Advanced Mechanical and Mechatronic Systems

Research Article
Volume 2(2), 2026, 106 −121
https://doi.org/10.53623/amms.v2i2.1040

Predictive Modeling of Lithium-Ion Battery During Discharging Stage Using CNN-DenseNet and LSTM

Mukhidin Wartam[1]*, Yogi Reza Ramadhan[2], Hayati Yassin[1], Taufik Taufik[3], Ryan Fitriawan[4]

[1]Faculty of Integrated Technologies, Universiti Brunei Darussalam, Jalan Tungku Link Gadong, Brunei Darussalam
[2]Redesma Technologies, Jalan Bulusan VI No 37, Tembalang, Semarang 50277, Indonesia
[3]Department of Electrical Engineering, Cal Poly State University, San Luis Obispo, USA
[4]PT. PLN Indonesia Power, Jl. Suralaya 21, Suralaya, Kota Cilegon, Banten 42439, Indonesia

*Correspondence: 22m8978@ubd.edu.bn

SUBMITTED: 9 February 2026; REVISED: 20 April 2026; ACCEPTED: 1 May 2026

ABSTRACT: Lithium-ion batteries (LIBs) have become essential to renewable energy technologies, enabling the storage of electricity generated from renewable sources. This study presents an application of Long Short-Term Memory (LSTM) method and CNN-DenseNet to predict the lithium-ion battery during the discharging stage to overcome the crucial problem of battery performance degradation in electric vehicles and energy storage systems. To examine the LSTM model, a specially developed battery monitoring system based on an STM32 microcontroller and a Raspberry Pi 4 microcomputer for data acquisition was used. Testing was done on different discharge patterns at different loads (100W, 130W, 180W, 200W, 220W) on 12V/70Ah lead-acid free-maintenance battery with data collected for 1-10 hours discharge cycles. The monitoring system was designed with voltage sensor, current sensor, and temperature sensor to anticipate disturbances during data capture. The proposed LSTM model achieved superior performance, with RMSE = 0.0847, MSE = 0.0505, and MAE = 0.03548, significantly outperforming the CNN-DenseNet approach (RMSE = 0.3333, MSE = 0.4037, MAE = 0.3051) across diverse testing conditions. Although DenseNet performed best under moderate load (100 W), the LSTM architecture excelled under high load (220 W), underscoring the adaptability of the proposed model across varying operating conditions. This study expands the maintenance framework for energy storage systems through an LSTM architecture offering greater accuracy and stability.

KEYWORDS: Battery prediction; CNN DenseNet, electric vehicle; LSTM.

1. Introduction

Promoting renewable energy worldwide has led lithium-ion batteries (LIBs) to become essential components in various applications, ranging from portable electronics and electric vehicles to grid energy storage [1–3]. As their use has grown, understanding battery degradation has become a critical issue for ensuring efficient performance and maximizing lifetime [1, 4–5]. Although previous research focusing on fundamental modeling concepts has been useful, the complex nature of battery degradation processes requires more advanced modeling techniques [6]. Recent developments have highlighted machine learning frameworks for capacity loss prediction, which

significantly improve the accuracy of degradation models through effective feature selection strategies [7–8]. Recent progress in this field has been driven by machine learning, which provides a strong foundation for modeling complex electrochemical systems [9–10]. Convolutional Neural Networks (CNNs), which are highly effective at capturing hierarchical feature representations, have been used to learn complex mappings between operational conditions and degradation patterns in LIBs [11–13].

Previous studies have reported substantial improvements in predicting battery degradation and developing accurate models; however, current predictive approaches still face limitations in capturing degradation behavior under varying operating conditions. For example, transfer learning approaches have improved early-stage degradation prediction by incorporating physical constraints and leveraging limited observational data [4]. Although DenseNet CNN architectures are highly effective in computer vision tasks due to feature reuse and deep supervision, their application in battery degradation prediction remains limited [2]. This study addresses three key gaps in the existing literature. First, there is a lack of systematic investigation into load-dependent degradation behavior using machine learning approaches, particularly for power levels between 100 W and 220 W and their influence on degradation trajectories. Second, there is an absence of cohesive validation frameworks for machine learning models in battery applications, especially those involving cross-load prediction and reliability assessment. Third, standardized evaluation paradigms for assessing model robustness under varying operational conditions in battery degradation modeling remain underdeveloped.

This study implements LSTM and DenseNet CNN architectures using a multi-stream input approach, processing voltage curves, load profiles, and temporal degradation indicators through parallel dense blocks. The novelty of this work lies in its systematic analysis of load-dependent battery degradation, the development of specialized validation metrics, and the integration of uncertainty quantification within the prediction pipeline. This investigation aims to establish new benchmarks for prediction accuracy across diverse operational scenarios while providing insights into both short-term performance impacts and long-term degradation trajectories.

2. Methods

2.1.1. CNN DenseNet.

DenseNet, introduced by Huang et al. [14–15], is a deep learning architecture designed to address two major challenges in deep neural networks: limited feature reuse and the vanishing gradient problem. The authors proposed an enhanced CNN framework by introducing dense connections, in which each layer is connected to every other layer in a feed-forward manner. In DenseNet, feature maps from all preceding layers are concatenated and passed as inputs to subsequent layers. This dense connectivity promotes efficient feature reuse and reduces the number of parameters, as features do not need to be relearned by later layers. In the proposed framework for regression tasks, DenseNet's principles of feature reuse and concatenation are adopted, with modifications to better accommodate nonlinear regression problems. The concatenation-based design improves training efficiency and enhances performance by ensuring that encoded features are fully reused rather than redundantly recomputed. DenseNet is particularly suitable for tasks involving complex pattern recognition and dependency learning, such as incremental capacity analysis (ICA) curves in battery systems. Its integration into this framework enables more efficient prediction with fewer parameters compared to conventional CNN models, while also mitigating issues such as vanishing gradients.

2.1.2. DenseNet architecture for discharging lithium-ion battery prediction.

This work proposed a CNN with DenseNet architecture for the prediction of discharge voltage, as shown in Figure 1. The particular input layer of the applied model is the current derived from the lithium-ion battery. The following data will then be used as input to the CNN. In this regard, current data is a scalar signal utilized for forecasting voltage.

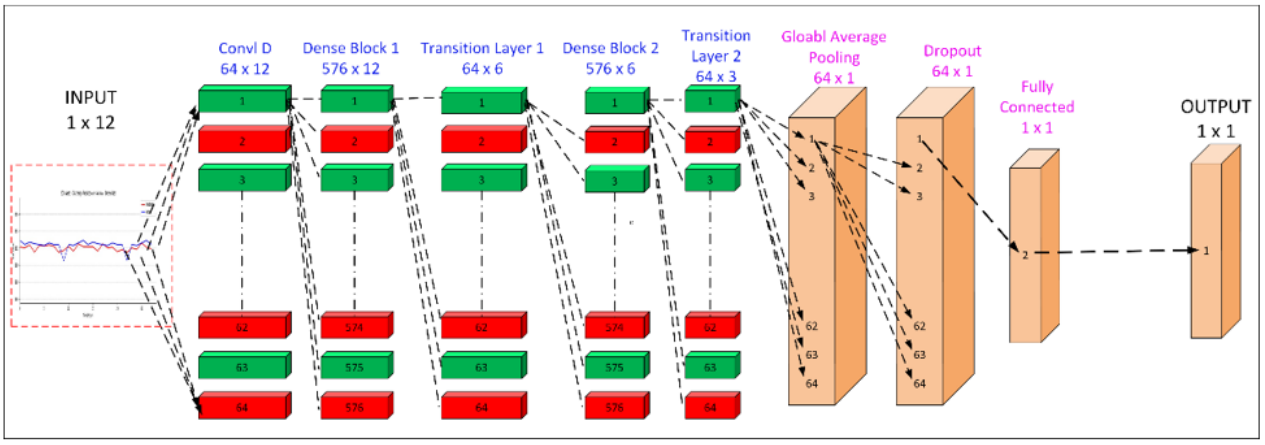

Figure 1. DenseNet regression architecture for battery degradation prediction.

In Dense Block 1 section, after accepting the input, it moves to next dense block 1 which has some set of convolutional layers which takes input from all subsequent layer in the block; that means each layer of DB 1 takes input from all the layer available above that layer in the same block. The dense block takes a job of feature extraction from the input data by carrying out consecutive processes of convolution and is connected to the transition layer. The transition layer also helps to change the dimension of the feature data produced by the dense block ahead of its integration with the dense block 2. After that, dense block 2's task is to continue the feature extraction from the perceived data, which have been processed by the previous dense block 2 and transition layer 1. Then, transition layer 2 is used to alter the dimensionality of the feature data which is generated by the dense block 2. The second transition layer includes batch normalization layers ReLu activation, and pooling. These layers are proposed to serve to make batches normalized with the ReLu activation function and to decrease the feature data dimensionality, which is from dense block 2 and will be followed by the Global Average Pooling.

For down sampling the feature data, reduction through global average pooling is used in order to calculate the average of all values of feature maps. The goal of global average pooling, as the name implies is thus to take the features and reduce them to some fundamental one- dimensional format while preserving some of the features which is going to be useful when we are making out predictions. The dropout layer is used to help prevent overfitting whereby, during the training session, those particular neurons are turned off randomly and will connect to the fully connected layer. In this layer, the fully connected layer wants to connect all neurons from the previous layer with each neuron in this layer. This fully connected layer is connected for doing regression tasks including features extracted by the previous layers then connecting to the next layer. In this output layer, it will create the final diagnosis, which is the diagnosis of the voltage of the discharging process of a lithium-ion battery that has been discharged. The formula that was used in the construction of the CNN DenseNet is presented in Equations (1) – (8).

$$y^{(0)} = f\left(\int_{c=1}^{C_{in}} w_c \times x_c + b\right) \tag{1}$$

Dense block for each l in dense block

$$x^{(l+1)} = ReLu\ (BN([x^{(0)}, x^{(1)}, \dots x^{(l)}])) \quad (2)$$

$$y^{(l+1)} = Conv1D(x^{(l+1)}, w^{(l)}, \dots b^{(l)}) \quad (3)$$

Output dense block

$$[x^{(0)}, x^{(1)}, \dots x^{(L)}] \quad (4)$$

Transition layer

$$z = Conv1D(x, w, b) \quad (5)$$

Global average pooling

$$g = \frac{1}{T}\Sigma_{t=1}^{T} z_t \quad (6)$$

Fully connected layer

$$y = Dropout(g) \quad (7)$$

$$Output = W.y + b \quad (8)$$

where: f is ReLu activation function; w i convolutional weight; $*$ is convolution operation; x is input; b is bias; BN is batch normalization; $[x^{(0)}, x^{(1)}, \dots x^{(l)}]$ is concatenation operation; z is transition layer; g is global average pooling; and T is the length of the sequence after convolution and pooling.

2.1.3. Long Short-Term Memory (LSTM).

Long Short-Term Memory (LSTM) is one of the subtypes of Recurrent Neural Network (RNN) that is highly efficient for data with temporal nature. The use of three gates: input gate, forget gate and output gate happen in LSTM is to determine what should go down the chain and what should be ignored. Each LSTM unit also has a cell state for long term memory path and hidden state for short term information available for the next time step for prediction in the LSTM architecture, voltage sequence data are processed over time to discover the relationship of the data in the context of the battery discharge prediction. This structure allows for the non-linear variations of the battery data in an LSTM [16 - 18].

The LSTM at each step will process the input data based on the cell state and hidden state from the previous step so that at each step it can save or forget the information depending on the requirements of the pattern of prediction. Firstly, this allows the LSTM to realize long-term patterns when data in sequence, e.g., change in voltage over a battery discharging cycle. The final predictions are generated after a series of layers from the LSTM process after multiple layers of information.

The LSTM method is used in many applications, which is due to its ability to deal with data that have temporal dependencies commonly used are in time series prediction, video analysis, and speech recognition. Figure 2. presents a schematic representation of the LSTM architecture, delineating its principal components and illustrating the interconnected data flow pathways.

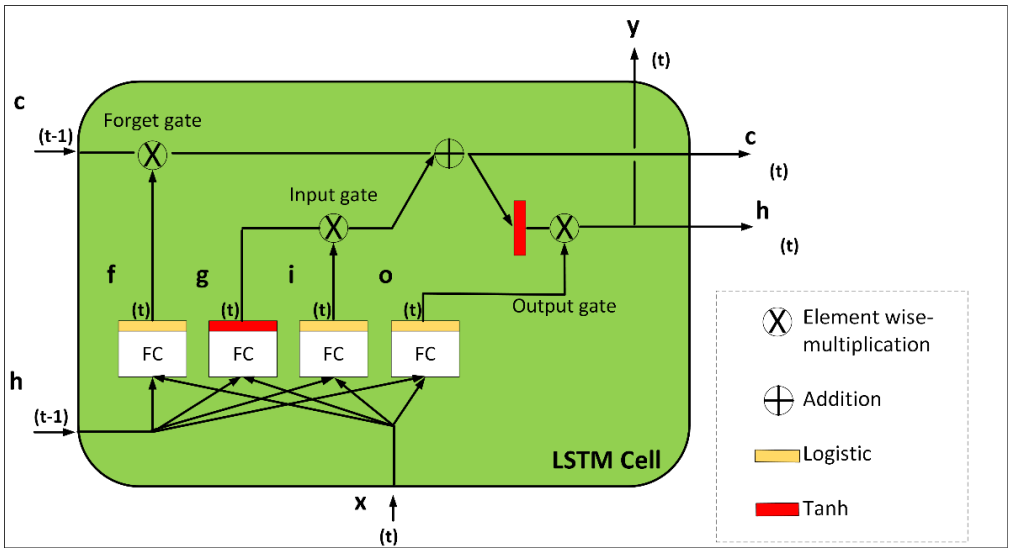

Figure 2. LSTM Architecture.

2.1.4. Performance metrics and model evaluation.

The last piece of process is the model assessment in terms of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE). As a result, a test data set is adopted with the purpose of testing the ability of a model to generalize from unseen data. Other differences between the models can be observed by examining how the models perform with regards to the actual data by plotting them in a predicted vs. actual bases.

Mean Absolute Error (MAE) quantifies the average of measurable errors between estimations and actual values disregarding their direction [19]. It is the mean of the absolute differences between the fitting function evaluated at the observation points, and the observations [20]. So, in the realm of battery discharge we are trying to predict here, a lower MAE is better [21]. The MAE calculation is presented in Equation (9) [22-23].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|yi - \acute{y}i| \tag{9}$$

where n is the number of data points; yi is the actual value; and $\acute{y}i$ is the predicted value.

Root Mean Square Error (RMSE) is a simplest and most comprehensible measure that calculates the square root of the average of the squared residuals [24]. It gives the amount by which the error may be expected to vary, and it is also generally applicable to large errors [25]. In state estimation especially when determining SoC of a battery, a lower RMSE shows that the model had better performance [26]. The RMSE calculation in represented in Equation (10) [27 - 28].

$$RMSE = \sqrt{\frac{1}{n}\sum^{n}\ (yi - \acute{y}i)^2} \tag{10}$$

where *n* is the number of data points; *yi* is the actual value; and *ýi* is the predicted value

2.1.5. Model validation using data experiment.

In this study, we implemented and compared two distinct deep learning architectures: Long Short-Term Memory (LSTM) and Dense Neural Network (DenseNet) for battery voltage prediction. Both models were trained using the same six input features from our battery dataset. The implementation of LSTM architecture involves two subsequent LSTM layers; each of which has 64 units. As interneurons, this model uses both native Sigmoid and Tanh activation functions inside LSTM gating mechanisms; the output related to the voltage is predicted with the help of linear activation. In this research, we trained the LSTM model for five thousand epochs with the Adam optimizer with a learning rate of 0.001 was used. In the meantime, the DenseNet architecture was depicted with pyramid-like topology and three hidden layers with shrinking size of 64, 32, and 16 neurons, correspondingly. Every hidden layer uses ReLU activation, and all the layers are supplemented with batch normalization for training purposes. To reduce the issues of overfitting, we added a dropout of 0.1 on all the layers of the network. Like in the LSTM model, the output layer of this model also uses linear activation function.

The DenseNet model was trained for 200 epochs using the Adam optimizer where the learning rate was set 0.001. Each of these two models were designed to yield voltage predictions via a single neuron output. The difference in the numbers of epochs between these two models is also notable: LSTM has 5000 epochs while DenseNet has only 200; this owes to the fact that different architectures have their own convergence features. The architectural parameters as well as the hyperparameters applied to construct the two models compared in this study are outlined in Table 1. This outline gives the general view of the LSTM and DenseNet architecture and disparities and similarities and forms an appropriate starting point for comparing their characteristics and pros and cons.

Table 1. Models' architecture.

Parameter	DenseNet	LSTM
Input layer	6 Features	6 Features
Hidden layer	3 Layers (64 x 32 x 16)	2 LSTM layers (64 x 64)
Output layer	1 Neuron	1 Neuron
Activation function	ReLu	Sigmoid & Tanh
Output function	Linear	Linear
Regularization	Batch normalization	-
Dropout	0.1	-
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Epoch	200	5000

In our experimental setup, we employed five different battery datasets, each representing varying load conditions: 100W, 130W, 180W, 200W, and 220W. This diverse range of load patterns allows us to evaluate the performance of the models across different operational scenarios. Following standard practice, the data for each condition was split into an 80:20 ratio, with 80% used for training the models and the remaining 20% reserved for testing. This structured approach ensures a reliable and consistent evaluation of the models across all load conditions. The experimental design was carefully crafted to leverage the unique strengths of each architecture while maintaining consistency in key hyperparameters, such as the optimizer and learning rate, across all models. This ensures a fair and accurate comparison of their predictive capabilities. Table 1 show the DenseNet and LSTM model's architecture.

3. Experimental Setup

The research design methodology is followed to identify multivariate interaction between battery parameters, environmental variables and discharge profile driving battery effectiveness and durability [29]. As for the case of battery systems, this phase is more crucial because it involves taking accurate time-series observations from Battery Monitoring Device (BMD), environmental and load metering devices. This detailed data acquisition and analysis paradigm allows tracking a wide range of discharge parameters and trends, both momentary and progressive, which is critical for building highly effective predictive models.

The design structure is divided into three interconnected components (as presented in Figure 3), data acquisition entails collection of basic battery performance data via different sensing methods; data preprocessing is the arrangement of the actual battery data in a more appropriate form for inspection; and data selection is the examination of the features that are most suitable for battery discharge behavior prediction. Every part of the system is modelled to mitigate problems arising from time series data, multiple sensors and different operating conditions in battery discharge prediction.

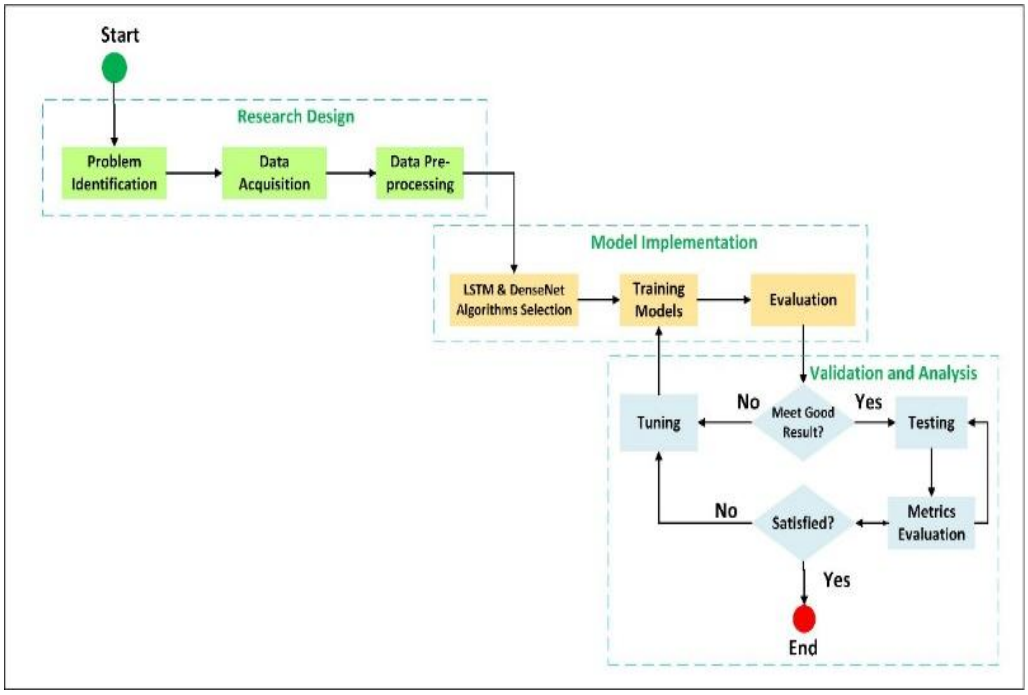

Figure 3. Flowchart of study.

3.1. Experimental Setup.

3.1.1. Experimental Device.

The device used in this study includes sensors, LCD display, battery, STM32 microcontroller and Raspberry Pi 4 microcomputer. Table 2. provides the battery with necessary information to ensure optimal operation of the data acquisition process. An experimental device used in this study is presented in Figure 4.

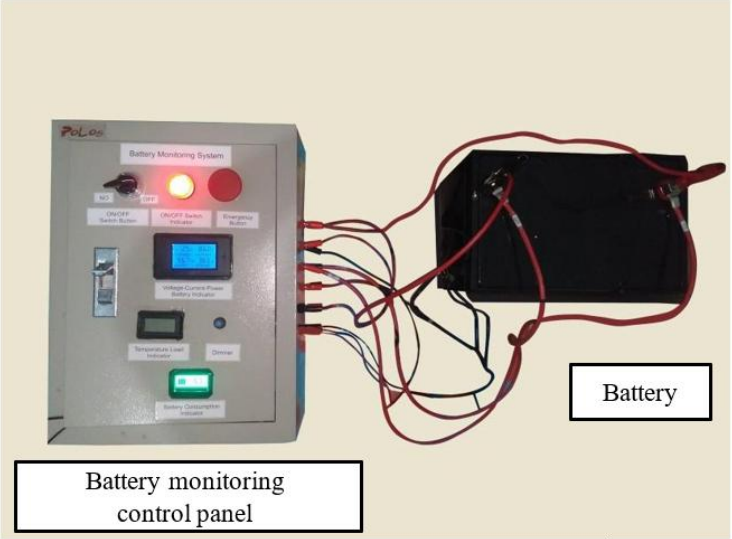

Figure 4. Battery monitoring experimental device.

Table 2. Battery specification used in the experiment.

Specification	Description
Brand	Luminous
Type	Lead acid
Voltage	12 Volt
Capacity	70 Ah (Ampere hour)
Dimension (L x W x H)	42 x 19 x 23 cm

3.1.2. Devices connection for real-time monitoring and data acquisition.

A detail of the device's connection including the sensors and the microcontroller is presented in Figure 5. A real-time monitoring system is employed to observe the dynamic working state of a system in real-time. Here, the measuring instruments installed on the mechanical system transmit data to various hardware components including STM32 or Raspberry Pi in real-time where the data is processed and then provided to users in a comprehensible format. This system will enable the continuable monitoring of machines or systems and will prompt actions where necessary such as closing down of more machinery in case of critical failure.

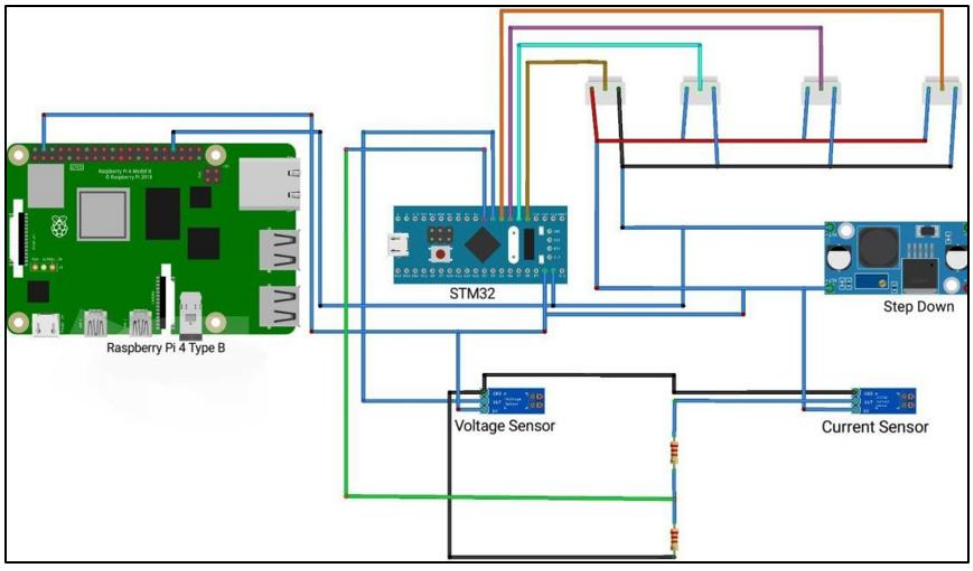

Figure 5. Sensors and microcontrollers wiring diagram.

3.2. Data acquisition.

3.2.1. Historical performance data.

In this study, the initial data used was obtained from experimental testing using a physical battery monitoring system. During the battery discharge process, parameters used as datasets including voltage, current, SOC, and temperature were measured in real-time using sensors integrated with a microcontroller. The collected data was then transmitted, continuously recorded, and stored in a structured dataset format in the form of CVS (Comma-Separated Values) file, where each entry represents a time series observation associated with a specific timestamp.

In this context, historical performance data derived from previous discharge cycles plays a crucial role in supporting the development of predictive models. Historical data is further processed during the pre-processing stage, including cleaning, alignment, and normalization, to ensure consistency with real-time measurements. By integrating historical patterns with current observations, the system is able to capture both long-term degradation trends and short-term dynamic responses of the battery. Furthermore, historical data is primarily used during the mode training phase, enabling machine learning models such ad LSTM and DenseNet-CNN to learn the underlying temporal dependencies and discharge characteristic.

3.2.2. Data Format Standardization.

The standardization process starts with importing raw csv files and organizing them in a necessary Excel format. This includes sorting measures such as voltage, current, temperature, and SOC into specific columns of the BMS. Excel template is clean and used with proper format for timestamp/ time value, numbers and units to enhance efficiency in analyzing the data. Depending on these load conditions, data is amassed ranging from 100W to 220W (100W, 130W, 180W, 200W, 220W). In terms of load conditions, data collection is performed based on a specific time schedule for each load. To perform the process, there is a need to keep environmental conditions under check in order to steadily recharge the battery before each test cycle. During each load point, voltage, current, and temperature data are recorded for the entire discharge cycle to establish a large dataset required for the predictive modeling system. Table 3 Provides details for load conditions ranging from 100 - 220W. The dataset is partitioned into training and testing sets with a ratio of 80:20, where 80% is allocated for model training and 20% for testing. This approach ensures a fair and consistent evaluation of the proposed machine learning models across all operating conditions. Each sample represents a time-series data point recorded at one-minute intervals, including parameters such as temperature, battery voltage, current, and capacity.

Table 3. Dataset distribution for each load condition.

Load condition (W)	Total data (Samples)	Training data (80%)	Testing data (20%)
100	280	224	56
130	1252	1001	251
180	441	353	88
200	492	394	98
220	714	571	143

3.2.3. Data preprocessing.

Data preprocessing includes data cleaning and data transforming. The filtering techniques are used in order to eliminate noise in which measurements of voltage, current or temperature, fluctuating

rapidly but not representative of battery behavior, are reduced. For anomaly detection, statistical techniques provide techniques of identification of outliers in the data set which distorted the expected pattern. This comprises of cases of abrupt voltage fluctuations such as low voltage, high current or high or low temperature which could be giving erroneous battery trends rather than the true battery behavior. Gapping treatment involves methods such as using estimates between actual measured data or excluding invalid records when gaps must be too wide to approximate with good accuracy. Hence, the choice of strategy depends on the duration and frequency of missing data segments. Time series alignment ensures that all measurements are synchronized and sampled at consistent intervals. This aligned data structure is crucial for developing accurate predictive models, as it enables proper correlation between different parameters and their temporal evolution during discharge cycles.

3.3. Model selection and parameters tuning.

The first of them is to select appropriate methods depending on the data nature and the objectives of the study. Battery discharging action is nonlinear in nature, and for the same reason, using methods such as AI-based algorithms are more suitable. Linear regression can be used for initial baseline models while the neural networks which include CNN DenseNet and LSTM are more appropriate when working with discharging data in terms of time series. More precisely, Random Forest Classification, Gradient Boosting Machines can be used to manage interactions in the dataset. The sampling method should match the study's goal and justification needs to be made depending on its feasibility in capturing temporal and non-linear relationships. Tuning hyperparameters facilitates generation of good results from the developed Model. Hyperparameters that act on the rates of learning or dropout, or on the number of trees in random forests, for example, are tuned by common methods grid search, random search, and Bayesian optimization, respectively [30]. This often requires assessing the model on validation sets and using cross-validation methods to reduce overfitting and increase generalizability.

4. Results and Discussion

A performance evaluation of LSTM and DenseNet and GRU-LSTM models using five operating conditions of 100W, 130W, 180W, 200W and 220W occurred in this chapter, organized into two distinct sections: Results and Discussion. Here, in the Results section, the research employs Root Mean Square Error (RMSE) alongside Mean Square Error (MSE) and Mean Absolute Error (MAE) for the evaluation process. A battery simulation machine collected the dataset that enabled simulation of load and voltage properties. The capability produced a broad dataset which contained different operational situations and scenarios.

A battery simulation machine was used to simulate battery conditions from a learning dataset. The dataset was obtained through a 10-hour controlled physical experiment, which was used to train the battery simulation system. Through this process, the simulation engine learned the underlying operational characteristics of the battery, thus mimicking real-world battery behavior under various operating conditions. The capability produced a broad dataset which contained different operational situations and scenarios from battery simulation machine.

The discussion section further elaborates on these analyses, providing insights into the reasoning behind the conclusions and exploring the potential conditions necessary for the effectiveness of the models studied. It also examines the broader implications of the research

findings. Through the use of graphical representations, our work aims to provide a clear and comprehensive understanding of the utility of these methods in predicting battery voltage. Additionally, it highlights their applicability, strengths, and limitations across various operational scenarios.

4.1. Prediction result.

4.1.1. DenseNet model.

In order to get better visualization of performance of different models proposed in this study, Figure 6 shows prediction of voltage against actual voltage for different loads. These figures show the DenseNet model prediction disposition during discharge cycles at different loads 100W, 130W, 180W, 200W, 220W each. Each graph has a red line which belongs to the predictions of the model and the blue line is the actual voltage output. The time duration in hours is on x-axis and the voltage levels are displayed on the Y- axis. Figures 6 from a – e, present a simple comparison of the predicted and actual measurements, aid in understanding the effectiveness and applicability of the model for examining voltage characteristics responding to different loads. The model's remarkable precision in all load scenarios can be appreciated where predicted and actual values are visually aligned with discrepancies noticeable at singular point instances.

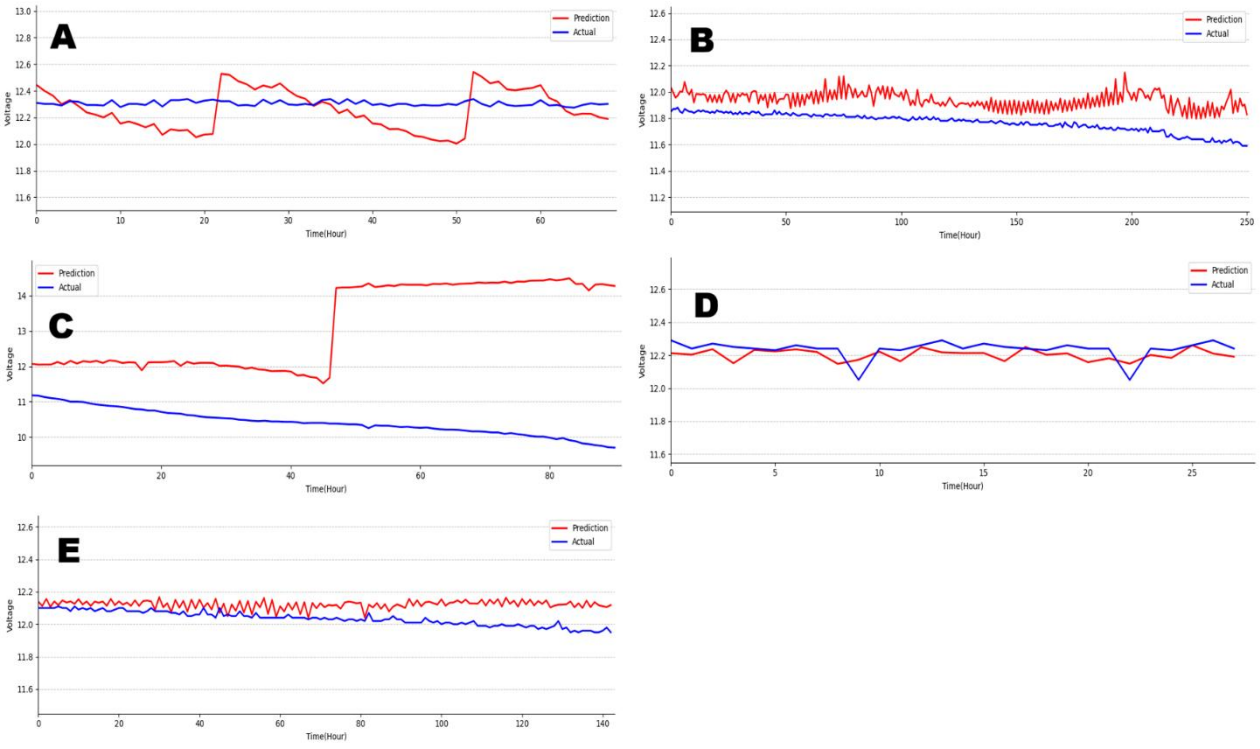

Figure 6. (A) DenseNet model for 100-watt discharge prediction vs actual; (B) DenseNet model for 130-watt discharge prediction vs actual; (C) DenseNet model for 180-watt discharge prediction vs actual; (D) DenseNet model for 200-watt discharge prediction vs actual; (E) DenseNet model for 220-watt discharge prediction vs actual.

Figure 7(a) shows the training and validation loss curves over 200 epochs. Both curves demonstrate a sharp decline in the initial epoch from 0.8 to approximately 0.1. Both curves' subsequent convergence and stability after epoch 50 indicate effective model learning without significant overfitting issues. In Figure 7(b), RMSE, epochs and the corresponding computational time are as follows; RMSE shows the error measured on the test set plotted over the training epochs

of the model. The subsequent epochs again show a sustained decline in error, after which the curve plateaus from epoch 50 onwards and shows that the model has effectively converged and requires only optimum training epochs.

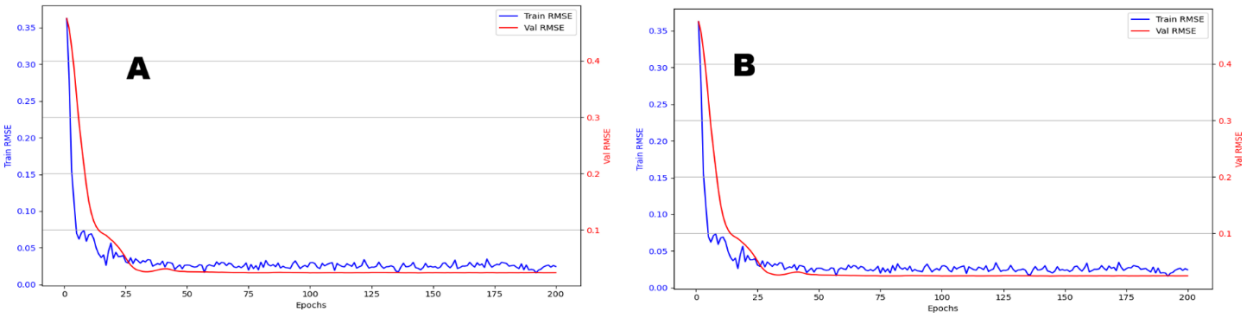

Figure 7. (A) CNN Dense model train and validation curve; (B) DenseNet Model RSME vs Epoch vs Time.

4.1.2. LSTM model.

In Figure 8, a comparison is made between the actual voltage value as acquired by the voltage sensor, and its corresponding LSTM model predicted value under five different load levels, namely 100W, 130W, 180W, 200W, and 220W. Both figures illustrate model prediction of voltage during discharge cycles in red line against actual values in blue line, with the x and y axis denoting time in hours and voltage respectively. From these plots, the reader can judge that the LSTM model will have varying levels of prediction performance for different loads, with some rather large and frequent changes being most evident in the 100W and 180W cases, although the 220W predictions appear more stable by comparison.

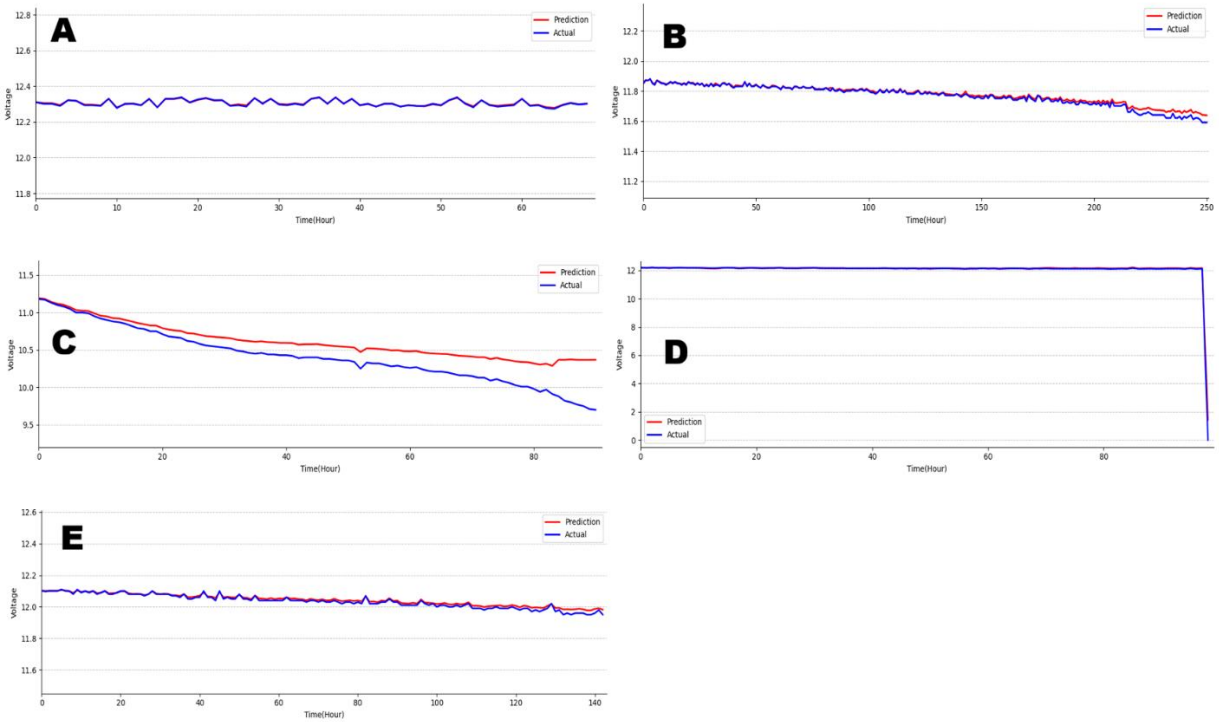

Figure 8. (A) LSTM model for 100-watt discharge prediction vs actual; (B) LSTM model for 130-watt discharge. prediction vs actual; (C) LSTM model for 180-watt discharge prediction vs actual; (D) LSTM model for 200-watt discharge prediction vs actual; (E) LSTM model for 220-watt discharge prediction vs actual.

Figure 9(a), plots the loss performance of the LSTM model through the 5000 epochs which shows both the training loss denoted by the blue line and the validation loss represented by the red line that reveal a steep initial decay but begin to diverge after the 2000th epoch, indicating possible cases of over fitting in the later epochs. Figure 9(b) illustrates the impact of epochs and computation time on RMSE of the LSTM model where the error rates do not remain constant throughout the epochs but have periodic sharp ascending values, whereas the computation time (in red line) is directly proportional with the number of epochs.

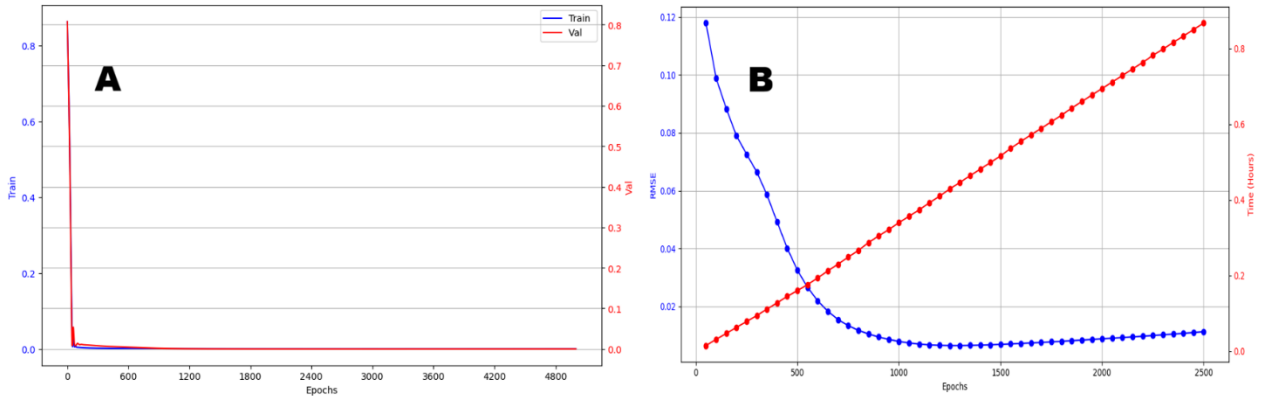

Figure 9. (A) LSTM model train and validation curve; (B) LSTM model RSME vs epoch and time.

4.2. Discussion.

Analysis of the two models reveals several important findings from the comparative evaluation. Overall, the LSTM model demonstrated superior performance under all tested conditions, consistently producing lower average prediction errors across different loading states. Its architectural design effectively captures temporal dependencies in voltage patterns, resulting in particularly accurate predictions under low-load conditions, especially at 100 W. The use of a battery simulation machine for dataset generation further enabled the LSTM model to develop strong generalization capabilities across different testing environments. In contrast, the CNN-DenseNet model showed relatively better performance under mild load conditions; however, its accuracy declined at higher load levels. While CNN-DenseNet is effective in capturing local feature patterns, it requires further optimization to maintain stability and accuracy under high-load conditions. These inconsistencies may be attributed to variability introduced during the battery simulation process, which increases the complexity of learning robust degradation patterns. For future research directions, several improvements are recommended to enhance the practical applicability of these models. First, hybrid architectures should be explored by combining LSTM with DenseNet or GRU–LSTM structures to improve prediction accuracy across varying workload conditions. Second, the dataset should be expanded by incorporating a wider range of operational scenarios and environmental variables that influence battery performance. Third, real-time prediction capabilities should be developed by optimizing model architectures to ensure faster inference without compromising accuracy. Fourth, attention mechanisms should be integrated to better capture long-term dependencies in voltage and degradation patterns. Finally, adaptive learning techniques should be implemented to allow model parameters to dynamically adjust across different testing conditions. Collectively, these enhancements could significantly improve the robustness and real-world applicability of deep learning-based battery voltage prediction systems.

5. Conclusions

The experimental analysis revealed that the LSTM model outperformed the CNN- DenseNet approach in predicting battery voltage, achieving higher accuracy across all evaluated metrics. This superior performance can be attributed to LSTM's ability to effectively capture long-term dependencies in time-series data, making it particularly well-suited for tasks involving sequential data like battery voltage prediction. In contrast, the DenseNet model demonstrated strong performance under low load conditions, leveraging its dense connectivity pattern to efficiently reuse features and maintain stability. However, its performance degraded at higher load conditions, suggesting the need for further architectural adjustments or hyperparameter tuning to enhance its robustness across a wider range of operational scenarios. To ensure a comprehensive evaluation, a battery simulation machine was employed to replicate real-world operational scenarios. This setup allowed for a thorough assessment of how each model performed under varying load conditions, providing valuable insights into their strengths and limitations. The simulation environment not only validated the models' predictive capabilities but also highlighted the importance of considering diverse operational contexts when developing battery voltage prediction systems. Overall, the findings underscore the need for continued research and development to optimize these models further, ensuring their applicability and reliability in practical, real-world applications.

Author Contribution

Conceptualization was carried out by Mukhidin Wartam, Yogi Reza Ramadhan, and Ryan Fitriawan. The methodology was developed by Mukhidin Wartam and Yogi Reza Ramadhan. Data collection was conducted by Mukhidin Wartam, Yogi Reza Ramadhan, and Ryan Fitriawan, while data analysis was performed by Mukhidin Wartam. The original manuscript was written by Mukhidin Wartam, Yogi Reza Ramadhan, and Hayati Yassin. Supervision of the study was provided by Hayati Yassin and Taufik Taufik.

Funding

Not available.

Competing Interest

All authors should disclose any financial, personal, or professional relationships that might influence or appear to influence their research.

References

[1] Li, W.; Zhang, H.; Van Vlijmen, B.; Dechent, P.; Sauer, D.U. (2022). Forecasting battery capacity and power degradation with multi-task learning. *Energy Storage Materials*, *53*, 453–466. https://doi.org/10.1016/j.ensm.2022.09.013.

[2] Wang, R.; et al. (2024). Degradation analysis of lithium-ion batteries under ultrahigh-rate discharge profile. *Applied Energy*, *376*, 124241. https://doi.org/10.1016/j.apenergy.2024.124241.

[3] Yang, H.; et al. (2024). Behavioral description of lithium-ion batteries by multiphysics modeling. *DeCarbon*, *6*, 100076. https://doi.org/10.1016/j.decarb.2024.100076.

[4] Liu, Q.; Shang, Z.; Lu, S.; Liu, Y.; Liu, Y.; Yu, S. (2025). Physics-guided TL-LSTM network for early-stage degradation trajectory prediction of lithium-ion batteries. *Journal of Energy Storage*, *106*, 114736. https://doi.org/10.1016/j.est.2024.114736.

[5] Tebbe, J.; et al. (2025). Innovations and prognostics in battery degradation and longevity for energy storage systems. *Journal of Energy Storage*, *114*, 115724. https://doi.org/10.1016/j.est.2025.115724.
[6] Safitri, M.; Adji, T.B.; Cahyadi, A.I. (2025). Enhanced early prediction of Li-ion battery degradation using multicycle features and an ensemble deep learning model. *Results in Engineering*, *25*, 104235. https://doi.org/10.1016/j.rineng.2025.104235.
[7] Madani, S.S.; Ziebert, C.; Marzband, M. (2023). Thermal behavior modeling of lithium-ion batteries: A comprehensive review. *Symmetry*, *15*, 1597. https://doi.org/10.3390/sym15081597.
[8] Liu, Y.; Hou, B.; Ahmed, M.; Mao, Z.; Feng, J.; Chen, Z. (2024). A hybrid deep learning approach for remaining useful life prediction of lithium-ion batteries based on discharging fragments. *Applied Energy*, *358*, 122555. https://doi.org/10.1016/j.apenergy.2023.122555.
[9] Mendhe, A.B.; Panda, H.S. (2025). Recent trends of machine learning on energy storage devices. *Nexus Research*, *2*, 100119. https://doi.org/10.1016/j.nexres.2024.100119.
[10] Islam, Md.A.; et al. (2025). Machine learning in advancing anode materials for lithium-ion batteries – A review. *Inorganic Chemistry Communications*, *171*, 113577. https://doi.org/10.1016/j.inoche.2024.113577.
[11] Li, S.; Zhao, P. (2021). Big data driven vehicle battery management method: A novel cyber-physical system perspective. *Journal of Energy Storage*, *33*, 102064. https://doi.org/10.1016/j.est.2020.102064.
[12] Chae, S.G.; Bae, S.J.; Oh, K.-Y. (2025). State-of-health estimation and remaining useful life prediction of lithium-ion batteries using DnCNN-CNN. *Journal of Energy Storage*, *106*, 114826. https://doi.org/10.1016/j.est.2024.114826.
[13] Zhao, J.; Wang, Z. (2024). Specialized convolutional transformer networks for estimating battery health via transfer learning. *Energy Storage Materials*, *71*, 103668. https://doi.org/10.1016/j.ensm.2024.103668.
[14] Gao, D.; Liu, X.; Zhu, Z.; Yang, Q. (2022). A hybrid CNN-BiLSTM approach for remaining useful life prediction of EVs lithium-ion battery. *Measurement and Control*, *56*, 371–383. https://doi.org/10.1177/00202940221103622.
[15] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. (2017). Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Honolulu, HI, USA; pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.
[16] Palma, G.; Chengalipunath, E.S.J.; Rizzo, A. (2024). Time series forecasting for energy management: Neural circuit policies (NCPs) vs. long short-term memory (LSTM) networks. *Electronics*, *13*, 3641. https://doi.org/10.3390/electronics13183641.
[17] Wu, L.; Guo, W.; Tang, Y.; Sun, Y.; Qin, T. (2024). Remaining useful life prediction of lithium-ion batteries based on neural network and adaptive unscented Kalman filter. *Electronics*, *13*, 2619. https://doi.org/10.3390/electronics13132619.
[18] Zhang, W.; Jia, J.; Pang, X.; Wen, J.; Shi, Y.; Zeng, J. (2024). An improved transformer model for remaining useful life prediction of lithium-ion batteries under random charging and discharging. *Electronics*, *13*, 1423. https://doi.org/10.3390/electronics13081423.
[19] Oyucu, S.; Dümen, S.; Duru, İ.; Aksöz, A.; Biçer, E. (2024). Discharge capacity estimation for Li-ion batteries: A comparative study. *Symmetry*, *16*, 436. https://doi.org/10.3390/sym16040436.
[20] Tu, H.; Borah, M.; Moura, S.; Wang, Y.; Fang, H. (2024). Remaining discharge energy prediction for lithium-ion batteries over broad current ranges: A machine learning approach. *Applied Energy*, *376*, 124086. https://doi.org/10.1016/j.apenergy.2024.124086.
[21] Ma, C.; Wang, Y.; Li, F.; Zhang, H.; Zhang, Y.; Zhang, H. (2024). Constructing Attention-LSTM-VAE power load model based on multiple features. *Advances in Mathematical Physics*, *2024*, 1041791. https://doi.org/10.1155/2024/1041791.
[22] Hodson, T.O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, *15*, 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022.
[23] Fu, B.; et al. (2025). Predictive modeling for durability characteristics of blended cement concrete utilizing machine learning algorithms. *Case Studies in Construction Materials*, *22*, e04209.

https://doi.org/10.1016/j.cscm.2025.e04209.

[24] Liang, C.; Xia, B.; Yue, S.; Zhang, F.; Qu, L.; Wang, S. (2024). A quantum particle swarm optimization extended Kalman quantum particle filter approach on state of charge estimation for lithium-ion battery. *Journal of Energy Storage*, *100*, 113677. https://doi.org/10.1016/j.est.2024.113677.

[25] Mao, S.; et al. (2023). Multi sensor fusion methods for state of charge estimation of smart lithium-ion batteries. *Journal of Energy Storage*, *72*, 108736. https://doi.org/10.1016/j.est.2023.108736.

[26] Pau, D.P.; Aniballi, A. (2024). Tiny machine learning battery state-of-charge estimation hardware accelerated. *Applied Sciences*, *14*, 6240. https://doi.org/10.3390/app14146240.

[27] Xia, W.; Xu, J.; Liu, B.; Duan, H. (2024). A novel denoising autoencoder hybrid network for remaining useful life estimation of lithium-ion batteries. *Energy Science & Engineering*, *12*, 3390–3400. https://doi.org/10.1002/ese3.1823.

[28] Willmott, C.; Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*, 79–82. https://doi.org/10.3354/cr030079.

[29] Hou, J.; Su, T.; Gao, T.; Yang, Y.; Xue, W. (2025). Early prediction of battery lifetime for lithium-ion batteries based on a hybrid clustered CNN model. *Energy*, *319*, 134992. https://doi.org/10.1016/j.energy.2025.134992.

[30] Zhao, L.; Song, S.; Wang, P.; Wang, C.; Wang, J.; Guo, M. (2024). A MLP-Mixer and mixture of expert model for remaining useful life prediction of lithium-ion batteries. *Frontiers of Computer Science*, *18*, 185329. https://doi.org/10.1007/s11704-023-3277-4.