electric vehicles: A review. *Machines*, 10(10), 912. <https://doi.org/10.3390/machines10100912>