

RESEARCH ON THE STATE OF CHARGE ESTIMATION OF LITHIUM-ION BATTERY USING TRANSFORMER REGRESSOR FOR INTELLIGENT BATTERY MANAGEMENT SYSTEMS

G.R.Radhika¹, S.Kanthalakshmi²

¹Electronics and Instrumentation Engineering, Government College of Technology, Coimbatore, Tamil Nadu, India, 641013, E-mail: radhika@gct.ac.in

²Electrical and Electronics Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India, 641004, Email: skl.eee@psgtech.ac.in

ABSTRACT: For the purpose of assuring the safety, dependability, and prolonged lifespan of energy storage systems in electric vehicles, it is vital to have an accurate estimation of the level of charge of the lithium-ion battery. Attempts to capture the complicated and non-linear behaviors of batteries are frequently unsuccessful when using traditional methods. From the perspective of Battery Management System (BMS) applications, this study gives a comparative analysis of a number of different artificial intelligence models, with a particular emphasis on how well these models perform in terms of state of charge estimate. Specifically, in order to estimate the State of Charge (SoC), models such as Random Forest, Support Vector Machine, Convolutional Neural Network, and Transformer are developed and assessed with the use of data pertaining to voltage, current, and temperature. The research endeavours an exhaustive performance comparison with the objective of determining which AI-based solution for BMS applications represents the highest level of certainty and robustness.

Keywords: State of Charge, Lithium-ion battery, Random Forest, Support Vector Machine, Convolutional Neural Network, Transformer Regressor.

1. INTRODUCTION

Greenhouse gas emissions turn the swift development of renewable energy and imminent energy technologies which brings electric vehicles (EV) into the primary need of our day-to-day activity. The automobile zone is transitioning from internal combustion engines to electric vehicles to mitigate carbon emissions. Lithium-ion batteries are largely selected for their notable durability and high energy efficiency. It is the most superior energy-storing system with high energy density, extended cycle life, and efficiency [1]. Reviewing battery performance are crucial for the safety operation.

State of Charge (SOC) estimation is a contentious matter for an effective Battery Management System (BMS). Accurate SOC estimation can avoid battery over-charge, over-discharge, resulting in improved safety, operational reliability, and better performance.[2]. The state of charge (SOC) is an implicit condition derived indirectly from other parameters like as current, voltage, and temperature, often influenced more by variations and measurement incorrectness, than by any fundamental physical law [3].

Convolutional neural networks (CNN) are largely studied for their competence in modelling the complexity of battery dynamics for state of charge (SOC) prediction, and modern growth in deep learning approaches have also improved state of health (SOH) estimate in scenarios with minimal labelling [4]. Traditional models like Random Forest (RF) and Support Vector Machines (SVM) are fast and easy to run, but they cannot understand long-term changes in the battery very well. Deep learning methods, such as 1D-CNN, excel at identifying patterns; nevertheless, they primarily concentrate on proximate data and lack the capability to comprehend long-term relationships. There is a need for a better model that can understand long-term battery behaviour accurately while still being fast and not needing very powerful computers.

The primary objective involves a comparative analysis of four distinct architectures: Random Forest, SVM, 1D-CNN, and the Transformer model. By training these models on battery datasets taken from NASA Prognostics Battery Dataset (B0005- single Li-ion 18650, rated 2 Ah, charged at 1.5A to 4.2V, discharged at 2A to 2.7V) including voltage, current, and temperature, the State of Charge for the battery is calculated. Finally, based on the model prediction and error metrics, model performance is evaluated.

2. LITERATURE SURVEY

Numerous Transformer-based battery estimates models created in the past decade have achieved RMSE values below 1% in recent research, as noted in Guirguis and Ahmed's review [5]. A prototype of the DRSN-CW-LSTM was originally presented by Wang et al. [6]. Despite the fact that these methods improved the performance of estimation, they continue to be heavily dependent on recurrent learning and exhibit a

considerable amount of computational complexity. Furthermore, they have limited adaptability to a variety of battery chemistries and running conditions, which actually hinders their feasible implementation in battery management systems. Masoud A. Sahhouk et al. [7] presented a comparative evaluation of the conventional coulomb counting method and a data-driven random forest model for state of charge estimation in new and second-life lithium-ion batteries and obtained R^2 value close to 0.998. Anitha.K et al. [8] applied Random Forest Regression for predicting battery state of health and achieved high accuracy with the mean absolute error below 2%. Sid-Ali Amamra [9] implemented a Random Forest model optimized through grid search for hyperparameter tuning to predict capacity fade in high-capacity lithium cells and achieved an R^2 of 0.92 and an RMSE of 0.06. Lijun Qian et al. [10] combined a decision tree for feature selection with an improved SVR algorithm for SoH estimation. The innovation lies in optimizing the differentiability of the SVR objective function to improve training efficiency and accuracy. The simulation findings indicate that the enhanced SVR algorithm reduces training time by 83.9% relative to the original SVR method, although with a minor compromise in predictive accuracy. Wei Xiong et al. [11] introduced an approach for online SOH estimation of second-use lithium-ion batteries based on a weighted least squares support vector machine and reported an RMSE less than 1.85%. Ran Li et al. [12] proposed an online estimation method of lithium-ion health based on particle swarm support vector machine algorithm using National Aeronautics and Space Administration (NASA) battery samples and showed the error of the SOH estimation result is not more than 2.5%. Vasanthan Buvanesan.A et al. [13] proposed a architecture combining Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory (BiLSTM) to boost SOH prediction. A MAE of 0.3867% and an average RMSE of 0.4951% were attained by the model after it was tested using the battery dataset from NASA. Ephrem Chemali et al [14] introduced a CNN framework that directly estimates SoH from voltage, current, and temperature measurements during charging and achieved a mean average error (MAE) as low as 0.8%. J. Harinarayanan, P. Balamurugan [15] conducted a thorough evaluation of multiple SOC estimate methods, including both classic approaches Coulomb Counting and extended Kalman filter and machine learning techniques under different load profile on lithium-ion pouch cell and reported an RMSE 0.0229, MSE 0.0005, MAE 0.0139 using Random Forest approach. Zhou.Y et al. [16] proposed A combined data-driven modelling approach based on Least squares support vector machine based on particle swarm optimization and Unscented Kalman Filter to obtain a better state of charge estimation accuracy and showed that maximum SOC error under various working situations is kept to 0.5%. Erwin Sutanto et al. [17] employed the gradient vector and a support vector machine (SVM) and achieved the values of MAPE (%), MAE, RMSE as 0.3%, 0.0106, 0.0136. Minggang Zheng, Xing Luo [18] utilized SVM for accurate SOH estimate and developed an effective SOC estimation model by integrating the advantages of CNN and LSTM networks. The findings of the validation process on the Oxford and the NASA battery dataset indicate that the model accomplish a high level of accuracy of certainty and stableness for the entirety of the battery's life. Xun Gong et al. [19] proposed a SOC estimation method based on the fusion of convolutional neural network-transformer (CNN-Transformer) and square root unscented Kalman filter (SRUKF). The data is collected in a controlled environment with three different temperatures. The study results show that the SOC estimate approach recommended in this research is stable, with an RMSE ranging from 2.69% to 4.22%. John Guirguis et al. [20] introduced a comprehensive analysis of the capabilities of diverse Transformer-based models in assessing the SOC, SOH, SOP of Lithium-Ion Batteries using both regular and inverted embedding techniques and achieved a RMSE of 0.74% using the Reformer model.

3.SOC estimation

In this research, instead of using complex battery equations, powerful tools like Google Colab and Python are used to train models with large amounts of data. These models learn how the battery behaves by finding patterns in the data. The State-of-Charge (SOC) estimation in this research is performed using the Coulomb Counting technique, given by:

$$SOC(t) = SOC(t_0) - \frac{1}{C_{nominal}} \int_{t_0}^t (\eta \cdot I(\tau) d\tau) \quad (1)$$

SOC(t) - current state of charge

SOC(t_0) - initial state of charge

$C_{nominal}$ - nominal capacity

$I(\tau)$ - instantaneous current

η - coulombic efficiency

τ - time variable used during the integration process to track current flow.

3.1DEVELOPMENT ENVIRONMENT

In this project work, Google Colab is used as the main working platform to run deep learning models skilfully. Complex time-series models requiring parallel processing power can be easily trained in Colab. Colab provides access to Graphics Processing Units (GPUs) NVIDIA T4, that helps in fast iteration of model.

3.2 DATA CLEANING AND NOISE REDUCTION USING PANDAS & NUMPY

The nature of the data used to train a model determines how accurate it will be. Under real-world situations, sensor data may exhibit inaccuracies due to noise, interference, or device malfunctions. The data must be cleaned and corrected prior to its utilization in the model. Initially, the missing data were found using Pandas. Rather than eliminating those data points, a technique known as linear interpolation was employed. This technique imputes missing values by estimating them from proximate data points, hence maintaining the data's smoothness and continuity for the model.

NumPy is used to find out data outliers in the battery dataset and were adjusted so that the model does not learn impractical battery nature. A moving average filter is used to remove noise in the sensor data. The data is smoothed by averaging proximate values over an interval and preserves important variations in voltage and temperature.

3.3 DATA NORMALIZATION AND SLIDING WINDOW TECHNIQUES

Neural networks are sensitive to the scale of input data. Sometimes, instead of limiting values between 0 and 1, it is useful to make the data centered around zero. Standard Scaler is used.

$$x_{\text{std}} = \frac{(x - \mu)}{\sigma} \quad (2)$$

$x \rightarrow$ input value

$\mu \rightarrow$ Mean of all data values

$\sigma \rightarrow$ Standard deviation

$x_{\text{std}} \rightarrow$ Standardized value

Sliding Window Transformation is used to prepare the time-series data for model training. Window Length: Each input sequence consists of 100-time steps. The window slides over the data with a stride of 10. Each time step within a window includes 3 features: voltage, current and temperature.

4. METHODOLOGY

The figure 1. shows the complete workflow of the project. Battery dataset that contains measurements like voltage (V), current (I), temperature (T), time is taken and first goes through data preprocessing and feature engineering. In this step, the data is cleaned, normalized, filtered and converted into windows.

The same training data is given to four different models. Each model is trained separately, resulting in four trained models. These models are then evaluated using metrics like RMSE, MAE and R^2 .

4.1 RANDOM FOREST

It is a machine learning algorithm that incorporates several trees to diminish errors and achieve accurate and stable predictions.

It reduces overfitting and improves overall dependability by aggregating the results of multiple trees instead of depending on a single model.

Every single decision tree within the forest is trained with a collection of data that is totally arbitrary.

All the trees make their own predictions, and the end result is just the mean of all these predictions.

$$\hat{y}(x) = \frac{1}{N} \sum_{i=1}^N \hat{y}_i(x) \quad (3)$$

where $\hat{y}_i(x)$ - forecast from the i -th tree;

N - overall sum of trees.

4.2 SUPPORT VECTOR MACHINE (SVM)

SVM is a supervised machine learning algorithm. identifies the optimal hyperplane to separates specific classes within the dataset. The primary objective of SVM is to optimize the margin between the two classes. Figure 3 illustrates SVM structure specifically designed for regression. The structure is formulated as three layers: an input, a kernel layer, and an output neuron.

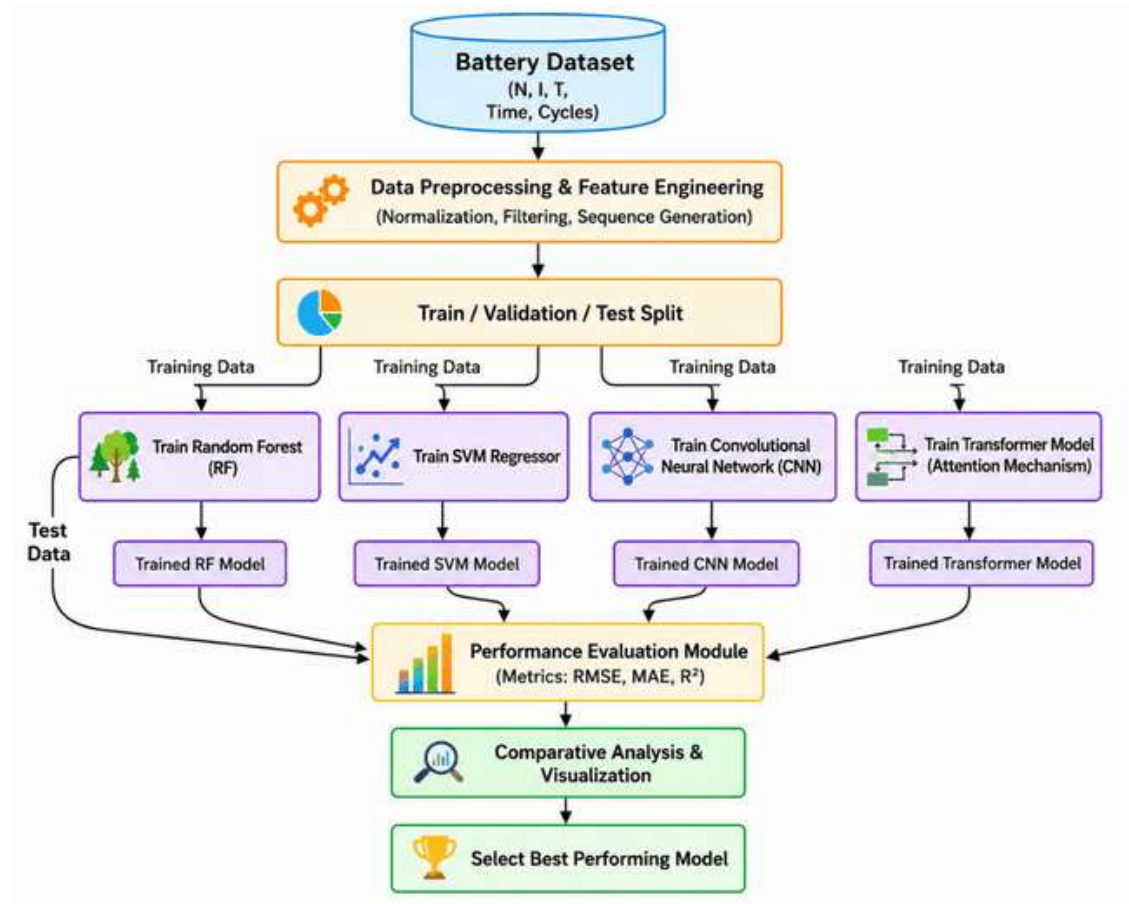


Figure 1. Methodology

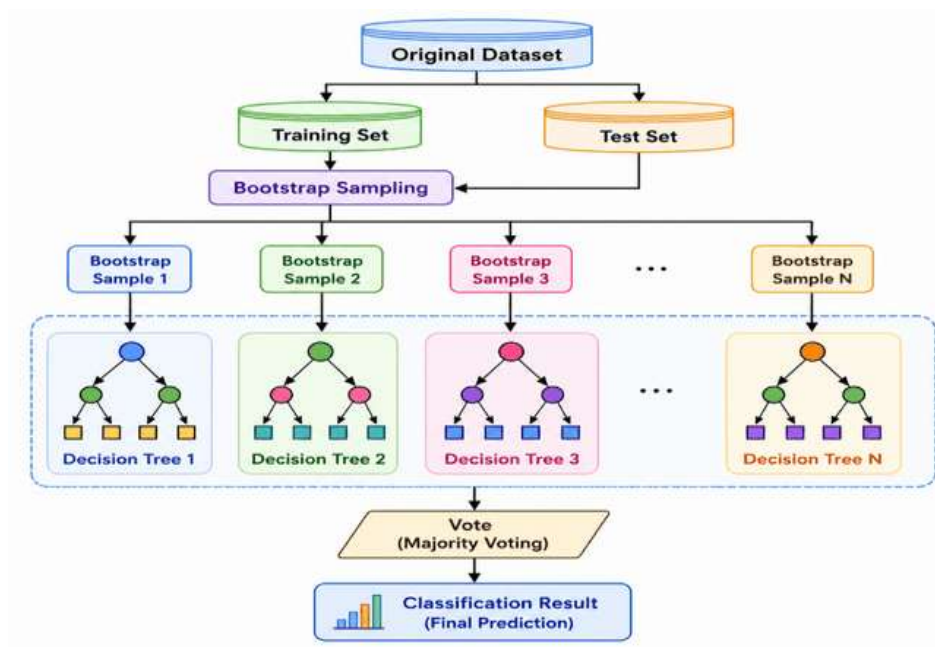


Figure 2. Random Forest Architecture

The procedure commences with the input vector X , which contains the feature elements $(x_1, x_2, \dots, x_N)$. In the hidden layer, the model utilizes a Kernel Function, denoted as $K(x_i, x)$ which maps the input data into a high-dimensional feature space.

The last stage is the output neuron, which aggregates the weighted results from the hidden layer. Each connection between the hidden nodes and the output is scaled by a specific weight, represented here as $(\alpha_i - \alpha_i^*)$.

4.4 1D-CNN

Feature extraction is a strong suit of the CNN, a feed-forward neural network. Its constituent layers are as follows: input, convolutional, pooling, fully connected, and output. The fully connected layer receives high-level data as input from the integration of latent features extracted by convolutional and pooling layers, which capture spatial attributes with a low number of parameters.

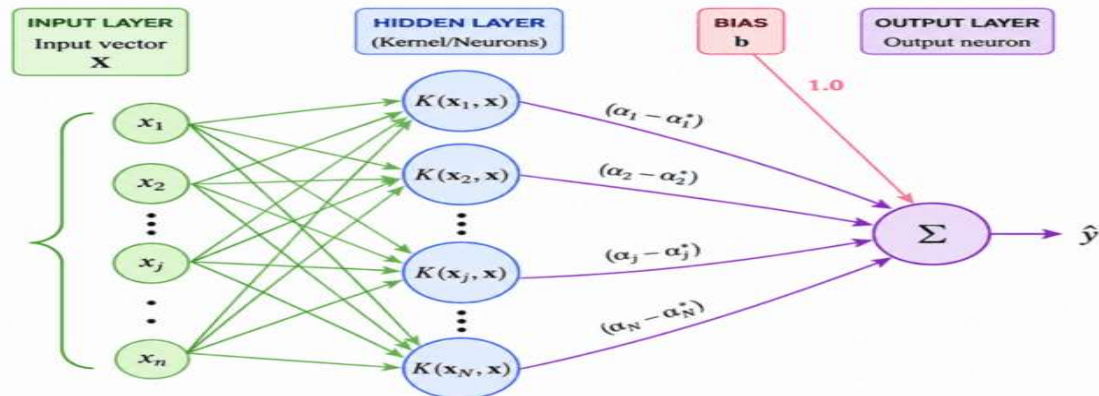


Figure 3. SVM Architecture

The equation that explains the relationship from the input to the convolutional layer is:

$$h_{nn} = \sigma(\omega_{nn} \times x_n + b_{nn}) \quad (4)$$

where ω_{nn} is the weight; b_{nn} is the bias; σ is the activation function; and x_n is the input at the present instant.

4.5 TRANSFORMER

A Transformer architecture can be applied to analyze battery sensor data sequences. Transformer's attention mechanism can examine the entire sequence of voltage, current, and temperature readings at once. Sequential models like RNNs forget important long-term trends in battery behavior due to the vanishing gradient problem. Using Transformers, the BMS can capture complex relationships across the full time-series data, such as how past temperature spikes or current fluctuations influence the battery's State of Health (SoH). This leads to more accurate and reliable battery predictions.

The components are described below:

1. Self-Attention Mechanism: Transformers are able to discern which words in a sentence are most significant to each other as a result of this. In order to accomplish this, a scaled dot-product attention method is utilized. All of the words in a sequence are assigned to 3 different vectors.: Query (q), Key (k), Value (v).

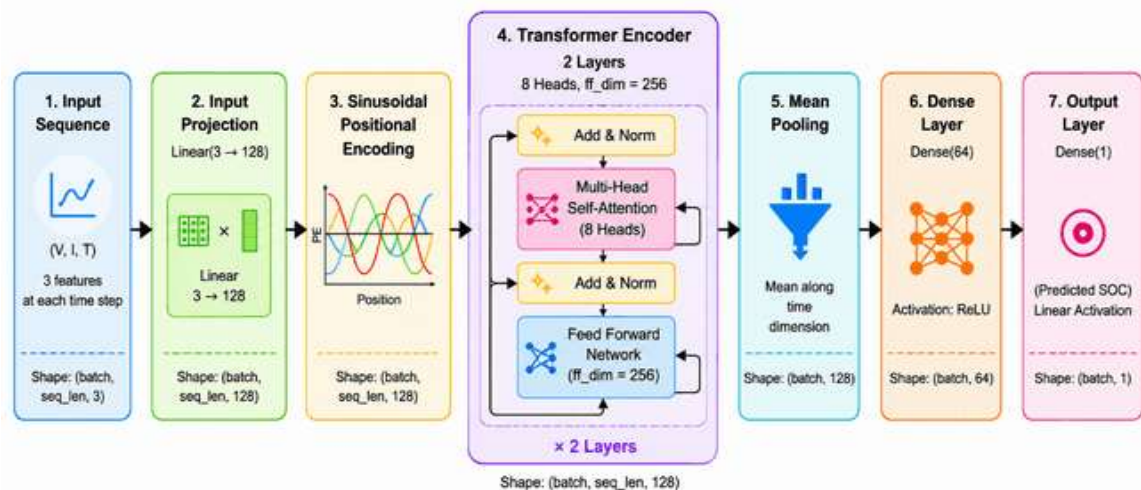


Figure 4. Transformer Regressor architecture

Attention scores are determined as:

$$Attention(Q, K, V) = softmax \left(\frac{qk^T}{\sqrt{d_k}} \right) v \quad (5)$$

q - current state;

k - label for all previous states in the battery;

v - actual information contained in those states;

qk^T - computes similarity among the query and all keys;

$\sqrt{d_k}$ - dimensionality of keys;

softmax - converts the similarities into weights between 0 and 1.

2. Multi-Head Attention: In Transformers, there are multiple attention heads running in parallel. By combining all the heads, the model gains more detailed understanding of the sequence. In a BMS, it allows the model to simultaneously focus on voltage trends, temperature spikes, and current fluctuations, improving its ability to accurately predict the battery's state.

3. Positional Encoding: This ensures the model knows the order of the data, it is essential for understanding patterns, such as the sequence of battery dataset. The positional encoding is computed using the following sine and cosine functions,

$$PE(posi, 2j) = \sin \left(\frac{posi}{10000^{2j/d}} \right) \quad (6)$$

$$PE(posi, 2j + 1) = \cos \left(\frac{posi}{10000^{2j/d}} \right) \quad (7)$$

where posi - position of the input element in the array;

j - embedding index; d - embedding dimension.

4. Position-wise Feed-Forward Networks: There are two linear transformations involved, and a ReLU activation is included in it. Each individual point in the sequence receives its own unique application of it. Mathematically:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (8)$$

where W_1 , W_2 – weight matrices, b_1 , b_2 – bias vectors, x – input feature vector

Figure 5. shows the flow chart of Transformer encoder model. The flow starts by loading time series data and then normalizing it using Min-Max scaling so all values fall within a small range, which helps the model learn better. Next, the data is converted into a supervised learning format, meaning past values are used as inputs to anticipate future values. The flow starts by loading time series data and then normalizing it using Min-Max scaling so all values fall within a small range, which helps the model learn better. Next, the data is converted into a supervised learning format, meaning past values are used as inputs to anticipate future values. After that, training and the testing set are separated from the dataset. A Transformer encoder model is then built, where the input sequence (past time steps) is first passed through an input layer and combined with positional encoding so the model understands the order of the data. Inside the Transformer encoder, the data goes through multiple repeated blocks. Each block contains a multi-head attention mechanism that helps the model focus on the most important parts of the sequence, followed by normalization and a feed-forward neural network that further processes the information. These steps are repeated several times to improve learning. Once the model is built, training begins using the training dataset, and it continues until a stopping condition is met. After training, the model is tested on the test dataset, and its performance is evaluated using metrics such as RMSE, MAE, R^2 .

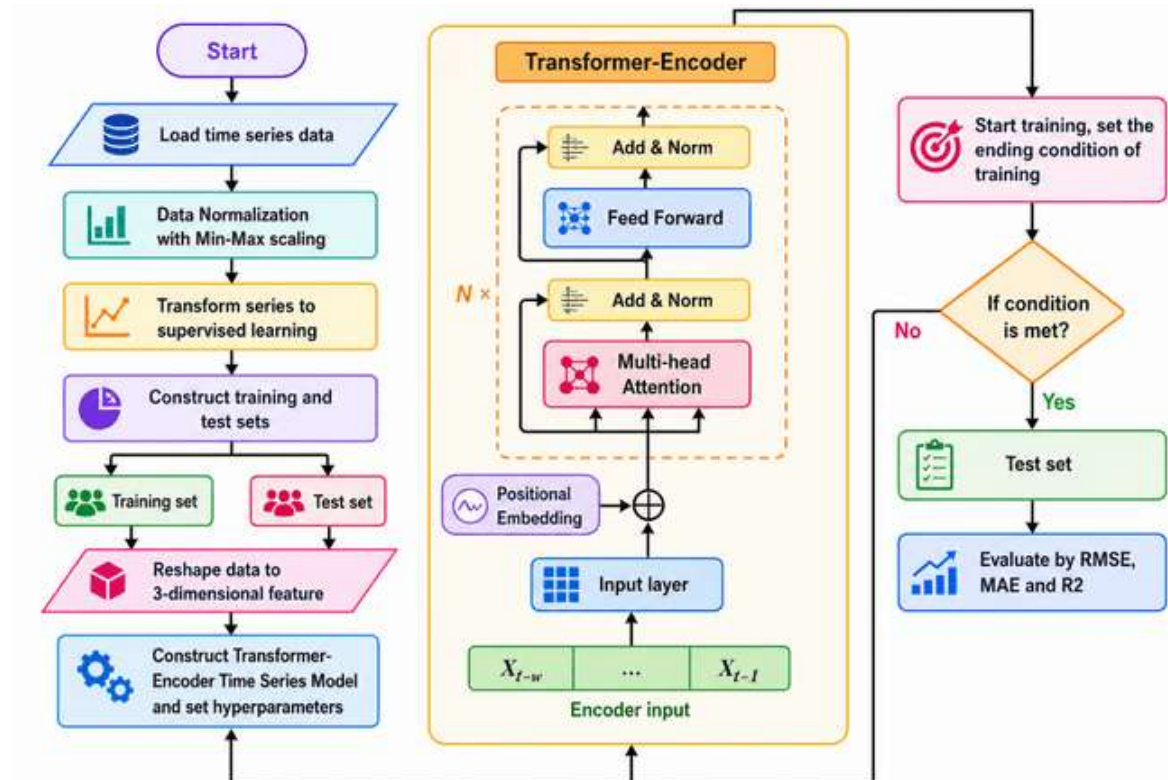


Figure 5. Flow chart of Transformer model

Table 1. HYPERPARAMETERS USED FOR THE MODEL

Feature	Random Forest	SVR	1D-CNN	Transformer
Model Type	Ensemble (Trees)	Kernel Regressor	Neural Network (Conv)	Transformer encoder (Attention)
No. of Trees	30	-	-	-
Max Depth / Layers	6	-	2 Conv, 2 FC layers	2 Encoder Layers
Neurons / Filters	-	50 (Selected Features)	16, 32 (Filters); 16 (Dense)	64 (d model), 128 (FFN), 4 Heads
Learning Rate	-	-	0.001 (AdamW)	0.001 (AdamW)
Activation	-	RBF Kernel	ReLU	GELU & ReLU
Hardware	GPU (cuML)	GPU (cuML)	GPU (PyTorch)	GPU (PyTorch)

5. RESULTS AND DISCUSSION

Table 2. EVALUATION METRICS

Root Mean Square Error (RMSE):	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Mean Absolute Error (MAE)	$MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Coefficient of Determination (R^2 Score)	$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$ $SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ $SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$

where n - Total number of data points; y_i - Actual value of the i-th observation; $\hat{y}_i$ - Predicted value of the i-th observation; SS_{res} - Sum of Squared Residuals;
 SS_{tot} -Total Sum of Squares

5.1 AI MODEL PREDICTION AND PERFORMANCE ANALYSIS

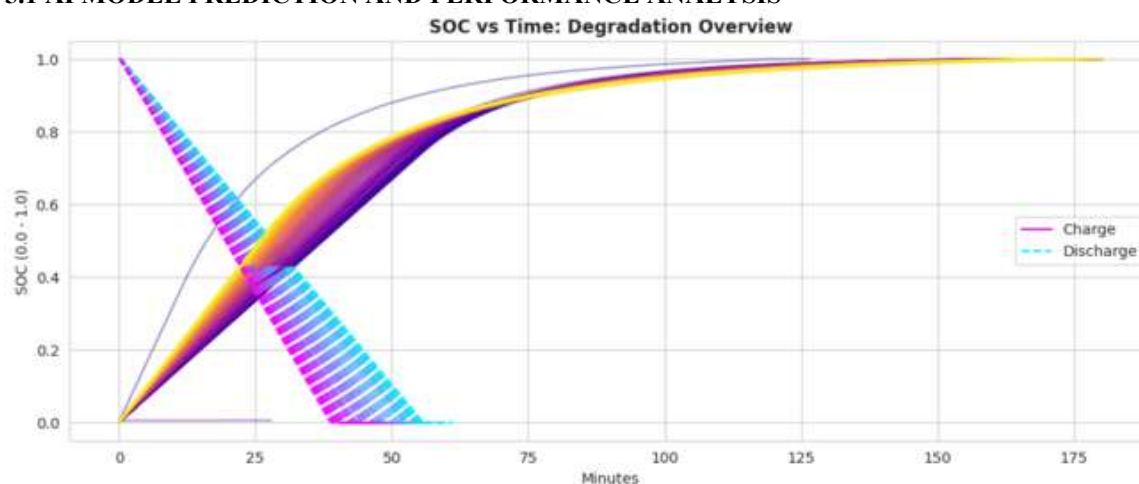
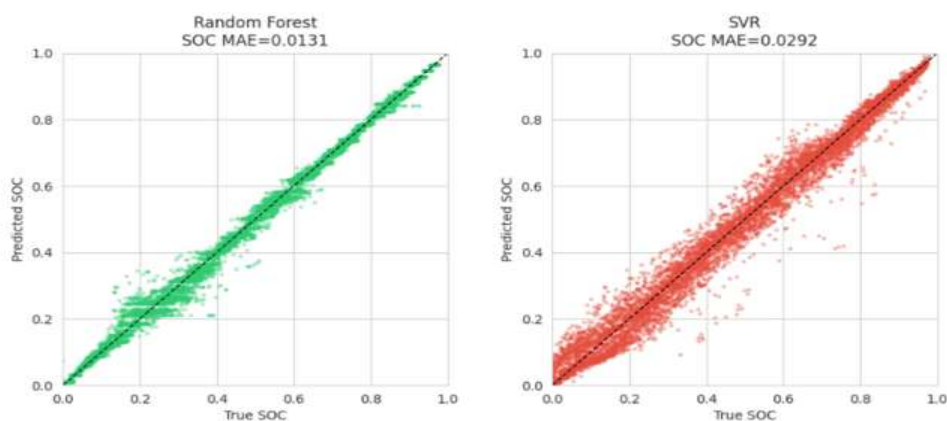


Figure 6. State of Charge Characteristics

The graph in figure 6 shows how the battery charges and discharges over time. The continuous lines represent charging, and the dashed lines represent discharging. At the beginning, the battery takes more time to fully charge and fully discharge. But as the battery is used again and again, the battery is getting weaker over time, so it can store less energy and runs out more quickly. This change in the graph shows how the battery slowly degrades as it ages.



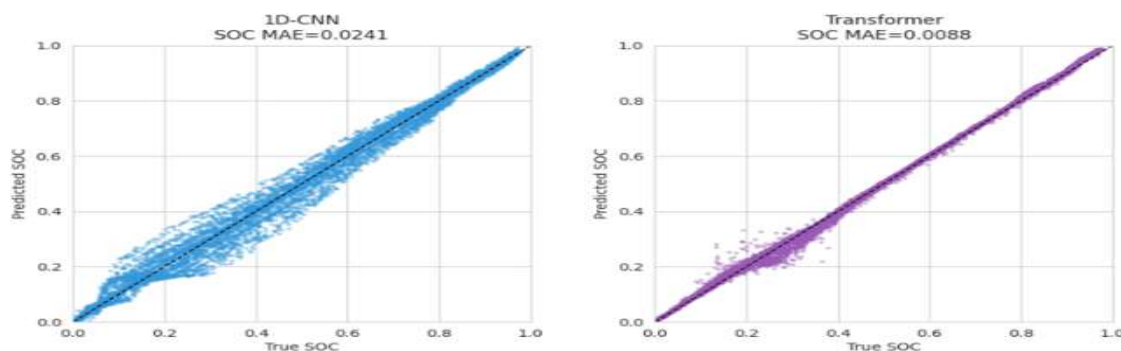


Figure 7. Statistical Correlation of Predicted vs. Ground Truth Values for SOC and SOH Metrics

Figure 8. of scatter plots compares the predicted and the actual values for battery SOC. Random Forest and Transformer models perform the best because their scattered datapoints are closer to the line.

The graph in figure 9. displays better performance of CNN and Transformer models during training. It indicates that the models are learning adequately. The validation error is higher compared to training error. There is no considerable overfitting and the models are generalizing strong. CNN model learns quickly but its improvement slows down, and its validation error stays relatively higher. Transformer model achieves lower training error and shows better performance which shows it is better at learning patterns in the data and gives accurate predictions.

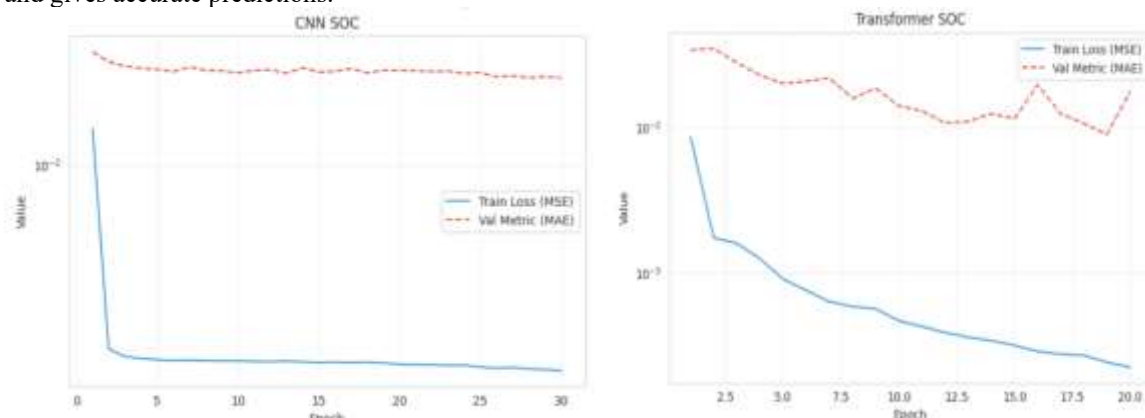
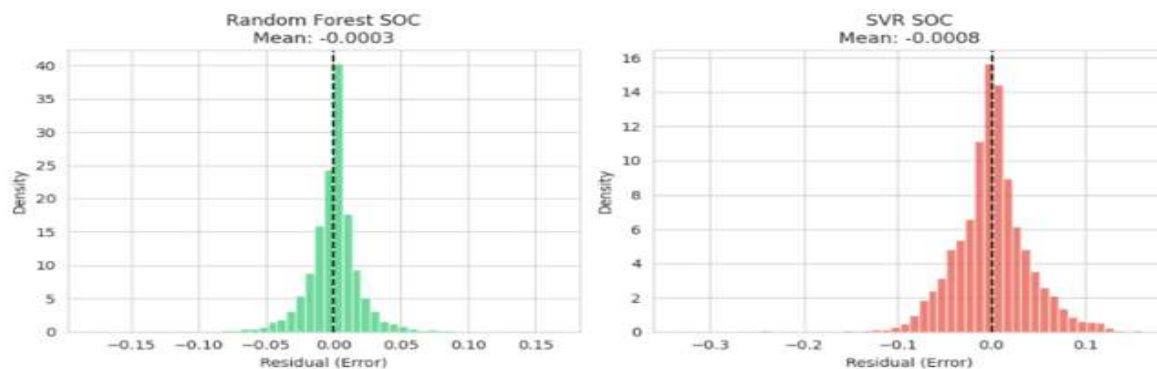


Figure 8. Learning Convergence Analysis: Training vs. Validation Loss across Epochs

Residual Histograms (Predicted – True)

Figure 10. illustrate the degree to which the model's predictions diverge from the values that actually existed. Random Forest and Transformer models have tall and narrow peaks which indicates less errors. SVM and CNN models have wider graphs, that indicates more error.

The figure 11. shows the comparison all the models using metrics: MAE and RMSE. Random Forest and Transformer models perform the best since they have the minimum inaccuracy values (MAE and RMSE) and the best R^2 scores. SVR and 1D-CNN have comparatively higher errors and lower R^2 scores.



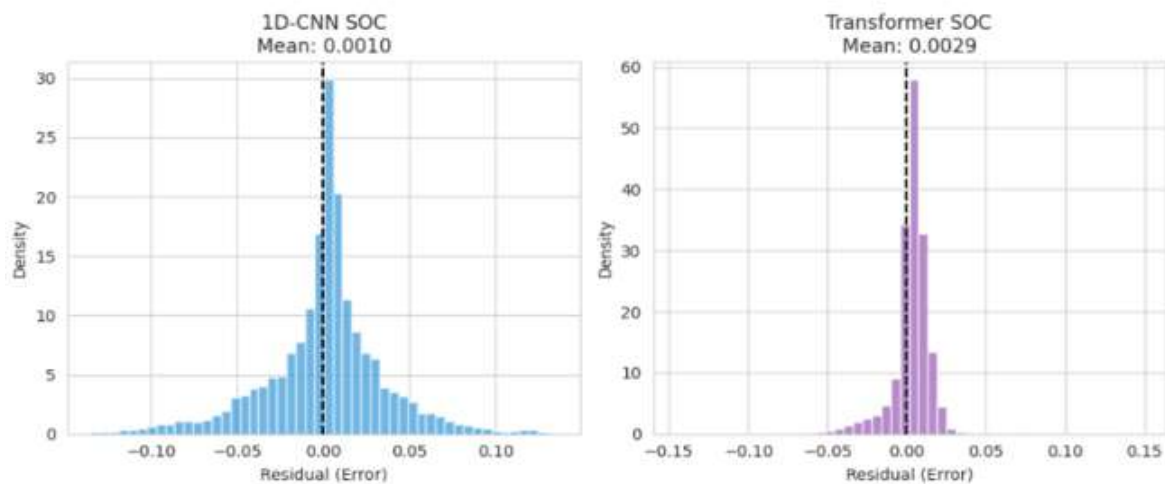


Figure 9. Residual Distribution Analysis: Error Density for SOC Prediction

The comparative results are shown in Table 1.

Table 3. ERROR REPORT

Model	SOC		
	MAE	RMSE	R ²
Random Forest	0.0131	0.0213	0.9951
SVR	0.0292	0.0417	0.9810
CNN	0.0248	0.0357	0.9861
Transformer	0.0088	0.0136	0.9980

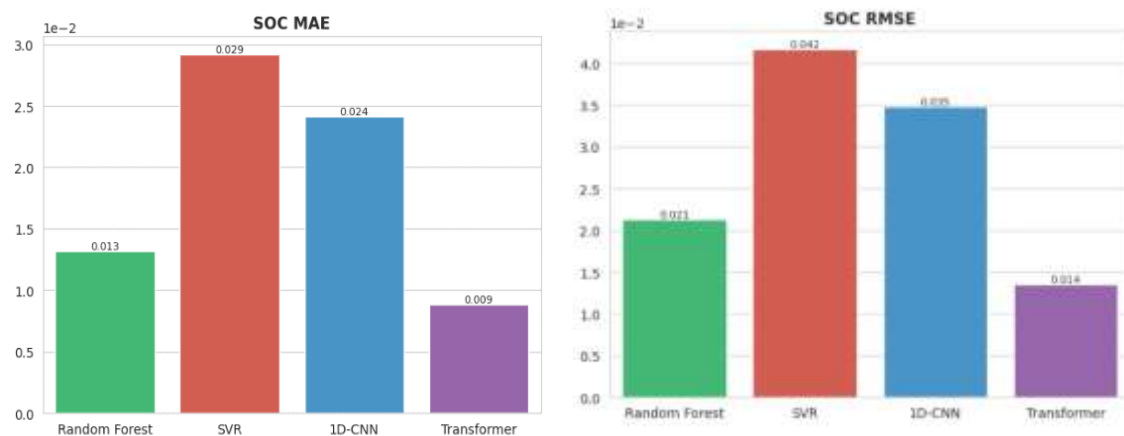


Figure 10. Comparative Performance Metrics

6. CONCLUSION

In this work, simulation experiment is conducted using battery cell B0005 from NASA batter dataset to predict the State-of-Charge using various Artificial Intelligence learning models - Random Forest, Support Vector Machine, 1D Convolutional Neural Network, and a Transformer Regressor model. Transformer Regressor model achieves higher accuracy for SOC estimation, achieving the highest R² score (0.9980). The Random Forest model also performed very well, while SVM and 1D-CNN showed slightly lower performance. Deep Learning architecture, Transformer Regressor, proved to be highly versatile in capturing complex time series battery data. This multi model comparison effectively demonstrated excellent approaches for estimating the level of charge of a battery.

REFERENCE:

- [1] Y. Zheng, Y. Che, X. Hu, X. Sui, D. I. Stroe, and R. Teodorescu, "Thermal state monitoring of lithium-ion batteries: Progress, challenges, and opportunities," *Prog. Energy Combust. Sci.* 100, 101120 (2024).
- [2] I. B. Espedal, A. Jinasena, O. S. Burheim, and J. J. Lamb, "Current trends for state-of-charge (SoC) estimation in lithium-ion battery electric vehicles," *Energies* 14(11), 3284 (2021).
- [3] C. Q. Du, J. B. Shao, D. M. Wu, Z. Ren, Z. Y. Wu, and W. Q. Ren, "Research on co-estimation algorithm of SOC and SOH for lithium-ion batteries in electric vehicles," *Electronics* 11(2), 181 (2022).
- [4] J. Lu, R. Xiong, J. Tian, C. Wang, and F. Sun, "DL to estimate lithium-ion battery SOH without additional degradation experiments," *Nat. Commun.* 14(1), 2760 (2023).
- [5] J. Guirguis and R. Ahmed, "Transformer-based DL models for SOC and SOH estimation of Li-ion batteries: A survey study," *Energies* 17(14), 3502 (2024).
- [6] X. Wang, Z. Hao, Z. Chen, and J. Zhang, "Joint prediction of Li-ion battery SOC and SOH based on the DRSN-CW-LSTM model," *IEEE Access* 11, 70263 (2023).
- [7] Sahhouk, M. A., "State of charge prediction for new and second-life lithium-ion batteries based on the random forest machine learning technique," *International Journal of Power Electronics and Drive System*, 17(1), 487 (2026).
- [8] Anitha, K. and Jerzy, R., "Data-Driven Approach Using Random Forest Regression for Accurate Electric Vehicle Battery State of Health Estimation," *SSRG International Journal of Electrical and Electronics Engineering*, 12(1), 104 (2025).
- [9] Amamra, S.A., "Random forest-based machine learning model design for 21,700/5 Ah lithium cell health prediction using experimental data," *Physchem*, 5(1), 12 (2025).
- [10] Qian, L., Xuan, L. and Chen, J., "Battery SOH estimation based on decision tree and improved support vector machine regression algorithm," *Frontiers in Energy Research*, 11, 1218580 (2023).
- [11] Chen, Z., Sun, M., Shu, X., Xiao, R. and Shen, J., "Online state of health estimation for lithium-ion batteries based on support vector machine," *Applied Sciences*, 8(6), 925 (2018).
- [12] Li, R., Li, W., Zhang, H., Zhou, Y. and Tian, W., "On-line estimation method of lithium-ion battery health status based on PSO-SV," *Frontiers in Energy Research*, 9, 693249 (2021).
- [13] Buvanesan, A.V., Venugopal, B.M., Murugan, C.K. and Velusamy, D.H., "Deep learning for state of health estimation in EV batteries: A CNN-BiLSTM approach," *AIP Advances*, 15(7) (2025).
- [14] Chemali, E., Kollmeyer, P.J., Preindl, M., Fahmy, Y. and Emadi, A., "A convolutional neural network approach for estimation of li-ion battery state of health from charge profiles," *Energies*, 15(3), 1185 (2022).
- [15] Harinarayanan, J. and Balamurugan, P., "SOC estimation for a lithium-ion pouch cell using machine learning under different load profiles," *Scientific Reports*, 15(1), 18091 (2025).
- [16] Zhou, Y., Wang, S., Xie, Y., Zhu, T. and Fernandez, C., "An improved particle swarm optimization-least squares support vector machine-unscented Kalman filtering algorithm on SOC estimation of lithium-ion battery," *International journal of green energy*, 21(2), 376 (2024).
- [17] Sutanto, E., Astawa, P.E., Fahmi, F., Hamid, M.I., Yazid, M., Shalannanda, W. and Aziz, M., "Lithium-ion battery state-of-charge estimation from the voltage discharge profile using gradient vector and support vector machine," *Energies*, 16(3), 1083 (2023).
- [18] Zheng, M. and Luo, X., "Joint estimation of state of charge (SOC) and state of health (SOH) for lithium-ion batteries using support vector machine (SVM), convolutional neural network (CNN) and long sort term memory network (LSTM) models," *International Journal of Electrochemical Science*, 19(9), 100747 (2024).
- [19] Gong, X., Jiang, T., Zou, B., Wang, H., Yang, K., Liu, X., Ma, B. and Lin, J., "SOC estimation of a lithium-ion battery at low temperatures based on a CNN-transformer and SRUKF," *Batteries*, 10(12), 426 (2024).
- [20] Guirguis, J., Abdulkaksoud, A., Ismail, M., Kollmeyer, P.J. and Ahmed, R., "Transformer-based deep learning strategies for lithium-ion batteries SOX estimation using regular and inverted embedding," *IEEE Access*, 12, 167108 (2024).