A Comparative Analysis of Various Deep Learning Models for State-of-Charge Estimation in Lithium-Ion Batteries

Sourabh Joshi
Department of Electrical Engineering
Visvesvaraya National Institute of Technology, Nagpur 440010
joshisourabh47@gmail.com

Dr Arghya Mitra
Department of Electrical Engineering
Visvesvaraya National Institute of Technology, Nagpur 440010
mitraarghya@gmail.com

Abstract —**Lithium-ion batteries (LIBs) have emerged as a key technology driving recent advancements in electrical and electronic systems. They are widely used in electric vehicles, renewable energy storage systems, portable electronic devices and even in orbiting satellites due to their high energy density, fast charging capabilities, long operational lifespan, and minimal maintenance requirements. Accurate estimation of the State-of-Charge (SoC) is obligatory to ensure the safe, reliable, and efficient performance of LIBs. Conventional approaches like Coulomb counting and Kalman filters are commonly used, but they often face issues in handling nonlinear battery behaviour, high-dimensional data, and complex operational conditions. In this study, we perform a comparative analysis of three deep learning architectures — Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and a hybrid Convolutional Neural Network–LSTM (CNN-LSTM) for data-driven SoC estimation using voltage, current, and temperature as input parameters. Using a publicly available Li-ion battery dataset, we preprocess the time-series data into fixed-length windows and train each model under identical conditions. The models are then evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). To ensure a statistically sound assessment, each model's performance was evaluated over 10 independent runs. The results show that all three architectures are capable of learning time-based patterns for SoC estimation, but their performance and stability vary significantly.**

Keywords— Lithium-ion battery (LIB), State-of-Charge (SoC) estimation, LSTM, GRU, CNN-LSTM, Deep learning, Time-series prediction, Battery Management Systems (BMS)

I. INTRODUCTION

Lithium-ion batteries have become exceedingly important components in modern energy storage systems (ESS) due to their high energy density, prolonged cycle life, low discharge rate, and adaptability across various applications, such as EVs [1], portable electronics, renewable energy storage systems [2][3] and even in space applications [4]. A critical parameter in ensuring their efficient and safe operation is SoC, which represents the remaining usable capacity of the battery [3]. LIBs are complex electrochemical systems whose characteristics are highly dependent on various internal and external conditions, making direct and precise SoC determination inherently challenging. Accurate SoC estimation empowers optimal energy management and it prevents overcharging or deep discharging, and extends battery lifespan. Traditional SoC estimating approaches, including Coulomb counting, open-circuit voltage (OCV) methods, and model-based techniques including Kalman filters, at times encounter limitations such as drift errors, sensitivity to measurement noise, and the need for precise battery modelling [2]. These drawbacks have prompted a trend toward data-driven techniques, particularly deep learning (DL), which can efficiently simulate complicated, nonlinear, and time-dependent correlations in battery behaviour without requiring explicit physical models. RNNs, LSTMs [3][4][5], and GRU models excel in capturing temporal correlations in time-series data like battery voltage, current, and temperature. Similarly, hybrid architectures such as CNN-LSTM have shown potential by combining the feature extraction strength of Convolutional Neural Networks (CNNs) with the sequential modelling capabilities of LSTMs.

This research presents a rigorous comparative study of three prominent DL models—LSTM, GRU, and Hybrid CNN-LSTM—for the critical task of SoC estimation in LIBs. The study leverages real-world battery cycling data; the models were trained on time-series inputs of voltage, current, and temperature, with SoC values calculated via a robust Coulomb counting method. To provide a statistically sound and reliable assessment, each model's performance was evaluated over 10 independent training and evaluation runs. The study assesses performance in terms of average accuracy, model stability, and computational efficiency. The findings reveal a compelling trade-off: while the GRU model achieved the highest average accuracy, the CNN-LSTM proved to be the most stable and robust architecture, demonstrating the lowest variability across all runs. The study concludes that for reliable SoC prediction in intelligent battery management systems, model stability is as critical as average accuracy.

II. BACKGROUND AND RELATED WORK

A. Traditional SOC Estimation Method

The precise estimation of a LIB's SoC is a foundational task in Battery Management Systems (BMS). Traditionally, SoC has been estimated using model-based or direct measurement methods. One of the most widely used techniques is Coulomb Counting, which estimates the SoC by integrating the battery's current over time. Even though it is simple and easy to implement, Coulomb counting is highly prone to measurement noise and accumulated errors over consecutive cycles, leading to a phenomenon known as "SoC drift."

Another common method is the Open Circuit Voltage (OCV) method [2], which correlates the battery's voltage at equilibrium with its SoC. This approach is generally highly accurate, but it requires the battery to remain at rest for extended periods (often several hours) to reach a stable OCV. Consequently, it is impractical for real-time SoC estimation in dynamic applications such as electric vehicles. The limitations of these traditional methods, coupled with the complex and non-linear behaviour of battery systems, have driven research towards more advanced, data-driven approaches that can learn the complicated relationships between battery parameters and SoC without relying on simplified models

B. Data-Driven and Machine Learning Approaches

Since the contemporary world has been shifting towards data-driven approaches in almost all domains, electrical engineering is no exception. This paradigm shift is driven by the increasing availability of vast datasets from sensors and monitoring systems, coupled with the computational power to process them effectively. In the field of battery management, this trend is particularly transformative.
Traditional methods, while being foundational, often struggle to handle the inherent complexities and nonlinearities of battery behaviour over time. The degradation of internal components, changes in electrochemical properties due to temperature fluctuations, and the effects of varied charging/discharging cycles are difficult to capture with fixed-parameter physical models. Consequently, model-based estimation techniques often require frequent recalibration and can suffer from reduced accuracy as the battery ages.
Data-driven methods, particularly those based on machine learning and deep learning, provide a compelling alternative to traditional modeling approaches [6][7]. By leveraging historical operational data—such as voltage, current, temperature, and observed capacity variations—these models can automatically learn the complex, time-dependent behaviors that govern battery performance. This removes the need for manually derived physical models and enables more flexible and adaptive estimation. Moreover, the capacity of deep learning techniques to handle large-scale, noisy datasets and uncover hidden patterns makes them highly promising for achieving the accuracy and reliability demanded by next-generation battery management systems, from electric vehicles to large-scale energy storage applications.

C. *LSTM and GRU Networks*

a) LSTM networks represent a specialised and highly effective variant of RNNs [3][4]. They are specifically designed to learn, process, and classify sequential data, excelling where traditional RNNs fail. An identifying feature of LSTMs is the introduction of a "memory cell," also known as the cell state, which allows the network to retain information over extended periods, effectively addressing the challenge of learning long-term dependencies. Traditional RNNs are prone to the vanishing gradient problem during backpropagation through time. This issue severely limits their ability to capture and utilize information from distant time steps, making them ineffective for tasks requiring long-term memory. LSTMs ingeniously overcome this by incorporating a sophisticated gating mechanism that precisely controls the flow of information into, out of, and within the memory cell. The LSTM architecture features a chain-like structure comprising multiple neural networks and distinct memory blocks, known as cells. Information is stored within these cells, and their manipulation is governed by three primary gates: the input gate, forget gate, and the output gate.

1. Input gate: Regulates what data is entered into the memory cell.
2. The forget gate: Chooses which data is deleted from the memory cell.
3. Output gate: Regulates the data that the memory cell

Let x_t be the input at time step t, h_{t-1} be the hidden state from the previous step, and c_{t-1} be the previous cell state. Each gate in the network has its own set of weights W and biases b. We use the sigmoid function σ and the hyperbolic tangent function tanh for the gate activations and state updates.

The calculations for a single LSTM time step are:

$$f_t = \sigma\left(Wf \cdot [h_{t-1}, x_t] + b_f\right)$$
$$i_t = \sigma\,(Wi \cdot [h_{t-1}, x_t] + b_i)$$
$$c{\sim}t = tanh(Wc \cdot [h_{t-1}, x_t] + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot c_t$$
$$o_t = \sigma\,(Wo \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \odot tanh(c_t)$$

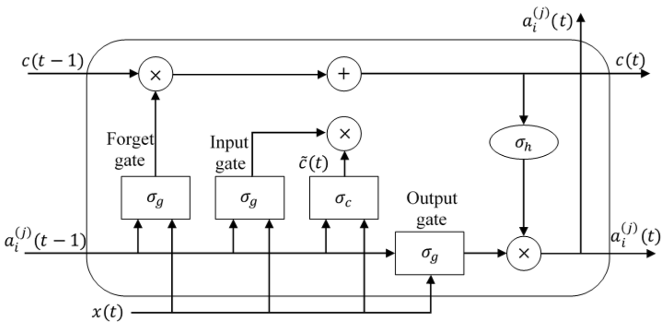

Fig. 1. Basic architecture of an LSTM network

b) Gated Recurrent Units (GRUs) are a type of RNNs that emerged as a simpler, but more computationally efficient alternative to LSTMs for processing sequential data [5]. Like LSTMs, GRUs were designed to address the vanishing gradient problem inherent in traditional RNNs, enabling them to capture long-range dependencies effectively. Their key distinction lies in their streamlined architecture, utilising fewer gates than LSTMs. GRUs operate with two primary gates that regulate the flow of information: the Update gate and the reset gate.
The following set of equations can describe the operations within a GRU:

Update Gate: $z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$

Reset Gate: $r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$

Candidate Hidden State: $\widetilde{h} = tanh(W * [r_t \odot h_{t-1}, x_t])$

Final Hidden State: $h_t = (1 - z_t) \odot z_t \odot \widetilde{h_t}$

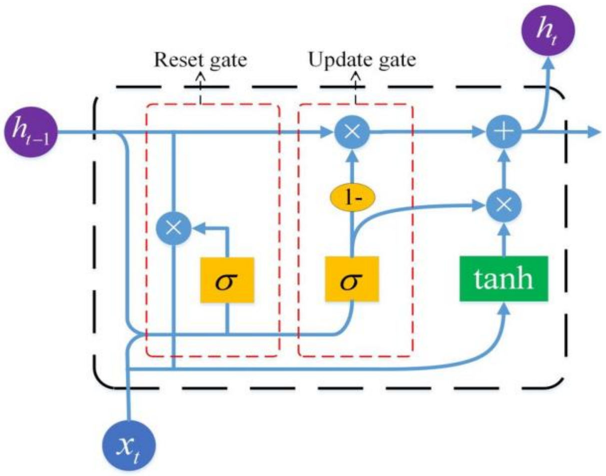

Fig. 2. Basic architecture of a GRU model

D) CNN-LSTM Hybrid Models

CNN-LSTM hybrid models represent a robust class of deep learning architectures that combine the complementary capabilities of CNNs and LSTMs. CNNs are effective at capturing local patterns and hierarchical spatial features from the data, whereas LSTMs are well-suited for modelling sequential information and long-range temporal dependencies. Together, they enable efficient learning from data that contains both spatial and temporal characteristics. This synergistic combination enables CNN-LSTMs to effectively process sequential data that contains both spatial correlations within individual time steps and temporal patterns across the sequence, such as multivariate time series originating from sensor arrays.[8]

In a typical CNN-LSTM configuration, the CNN layers are positioned at the front end of the network. Their role is to process the raw input data, such as battery current, voltage, and temperature readings at each time step, to extract relevant, high-level features. For battery SoC estimation, this might involve identifying characteristic patterns or signatures across the multiple input variables within a short time window. The feature maps generated by the CNN layers are then reshaped and fed as input into the subsequent LSTM layers. These LSTM layers are then responsible for learning the temporal dynamics and long-term dependencies that exist within these extracted features over the entire sequence. For time series data, 1D CNNs are commonly employed, where convolutional kernels traverse in a single direction across the time dimension to detect patterns. To mitigate overfitting, dropout layers are frequently incorporated between LSTM layers. The final layer is typically a dense (fully connected) layer, configured for regression to output the estimated SoC value.[9][10]

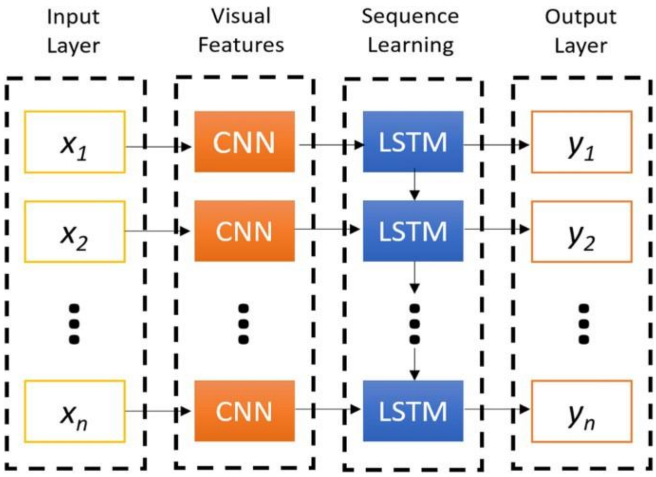

Fig. 3. Hybrid CNN-LSTM model architecture

III. METHODOLOGY

A. *Data Collection, Input Variables & Data Preposition*

The experimental data for this study was sourced from a publicly available dataset of a Li-ion battery cell. This dataset comprises real-world battery cycling data collected during various charging and discharging cycles under controlled conditions. The data is presented as a high-frequency time series, which is essential for capturing the dynamic behavior of the battery. To serve as input for the deep learning models, three key operational parameters were selected: Voltage measured, current measured & temperature measured. The data did not include an explicit time-step between measurements; therefore, a critical assumption of a constant 1-second time interval was made to facilitate subsequent calculations. The target variable - SoC, was not a direct measurement in the original file. It was derived through a simplified Coulomb counting method, which integrates the measured current over the assumed time intervals relative to a nominal battery capacity of 2900 mAh. Finally, the dataset was partitioned into a training set (80%) and a separate, held-out test set (20%) to ensure an unbiased evaluation of the models' performance. The time-series data was then formatted into fixed-length windows of 50-time steps to align with the input requirements of the recurrent neural networks.

B. *Model Architectures and Parameters:*

Three distinct deep learning models—LSTM, GRU, and a hybrid CNN-LSTM—were constructed to perform a comparative analysis for SoC estimation. Each model was designed to capture temporal correlations in time-series data like battery voltage, current, and temperature.[10][11][12]

1. Standalone LSTM Model: This model is a specialized variant of RNNs. It is designed with a sophisticated gating mechanism to control the flow of information through its "memory cell". The LSTM architecture is prone to the vanishing or exploding gradient problem during backpropagation through time. The total number of parameters for this model was 10,851.

2. Standalone GRU Model: The GRU is a simpler, more computationally efficient alternative to the LSTM. It addresses the vanishing gradient problem by using fewer gates and a streamlined architecture. This allows it to capture long-range dependencies effectively. The GRU model had a total of 7,851 parameters.

3. CNN-LSTM Hybrid Model: This model was designed to leverage the complementary strengths of Convolutional Neural Networks (CNNs) and LSTMs. The CNN layers act as a front-end to the network, adept at extracting hierarchical spatial features and local patterns from the input data. The features extracted by the CNN are then fed into the subsequent LSTM layers, which are responsible for learning the temporal dynamics and long-term dependencies within the sequence. The total number of parameters for this hybrid model was 23,691.

TABLE I. Model Architecture Summary

Model	Trainable Parameters	Neuron s/Filters	Architecture
LSTM	10,851	50 units	Single standalone recurrent layer
GRU	7,851	50 units	Single standalone recurrent layer
CNN-LSTM	23,691	64 filters, 50 units	Hybrid (1D CNN followed by LSTM)

C. Training and Evaluation Protocol:

The training and evaluation protocol was designed to provide a rigorous and fair comparative study of all three deep learning models. Each model was trained under identical conditions to assess their performance. The models were optimised using the Adam optimiser with the Mean Squared Error (MSE) as the loss function. To ensure a statistically sound and reliable assessment, each model's performance was evaluated over 10 independent training and evaluation runs. Performance was assessed in terms of average accuracy, model stability, and computational efficiency. The models were evaluated using three key metrics: RMSE, MAE, and R^2 [1]. Computational efficiency was also analyzed based on the total training time per epoch.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the results of a rigorous comparative study of the three deep learning models. Each model's performance was evaluated over 10 independent runs to ensure a statistically sound and reliable assessment of its capabilities and stability

A. The final performance of each model, assessed on a held-out test set, is summarized in the table below. The metrics are presented as an average and standard deviation across the 10 experimental runs.

TABLE. II. Model Evaluation Metrics

Model	Avg RMSE	Std Dev RMSE	Avg MAE	Std Dev MAE	Avg R^2	Std Dev R^2
LSTM	0.0059	0.0039	0.0045	0.0031	0.5728	0.6289
GRU	0.0046	0.0021	0.0036	0.0017	0.7867	0.2117
CNN-LSTM	0.005	0.002	0.0039	0.0016	0.759	0.1617

The GRU model achieved the highest average accuracy, with the lowest average RMSE of 0.0046 and the highest average R^2 of 0.7867. The CNN-LSTM model followed closely, with a competitive average R^2 of 0.7590. In contrast, the standalone LSTM model proved to be the least effective, demonstrating the lowest average performance across all metrics.

B. Model Stability and Trade-offs

The standard deviation metrics provide crucial insights into the stability and robustness of each model. A lower standard deviation indicates that the model's performance is consistent and reliable across different training runs. The CNN-LSTM model proved to be the most stable and robust architecture. It demonstrated the lowest standard deviation across all performance metrics, including an exceptionally low Std Dev R^2 of 0.1617. This indicates that its performance is the most consistent and predictable, a critical factor for a reliable battery management system. The GRU model, while achieving the best average accuracy, showed a higher degree of variability in its performance (R^2 of 0.2117). The standalone LSTM model was the most unstable, with its performance varying dramatically across runs.

This analysis highlights a key finding of this research: a clear trade-off exists between a model's average accuracy and its stability. While the GRU model excels in average performance, the CNN-LSTM offers superior reliability and consistency, which may be more valuable in safety-critical applications.

C. Virtual Results

Visual comparisons of the models' performance further support the numerical results. The prediction plots show that both the GRU and CNN-LSTM models accurately tracked the actual SoC values on the test set. These models successfully captured the temporal dynamics of the battery, producing smooth and consistent prediction curves. In contrast, the LSTM model's prediction curve was often erratic and failed to accurately follow the actual SoC, which visually confirms its poor performance metrics. The loss plots also provide evidence of this, showing that the GRU and CNN-LSTM models converged quickly and stably, while the

LSTM model's training process was less consistent.

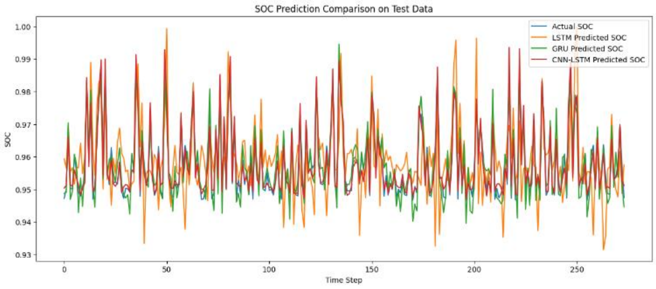

Fig. 4. Actual vs predicted SOC data for LSTM, GRU, and CNN- LSTM Deep Learning Models

The plot visually demonstrates the performance of each model on a held-out test set. The CNN-LSTM and GRU models' predicted SoC curves closely track the actual SoC values, indicating their effectiveness in capturing the temporal dynamics of the battery. In contrast, the LSTM model's prediction curve appears erratic and deviates significantly from the actual SoC, confirming its inferior performance metrics. The visual evidence from this figure supports the numerical results, showing that the GRU and CNN-LSTM architectures provide a smoother and more consistent prediction of the battery's SoC.

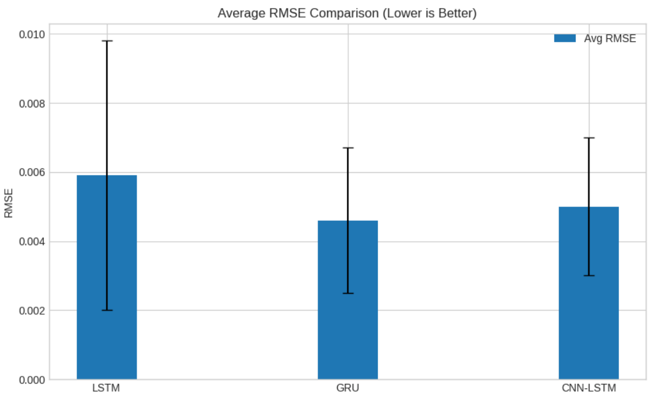

Fig. 5. Average RMSE Comparison

Root Mean Square Error (RMSE) is given by:

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (SoC\_actual, i - SoC\_pred, i)^2}$$

The GRU model achieved the highest average accuracy, as indicated by its lowest average RMSE of 0.0046. However, the CNN-LSTM model proved to be the most stable and reliable, showing the lowest standard deviation in its performance metrics. This is visually represented by its shorter error bar compared to the other models. The LSTM model, in contrast, was the least accurate and most unstable, with the highest average RMSE and the largest standard deviation.

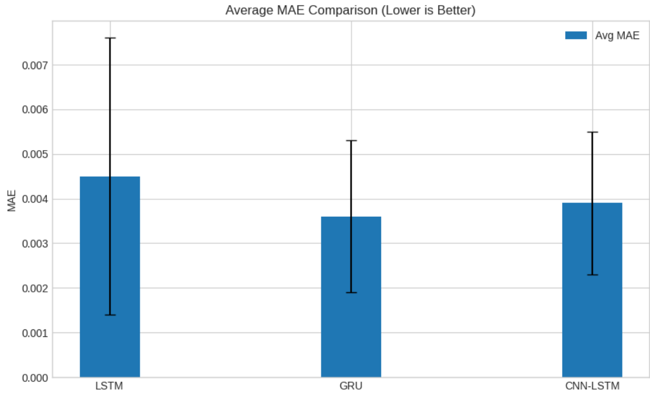

Fig. 6. Average MAE Comparison

Mean Absolute Error (MAE) is given by:

$$MAE = 1/n \sum_{i=1}^{n} |(SoC\_actual, i - SoC\_pred, i)|$$

The GRU model has the lowest average MAE, indicating its superior average accuracy.

The CNN-LSTM model has a slightly higher average MAE than the GRU, but is still more accurate on average than the LSTM model.

The LSTM model has the highest average MAE and the largest standard deviation, suggesting it is the least accurate and least stable of the three models. In contrast, the CNN-LSTM model has the smallest standard deviation, highlighting its consistency and robustness.

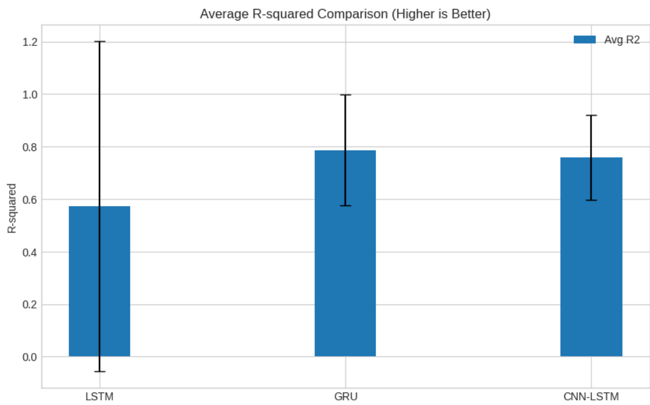

Fig 7. Average R^2 Comparison

The coefficient of determination, or R-squared, is given by:

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n} ((SoC\_actual, i - SoC\_pred, i)^2)}{\left(\sum_{i=1}^{n} (SoC\_actual, i - \overline{SoC\_actual, \ i})^2\right)} \right)$$

The bar chart titled "Average R-squared Comparison" visually compares the performance of the LSTM, GRU, and CNN-LSTM models. The GRU model shows the highest average R^2 score, indicating it has the best fit on average. The CNN-LSTM model is a close second. The LSTM model performs the worst, with the lowest average R^2. The error bars highlight the superior stability and reliability of the GRU and CNN-LSTM models compared to the highly variable LSTM.

D. Computational Details

The entire project was built and executed within a single, consistent environment to ensure a fair and reproducible comparative analysis. The core programming language used was Python, with all deep learning models developed using the TensorFlow framework and its high-level Keras API. For data management and preprocessing, essential libraries such as Pandas for data manipulation and NumPy for numerical operations were employed. The Scikit-learn library was utilized for data normalization and calculating final evaluation metrics.

All experiments were conducted in a Google Colab environment, which provided access to a GPU (Graphics Processing Unit). The use of a GPU was critical for accelerating the training process, allowing for the efficient completion of the multi-run comparative analysis. This setup ensured that all three models: LSTM, GRU, and CNN-LSTM were trained and evaluated under identical software and hardware conditions, meaning any observed differences in performance, stability, or computational efficiency can be confidently attributed to the models' distinct architectural designs.

V. CONCLUSION

In Conclusion, in this study, the authors compared various deep learning models to find the most suitable architecture for a reliable BMS. The study found a key trade-off between the accuracy and stability of these models for estimating the SoC of LIBs. While the GRU model achieved the highest average accuracy, the CNN-LSTM model was the most stable and robust. The CNN-LSTM model demonstrated the lowest variability in its performance across multiple runs, making it the most consistent and predictable architecture. This research highlights that for reliable SoC prediction, model stability is as critical as average accuracy.

Future research can extend this project by systematically tuning the models to maximize their performance and stability. The scope can be expanded to include State of Health (SOH) prediction and to test the models' generalization on the full battery dataset. Ultimately, the next step involves preparing the optimal model for real-time deployment in a Battery Management Systems.

REFERENCES

[1] Madani, S.S.; Shabeer, Y.; Fowler, M.; Panchal, S.; Chaoui, H.; Mekhilef, S.; Dou, S.X.; See, K. Artificial Intelligence and Digital Twin Technologies for Intelligent Lithium-Ion Battery Management Systems: A Comprehensive Review of State Estimation, Lifecycle Optimization, and Cloud-Edge Integration. *Batteries* 2025, *11*, 298. https://doi.org/10.3390/batteries11080298

[2] Ralls AM, Leong K, Clayton J, Fuelling P, Mercer C, Navarro V, Menezes PL. The Role of Lithium-Ion Batteries in the Growing Trend of Electric Vehicles. Materials (Basel). 2023 Sep 4;16(17):6063. doi: 10.3390/ma16176063. PMID: 37687758; PMCID: PMC10488475.

[3] Dipanshu Naware, Raviteja Badigenchala, Arghya Mitra, Debapriya Das, Impact of demand response on battery energy storage degradation using gbest-guided artificial bee colony algorithm with forecasted solar insolation, Journal of Energy Storage, Volume 52, Part B, 2022, 104915, ISSN 2352-152X, https://doi.org/10.1016/j.est.2022.104915.

[4] R. Bugga, M. Smart, J. Whitacre and W. West, "Lithium Ion Batteries for Space Applications," *2007 IEEE Aerospace Conference*, Big Sky, MT, USA, 2007, pp. 1-7, doi: 10.1109/AERO.2007.352728.

[5] D. H. Cheng Lam, Y. Seng Lim, L. C. Hau and J. Wong, "Long Short-Term Memory Recurrent Neural Network for Estimating State of Charge of Energy Storage System for Grid Services," *2022 4th International Conference on Smart Power & Internet Energy Systems (SPIES)*, Beijing, China, 2022, pp. 1887-1894, doi: 10.1109/SPIES55999.2022.10082116.

[6] Wu, Y.; Bai, D.; Zhang, K.; Li, Y.; Yang, F. Advancements in the estimation of the state of charge of lithium-ion battery: a comprehensive review of traditional and deep learning approaches. *J.Mater. Inf.* 2025, *5*, 18. http://dx.doi.org/10.20517/jmi.2024.84

[7] T. Bhardwaj, V. S. Kale, M. S. Ballal and S. Khond, "Comparative Analysis of Machine Learning Models for Li-Ion Battery SoC Estimation," *2023 IEEE 3rd International Conference on Smart Technologies for Power, Energy and Control (STPEC)*, Bhubaneswar, India, 2023, pp. 1-6, doi: 10.1109/STPEC59253.2023.10430560.

[8] Zeinab Sherkatghanad, Amin Ghazanfari, Vladimir Makarenkov, A self-attention-based CNN-Bi-LSTM model for accurate state-of-charge estimation of lithium-ion batteries, Journal of Energy Storage, Volume 88, 2024, 111524, ISSN 2352-152X, https://doi.org/10.1016/j.est.2024.111524.

[9] M. Naguib, P. J. Kollmeyer and A. Emadi, "State of Charge Estimation of Lithium-Ion Batteries: Comparison of GRU, LSTM, and Temporal Convolutional Deep Neural Networks," *2023 IEEE Transportation Electrification Conference & Expo (ITEC)*, Detroit, MI, USA, 2023, pp. 1-6, doi: 10.1109/ITEC55900.2023.10186991.

[10] Y. Liu, P. Dai and X. Chen, "Joint Estimation of SOC, SOH and SOT for Battery Energy Storage System Based on CNN-LSTM," *2024 IEEE 2nd International Conference on Power Science and Technology (ICPST)*, Dali, China, 2024, pp. 1884-1888, doi: 10.1109/ICPST61417.2024.10602156.

[11] Miao, Y.; Gao, Y.; Liu, X.; Liang, Y.; Liu, L. Analysis of State-of-Charge Estimation Methods for Li-Ion Batteries Considering Wide Temperature Range. *Energies* 2025, *18*, 1188. https://doi.org/10.3390/en18051188

[12] Carrera, R.; Quiroz, L.; Guevara, C.; Acosta-Vargas, P. State-of-Charge Estimation of Medium- and High-Voltage Batteries Using LSTM Neural Networks Optimized with Genetic Algorithms. *Sensors* 2025, *25*, 4632. https://doi.org/10.3390/s2515463