Performance Comparison of FCN, LSTM and GRU for State of Charge Estimation

RamPrakash S
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
ramprakash3232@gmail.com

Sivraj P.
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
p_sivraj@cb.amrita.edu

***Abstract*—The longevity and safety of lithium-ion (Li-ion) batteries depend on accurate State of Charge (SoC), a key and vital characteristic in battery management systems (BMS) of electric vehicles (EV). This work involves implementing and comparing the performance of Fully Convoluted Neural Network (FCN), Long Short-term Memory (LSTM) and Gated Recurrent Unit (GRU) for SoC estimation. The models for SoC estimation, based on these algorithms, are applied to the US06 Highway driving Schedule, the eVTOL battery dataset and the BMW i3 dataset from IEEE Dataport. The models are trained to predict the SoC when voltage, current and temperature are given as inputs using Jupyter Notebook and libraries like Keras and Tensorflow. The effectiveness of the models is assessed using quantitative performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), time and memory consumed by the models to train and predict the SoC. The GRU model shows better results with an RMSE score of 0.0013 and MAE score of 0.009 compared to the LSTM and FCN model. The work can be further extended by implementing and analysing more models and more battery profiles facilitating an easier selection of best suited model for estimation of SoC.**

Keywords—Battery Management Systems, State of Charge, Neural Networks, Deep Learning, Electric Vehicles.

I. INTRODUCTION

The automotive industry is passing through a critical phase where most auto-makers are shifting from internal combustion (IC) engines to hybrid and battery powered electric vehicles (EV) [1]. Owing to their high energy density, life and efficiency, Lithium-ion (Li-ion) batteries are the most commonly used batteries in electric vehicles and their composition may vary in different electric vehicles depending upon various requirements [2].

Individual battery cells connected in series or parallel and placed within a frame, to protect it from external forces like vibration, heat, etc., are called as battery modules [1]. Battery packs contain lots of individual battery modules with hundreds of cells which require constant monitoring [1]. Battery management systems (BMS) are electronic control units that monitor the current, voltage and the state of charge (SoC) of the batteries and regulate the charging and discharging, optimize the energy and power output provided by the battery pack [3]. It also helps in protecting the battery pack, prolonging its life and keeping the battery operating in within safety limits [3].

Artificial intelligence (AI) has contributed a lot to the recent advancements made in the automotive industry, particularly in areas like estimation algorithms for various parameters in BMS [3][4]. Data driven approaches have shown better accuracy with respect to SoC estimation than the traditional methods used [5][6][7]. Now many researches are working on adapting and commercialization of deep neural network models for accurate and estimation of SoC for BMS [8]. The use of artificial neural networks (ANN) can provide SoC estimation under any condition. Algorithms like multi-layer perceptron (MLP) and recurrent neural networks (RNN) have the capability of predicting battery states with outstanding accuracy [8]. Intelligent computation techniques such as ANN and Kalman filter (KF) is being employed for a lot of battery applications in EV [9]. Compared to other techniques, they have several advantages such as high accuracy, real-time calculation, simple current and voltage measurements, etc. [9].

The objective of this work is to review the performance of algorithms like long short-term memory (LSTM), gated recurrent unit (GRU) and fully convoluted neural network (FCN) for SoC estimation for multiple datasets. The models are implemented using python. Mean absolute error (MAE) and root mean squared error (RMSE), time and memory are the performance metrics that are used to compare the performance of the algorithm.

The next session gives an overview of the existing methods and also the datasets used. Section 3 presents the overview of the system, methods adopted, data pre-processing, and section 4 describes the architecture and implementation of the FCN, LSTM, GRU algorithms. Section 5 brings out the results and discussion and the paper is concluded in section 6.

II. Literature Review

Lithium-ion batteries are the source of power storage technology for EVs due to its high denisty in volume, increased efficiency, and better life cycle [3]. Lithium-ion batteries largely consists of four components cathode, anode electrolyte and separator. The capacity and voltage of a battery is determined by the cathode, and electrons are sent through a wire by the anode. The electrolyte is the medium which helps in the movement of ions, the separator prevents the anode and cathode from contacting [1]. In the context of an electric vehicle, the varying battery capacity across different types, real time data analysis is essential for maintaining battery lifecycle [1]. Fault detection of the battery module is also a primary function of the BMS for lifecycle maintenance. . This is done by measuring thermal runaway, SoC, drive range, etc. BMS is a component which is responsible for monitoring the various parameters of the battery where SoC is one of the important parameters to be checked in real-time.

SoC is the amount of charge that is leftover in a battery compared to its capacity [10]. The SoC can be interpreted as the propotion of available capacity Q at time t, to the nominal capacity Q_n [10].

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

Direct estimation approach and model-based approach are the two types of approaches that are developed for state of charge estimation. Coulomb counting is the most commonly used direct estimation method where the charging and discharging of current is integral over time period. The illustration of SoC is provided by [11],

$$SoC(t+1) = SoC(t) + \frac{\int_t^{t+\tau} I dt}{Q_{Rated}} \tag{2}$$

Where, the initial SoC is denoted by SoC(t), the symbol ‘I’ represent charging or discharging of current in amperes (A), and Q_{Rated} reflects the battery's nominal capacity, expressed in Ampere Hours (Ah) [11].

Even though the coulomb counting method is a widely used and common method to estimate the SoC using current, it is not very accurate [12]. Voltage based measurement of SoC is one method to estimate the SoC but it isn’t dependable due to the flat plateau of discharge characteristics [12]. Model based approaches uses simulation techniques where the battery models are simulated and to measure battery SoC a nonlinear state equation along with an adaptive filter is used [12].

M.A. Hannan proposes a novel deep fully convoluted network where their architecture is able to outperform LSTM and GRU for the 18650OF Panasonic dataset with an RMSE of 0.57 and MAE 0.45 [2]. In recent studies it is shown that the mist recurrent neural networks shows better results for temporal data [2]. The work done by Kailong Liu et al. reviews all the technologies employed in battery model design, SoC estimation, SoH estimation with a detailed explanation of traditional and conventional methods and also system model approach of BMS [3].

Van Quan Dao et al. [13] have proposed a system combining ANN and extended Kalman filter (EKF). The results produced by these two algorithms individually shows errors up to 2.4% and 2.8%, where the inputs to EKF are voltage and current and the output is SoC. This output is fed to the ANN Model along with other parameters like current, voltage and temperature as input and when validating the results on a Lithium-ion battery the error was found to be less than 1% [14]. The work by K.L Man and T. Krilavičius [14] also reviews the KF and ANN usage in the estimation of SoC, and the authors suggest that the future of SoC and other estimations of BMS would be achieved using AI [14].

Wei Hi and Nicholas Willard states the use of feed forward neural network for SoC estimation, where EKF was used to eliminate errors obtained from the neural network algorithm. By using EKF, RMSE between 2.5%-3.5% was achieved eliminating the need for open circuit voltage [11]. Dona George et al. proposes a regression based neural network, where an EV powertrain model is simulated using MATLAB and with the data acquired a BI-LSTM model is implemented which gives a RMSE of 0.029 [15]. The work done by A. Rastegarpanah et al. uses extracted data from the 13 battery pack modules of Nissan Leaf 2011 model, where the RMSE and MAE have average values of 1.790% and 1.2006% respectively [16].

Based on the literature survey it is evident that the recent advances in AI can give better SoC estimation results on variety of benchmarks. However, most of the research done is on proprietary data. Open-source datasets like eVTOL and IEEE BMW i3 dataset are not widely utilized for evaluating the algorithms specifically for SoC estimation. Many intelligent models have been widely adopted for Soc estimation. Diagnosing the performance of each algorithm based on implementation of multiple datasets might be the well-reasoned approach. Hence, in this paper we compare the performance of recurrent and convolution neural network on multiple datasets.

III. System Overview

A. System Overview

Figure 1 shows the overview of the system considered for estimating the SoC which consists of the battery datasets that would be fed to the neural network algorithms and which would give the output as estimated SoC. The implementation of the models was done using python, with jupyter notebook as the development environment. Keras and Tensorflow were the libraries used for the implementation of the model, MinMaxScaler for normalizing the datasets and matplotlib is used for plotting the loss and accuracy. The inbuilt model.evaluate function from the keras library was used to evaluate the model to provide average loss and accuracy.

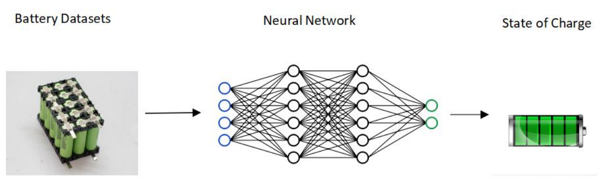

Fig. 1. System overview

B. Methodology

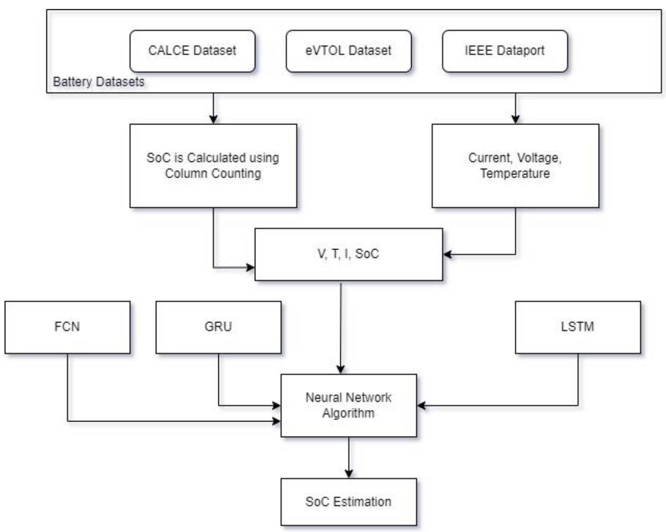

Fig. 2. Work flow of implementation of models for performance comparison

Figure 2 shows the flow of work for the implementation of all the three models and comparison of their performances. The battery datasets, available online [17][18][19], contains the input data for the neural network. Current, temperature and voltage are the parameters that are present in the dataset. The SoC is not readily available for the eVTOL and CALCE dataset and has to be calculated mathematically.

C. Data Pre-processing

Many battery datasets with different battery profiles, current and voltage profiles and also temperature are tested and are made available online in the public domain [12]. CALCE dataset is one of widely used datasets, which is tested with profile US06 highway driving schedule current profile with ambient temperature being at 45°C having 2000mAh capacity, with LiNiMnCo (Lithium Manganese Cobalt Oxide) cell chemistry. [17]. The dataset consists of 11446x19 rows and columns with 19 attributes like step_time, voltage, current, SoC, charge capacity, discharge capacity, charge energy and discharge energy. Second dataset is the eVTOL battery dataset from Carnegie Mellon University which was a Lithium-Ion battery from Sony energy with 3000mAh capacity with a 3.7V nominal capacity tested at 30°C for battery discharge at take-off, landing and cruise modes of an aircraft. The dataset is 385430x12 rows and columns with 12 attributes like current, voltage, temperature, time delta [18]. Third dataset which was used is the battery and heating data in real drive cycles from IEEE Dataport where a BMW i3 was tested with 72 drive cycles and the data was recorded.The dataset has 10091x28 rows and columns with 28 attributes where some of them are current, temperature, voltage, torque, throttle, velocity.The batteries were at an ambient temperature of 25°C. [19].

Data wrangling, a type of transformation and mapping from raw format to a format suitable for our application, is performed as a pre-processing step where the columns which are empty are eliminated and processed. The datasets are normalized using MinMaxscaler to get the data within the range of 0-1 prior to training. Some of the samples in the data which would create an imbalance and are corrupt are excluded. The dataset is visualized using python with the help of matplotlib and pandas libraries. Figure 3 gives the visualization of the dataset using matplotlib in a 3D view.

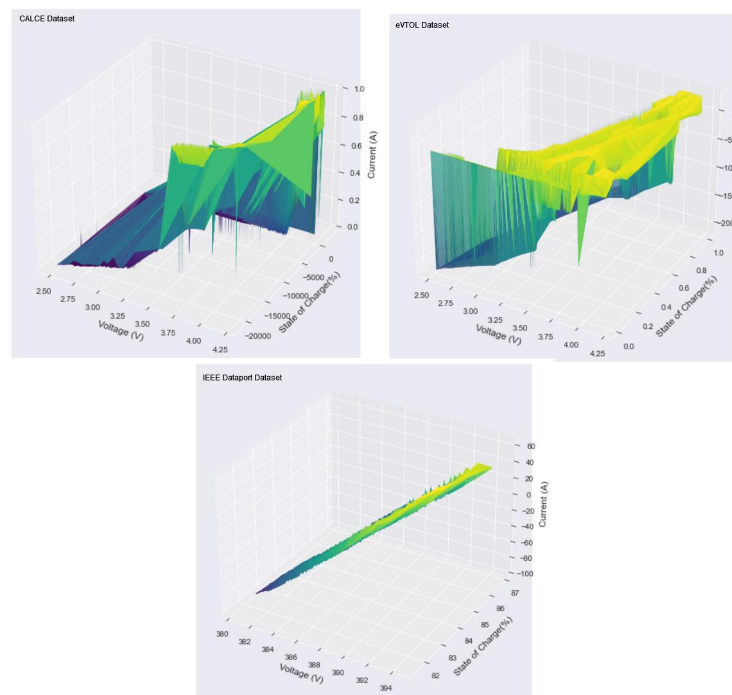

Fig. 3. Visualization of datasets using matplotlib

IV. Implementation

A. Calculating SoC

The SoC is calculated using equation (2), following the Columb counting method, by integrating the active flow of current over time to derive the energy of the battery pack for the battery datasets where SoC in not readily available. The Coulomb counting formula is implemented as a function in the spreadsheet for manually estimating the SoC.

B. FCN, LSTM and GRU model Architecture

Convolutional neural networks (CNNs) are mainly used in image processing and computer vision applications as they are capable of processing complex and diverse datapoints. Here, first, a fully convoluted neural network (FCN) is implemented to estimate the SoC. Voltage, current, temperature and SoC as inputs is passed to the FCN model with the architecture consisting of four convolutional kernels with kernel width as w = [5,3,3,1] and the amount of kernels, n = [8,16,8,1]. A batch normalizing layer follows each convolutional layer followed by the Mish activation function [20].

$$\text{Mish} = x * \tanh(\ln(1+e^{x})) \quad (3)$$

Mish activation function is similar to the Swish activation function. Mish provides strong regularization and better results compared to Swish and Rectified Linear Unit (ReLU) as it has non-monotonic smooth curve preserving a small amount of negative [20]. L2 regularization method also called as Ridge regression is used to reduce overfitting where the coefficient squared magnitude is added to the loss function as a penalty term, followed by a Global Average Pooling layer (GAP). To get the outcome Instead of a dense layer, a ReLU layer follows the GAP layer. The dataset was split and 80% was used as training data and 20% as test data. The model is then compiled with MAE which is the mean of the absolute values of each prediction error over all occurrences of the test data set, for loss. It's optimized using Adam optimizer with 0,001 as learning rate as it's computationally efficient and little memory is needed. With a batch size of 64 the model is trained for 1000 epochs.

LSTM and GRU model with 32 and 36 hidden layers respectively followed by dropout regularization were used to reduce overfitting of the data followed by a ReLU activation function where the output layer is a dense layer. The data was split as 80% train data and 20% test data. Both the models were trained with MAE as loss and Adam optimizer. With a batch size of 32 the model is trained for 50 epochs.

All the models are evaluated using regression metrics MAE and RMSE,

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - x) \quad (4)$$

X represents true value and x_i represents the predicted value and n represents number of data points [2].

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}} \quad (5)$$

Where, Y_i represents the true value of and $\widehat{Y}_i$ represents the predicted value, n is the number of non-missing datapoints it's the most commonly used for regression problems [2].

V. Results

In this section, the results obtained from the implementation of the models described in the previous section on the three datasets are evaluated. Table 1 shows the performance evaluation of the models with regression metrics and Table 2 shows the time and memory consumed by the models.

TABLE I. PERFORMANCE METRICS COMPARISON OF VARIOUS MODELS

Models	Dataset	Metrics		
		Loss	***RMSE (in %)***	***MAE (in %)***
FCN	IEEE	0.04	0.17	0.4
	eVTOL	0.0231	0.0143	0.0091
	CALCE	0.0106	0.016	0.0096
LSTM	IEEE	2.4052	1.3424	1.5509
	eVTOL	0.0032	0.0567	0.0375
	CALCE	0.00018	0.014	0.006
GRU	IEEE	1.76	0.013	0.009
	eVTOL	0.0031	0.0554	0.0398
	CALCE	0.00017	0.0013	0.009

TABLE II. COMPARISON OF TIME AND MEMORY CONSUMED BY THE MODELS

Models	**Dataset**	**Metrics**	
		Time (in minutes)	***Memory(in MB)***
FCN	IEEE	52.8	600
	eVTOL	28	501
	CALCE	65	470
LSTM	IEEE	97	577
	eVTOL	15	593
	CALCE	50.5	401
GRU	IEEE	85	577
	eVTOL	14	539
	CALCE	50.65	439

Table 1 shows that all the three models have almost similar performance, and the variation of error between the models is within the range of 1% to 2% for the CALCE and the eVTOL dataset. The values indicate that LSTM and GRU perform better compared to the FCN for these datasets. Subsequently the FCN model for the IEEE dataset shows considerable amount of loss where the error seems to be more compared to the other two datasets. Table 2 shows the time and memory consumed by each model when they were trained on an AMD Ryzen 7 3750H 2.30GHz device.

Loss function is used to estimate how far the estimated value is compared to the true value. The FCN model shows an average accuracy of above 94% while the LSTM and GRU models show an average accuracy of 97% and 98% respectively.

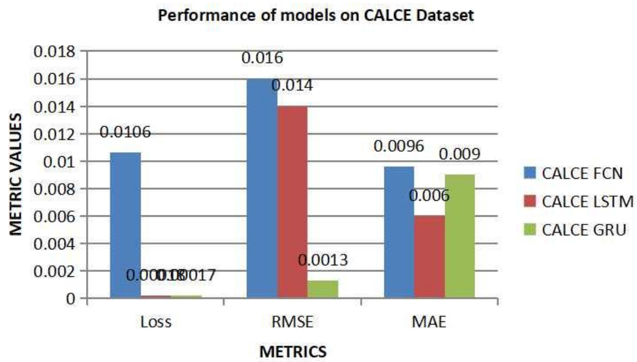

Fig. 4. Performance of models in CALCE dataset

Figure 4 shows the comparison of CALCE dataset performance in graphical representation with different metrics and models, where we can see the loss for the FCN is on the higher side compared to LSTM and GRU, The RMSE values and MAE shows that GRU performs better for the CALCE dataset when compared. As compared to LSTM and FCN, GRU performs better because the architecture of GRU is such that it has two gates; reset and update.

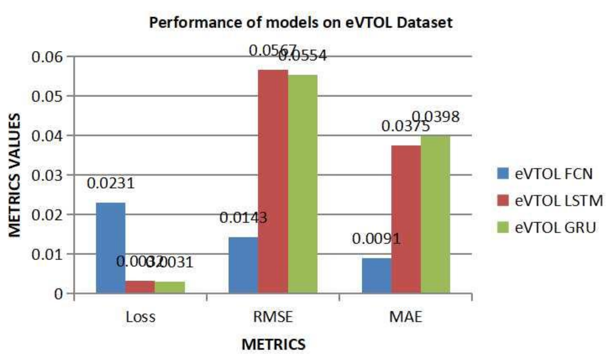

Fig. 5. Performance of models in eVTOL dataset

Figure 5 shows the performance of the models on the eVTOL dataset, where we can see the loss is higher for the FCN. GRU is observed to be performing better compared to other models with this dataset due to dropout combined with dense layer output.

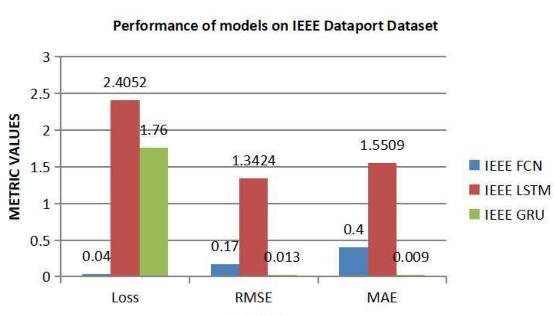

Fig. 6. Performance of models in IEEE dataport dataset

Figure 6 shows the performance of models on IEEE dataset, where it can be seen that the loss is lower for FCN. GRU gives better results compared to other models. It is evident, as seen from figures 4, 5 & 6, that GRU performs better compared to LSTM and FCN because GRU as an architecture itself help in retaining long-term historical data due to the reduced gating signals, where the update gate z_t and a reset gate r_t are represented by,

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{6}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{7}$$

For a hidden state of size n and an input of size m, W is a n × m matrix, U is a n× n matrix, and b is a n × 1 matrix (or vector). [21].

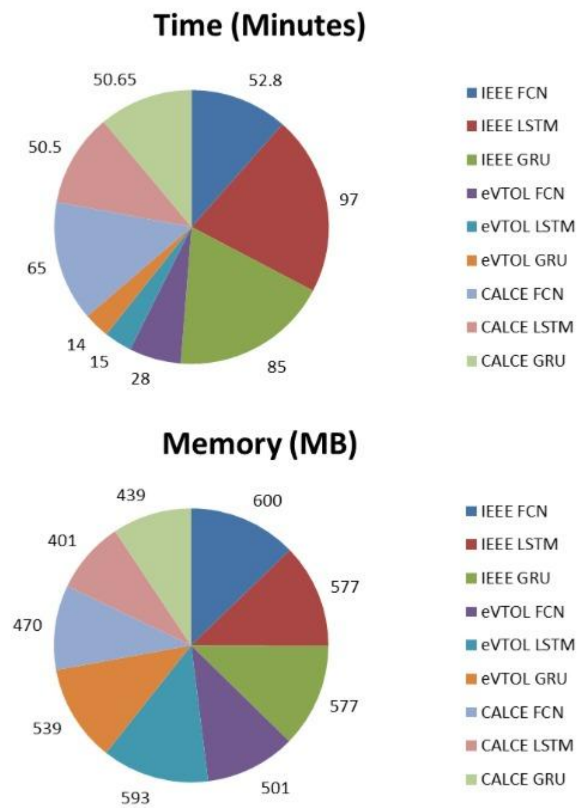

Fig. 7. Time And Memory Consumption of the Models

Figure 7 shows the memory and time consumption of different models to different datasets where we can see the FCN consumes more time and data for fitting and learning the datasets compared to LSTM and GRU. The time and memory consumed also shows that there's a 2% to 5% difference in both LSTM and GRU models when compared with the FCN, because the computational cost is higher compared to LSTM and GRU as the FCN model consumes more memory for these specific datasets.

VI. CONCLUSION

This paper presents the implementation of three models FCN, LSTM and GRU for three different datasets CALCE, eVTOL, IEEE dataport. The SoC is calculated using Coulomb counting using a new activation function called mish. It is observed that GRU performs better with the RMSE of around 0.13-0.16 for the CALCE dataset, 0.05-0.01 for the eVTOL dataset and 0.009-0.005 for the IEEE dataport dataset. GRU also has less memory footprint. This work can be further extended to different battery cycles and chemistries, using modified algorithms, different optimizers and activation functions for the current and different battery profiles, and also considering important and different scenarios like thermal runaway.

REFERENCES

[1] "The Composition of EV Batteries: Cells? Modules? Packs? Let's Understand Properly!", Samsungsdi.com, 2022. [Online]. Available: https://www.samsungsdi.com/column/all/detail/54344.html. [Accessed: 12- May- 2022].

[2] M. Hannan et al., D. N. T. How, P. J. Ker, Z. Y. Dong, M. Mansur and F. Blaabjerg "SOC Estimation of Li-ion Batteries With Learning Rate-Optimized Deep Fully Convolutional Network", IEEE Transactions on Power Electronics, vol. 36, no. 7, pp. 7349-7353, 2021. Available: 10.1109/tpel.2020.3041876.

[3] K. Liu, K. Li, Q. Peng and C. Zhang, "A brief review on key technologies in the battery management system of electric vehicles", Frontiers of Mechanical Engineering, vol. 14, no. 1, pp. 47-64, 2018. Available: 10.1007/s11465-018-0516-8.

[4] H. Gabbar, A. Othman and M. Abdussami, "Review of Battery Management Systems (BMS) Development and Industrial Standards", Technologies, vol. 9, no. 2, p. 28, 2021. Available: 10.3390/technologies9020028.

[5] T. Duraisamy and D. Kaliyaperumal, "Active cell balancing for electric vehicle battery management system", International Journal of Power Electronics and Drive Systems (IJPEDS), vol. 11, no. 2, p. 571, 2020. Available: 10.11591/ijpeds.v11.i2.pp571-579.

[6] T. Biya and M. Sindhu, "Design and Power Management of Solar Powered Electric Vehicle Charging Station with Energy Storage System", 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA*)*, 2019. Available: 10.1109/iceca.2019.8821896 [Accessed 21 June 2022].

[7] S. Feraco, P. Anselma, A. Bonfitto and P. Kollmeyer, "Robust Data-Driven Battery State of Charge Estimation for Hybrid Electric Vehicles", SAE International Journal of Electrified Vehicles, vol. 11, no. 2, 2021. Available: 10.4271/14-11-02-0017.

[8] V. Chandran, C. Patil, A. Karthick, D. Ganeshaperumal, R. Rahim and A. Ghosh, "State of Charge Estimation of Lithium-Ion Battery for Electric Vehicles Using Machine Learning Algorithms", World Electric Vehicle Journal, vol. 12, no. 1, p. 38, 2021. Available: 10.3390/wevj12010038.

[9] Lim, D., Ahn, J., Kim, D. and Lee, B., 2016. A Mixed SOC Estimation Algorithm with High Accuracy in Various Driving Patterns of EVs. Journal of Power Electronics, 16(1), pp.27-3

[10] G. Almeida, A. Souza and P. Ribeiro, "A Neural Network Application for a Lithium-Ion Battery Pack State-of-Charge Estimator with Enhanced Accuracy", Proceedings, vol. 58, no. 1, p. 33, 2020. Available: 10.3390/wef-06915.

[11] W. He, N. Williard, C. Chen and M. Pecht, "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation", International Journal of Electrical Power & Energy Systems, vol. 62, pp. 783-791, 2014. Available: 10.1016/j.ijepes.2014.04.059.

[12] P. Tejaswini and P. Sivraj, "Artificial Intelligence based State of Charge estimation of Li-ion battery for EV applications", Proceedings of the Fifth International Conference on Communication and Electronics Systems, 2021. [Accessed 21 December 2021].

[13] V. Dao et al., "Design of an Effective State of Charge Estimation Method for a Lithium-Ion Battery Pack Using Extended Kalman Filter and Artificial Neural Network", Energies, vol. 14, no. 9, p. 2634, 2021. Available: 10.3390/en14092634.

[14] K. Man, C. Chen, T. Ting, T. Krilavičius, J. Chang and S. Poon, "Artificial Intelligent Approach To SoC Estimation For Smart BMS", Available:https://www.researchgate.net/publication/289666334,. [Accessed 6 November 2021].

[15] D. George and P. Sivraj, "Driving Range Estimation of Electric Vehicles using Deep Learning", International Conference on Electronics and Sustainable Communication Systems, 2021. [Accessed 22 March 2022].

[16] A. Rastegarpanah, J. Hathaway, M. Ahmeid, S. Lambert, A. Walton and R. Stolkin, "A rapid neural network–based state of health estimation scheme for screening of end of life electric vehicle batteries", Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systes and Control Engineering, vol. 235, no. 3, pp. 330-346, 2020. Available: 10.1177/0959651820953254.

[17] Calce Battery Research group, Data, https://web.calce.umd.edu/batteries/data.htm, last accessed, 2019/12/19.

[18] A. Bills, V. Viswanathan, S. Sripad, E. Frank, D. Charlesand W. L. Fredericks, "eVTOL Battery Dataset". Carnegie Mellon University, 18-Mar-2021, doi: 10.1184/R1/14226830.v1.

[19] Matthias Steinstraeter, Johannes Buberger, Dimitar Trifonov, October 19, 2020, "Battery and Heating Data in Real Driving Cycles", IEEE Dataport, doi: https://dx.doi.org/10.21227/6jr9-5235.

[20] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function", British Machine Vision Virtual Conference, 2019. Available: https://doi.org/10.48550/arXiv.1908.08681. [Accessed 7 April 2022].

[21] H. layer, "How many parameters are in a gated recurrent unit (GRU) recurrent neural network (RNN) layer?", *Cross* Validated, 2022. [Online].Available:https://stats.stackexchange.com/questions/328926/how-many-parameters-are-in-a-gated-recurrent-unit-gru-recurrent-neural-network. [Accessed: 27- Jun- 2022].