



State-of-Health Prediction for Li-ion Batteries for Efficient Battery Management System Using Hybrid Machine Learning Model

Varatharaj Myilsamy¹ · Sudhakar Sengan² · Roobaea Alroobaea³ · Majed Alsafyani³

Received: 30 April 2022 / Revised: 19 May 2023 / Accepted: 12 June 2023 / Published online: 19 June 2023
© The Author(s) under exclusive licence to The Korean Institute of Electrical Engineers 2023

Abstract

Since Lithium-ion (Li-ion) batteries are frequently used for real-time applications, evaluating their State of Health (SoH) is crucial to guarantee their effectiveness and safety. Model-based methods with SoH prediction are helpful. However, the issues with battery modelling have led to a greater dependence on machine learning (ML). As a significant step in assessing the effectiveness of ML techniques, data preprocessing has also drawn much attention. In this work, a new preprocessing method using relative State of Charge (SoC) is proposed; further, this paper describes a hybrid learning model (HLM) that combines auto-regressive integrated moving average (ARIMA), gated recurrent unit (GRU) and convolutional neural network (CNN). Data: proposed HLM uses time-series and SoC domain data; the ARIMA + GRU algorithm trains the time-series data, while CNN trains the SoC domain data. Both outputs are mean averaged to get the final output prediction. The proposed HLM is evaluated for root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) using the National Aeronautics and Space Administration (NASA's) randomized battery usage data set (RBUDS). The results indicate that the recommended HLM is more accurate and has a smaller error margin than existing ML models.

Keywords Machine learning · CNN · ARIMA · GRU · Energy storage devices · Li-ion batteries

1 Introduction

Global Carbonic Dioxide 2 (CO₂) and climate improvement emissions declined twice as much in the last 20 years as in 1945. At the end of the 2nd World War, demand for carbon

fuels fell significantly in the last quarter of 2020 and the 1st quarter of 2021. Despite this, the climate improvement was only temporary, and the rate of increase in greenhouse gas emissions (GHG) levels resumed when the worldwide economy recovered from the Coronavirus disease (COVID-19) pandemic [1]. They are concerned about intermittent renewable energy (RE) sources, including wind and solar energy. The planet progresses toward RE's growing availability, efficiency, and economic performance; sustainable development goal 7 (SDG-7) is achieved as energy becomes more sustainable and available [2].

As a result, energy storage systems (ESS) should be used in conjunction with them. As a result, the ESS, which provides a real-world solution for network control and power stability, has grown in importance [3]. On the other hand, the current ESS lacks the infrastructure to understand and use energy fully. The ESS industry demands, among other things, innovations in power loss planning, long life (high battery uptime), and better Return on Investment (RoI). There has never been a more critical time for businesses to act quickly, optimize, and differentiate themselves in space or risk falling behind. Innovative technology progresses at a breakneck pace. There have been significant advances in

✉ Sudhakar Sengan
sudhasengan@gmail.com

Varatharaj Myilsamy
varatharaj_ms80@rediffmail.com

Roobaea Alroobaea
r.robai@tu.edu.sa

Majed Alsafyani
alsufyani@tu.edu.sa

¹ Department of Electrical and Electronics Engineering, V.S.B College of Engineering Technical Campus, Coimbatore, Tamil Nadu 642109, India

² Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu 627152, India

³ Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia

machine learning (ML) and deep learning (DL), from chatbots to generative modelling. Machines can now process and analyse a high volume of data with sophisticated methods for these concepts.

Energy storage device (ESD) is categorised as mechanical, electrochemical, chemical, power systems, or heat transfer, depending on the ESS used. Depending on the technology and storage capacity, ESD is used for uninterruptible power supply (UPS), transmission and distribution system support (TDSS), or large-scale generation. The technologies that have recently received the most attention among electrochemical, chemical, and physical ESD are covered by UPS and TDSS. Sodium–Sulfur (Na–S), lead and advanced lead-acid, super-capacitor, Li-ion, and flywheel batteries are all examples of redox flow batteries. They are also examples of representative technologies. Lithium-ion (Li-ion) batteries are now widely used. Battery Energy Storage System (BESS) is an ESS that includes batteries.

A BESS is an ESS that collects energy from multiple sources, stores it in rechargeable batteries, and then uses it later. The battery's electrochemical energy is discharged and distributed to homes, electric vehicles (EV), and industrial and commercial resources if necessary. BESS is a hybrid system that includes hardware and lower and higher-end software. BESS is divided into (a) a battery system (BS), which is made up of individual battery cells that convert chemicals into electrical energy, and (b) battery management system (BMS). The BMS ensures the safety of the battery system. It includes two components: (1) a battery cell health monitor that measures parameters and states such as State of Charge (SoC) and State of Health (SoH), and (2) an inverter/power conversion system (PCS) that checks the health of battery cells. It converts the direct current (DC) generated by batteries into alternating current (AC), which is distributed to facilities, and (3) an Energy Management System (EMS). EMS coordinates the work of a BMS, a PCS, and other BESS components. Within ESS, it is in charge of energy flow monitoring and control.

Depending on the application, the complexity of a BMS varies greatly. A simple *fuel gauge* integrated circuit (IC), such as those that can suffice in simple cases, such as single-cell batteries in smartphones or e-book readers [4], using rudimentary methods, these ICs predict the battery's current SoC by measuring voltage, temperature, and current. The BMS must perform more sophisticated tasks in complex devices like EV cars. Advanced algorithms are yet required, for example, to find how far a ship can cruise without running out of energy. Measurement of basic parameters such as cell voltage, temperature, and current are needed.

Battery intelligence tools help diagnose problems by providing up-to-the-minute real-world information [5, 6]. All batteries can be checked remotely for battery life and reasons for downtime, if any, using real-time geo-tagging and data

visualization tools. The battery intelligence platform performs '*Standard Cycling Tests*' for BMS benchmark measures and implements novel methods to improve the grid's SoC accuracy in different states. Because of their versatility, Li-ion batteries have sparked interest among battery types. Electronic devices, EVs, and aircraft use environmentally friendly Li-ion batteries. A few benefits are energy, stability, high energy density, long lifetimes, and environmental protection. Because these battery packs are part of a power supply device, their system performance declines and causes accidents if not checked. Because of this, computing how healthy a battery is in a way that is accurate and useful has become a pressing and significant issue.

Even though Li-ion batteries have few benefits, one of the main drawbacks is that their capacity fades over time. In addition, accurate monitoring and ability prediction is critical, as incorrect capacity predictions can result in permanent battery damage due to overcharging or over-discharging [7]. The BMS, on the other hand, is still a challenging problem worth investigating [8]; SoH, State of Function (SoF), and thermal management [9] are examples of typical BMS. One of the most analytical methods used during battery electrode clean production and manufacturing operation is the BMS prediction of SoH [10]. Its capacity and charge figure out a battery's SoH and internal resistance, with the SoH definition derived from the following factors: [11–13]. The SoH battery is essential for deciding how quickly a battery's ability fades. Therefore, finding the SoH of Li-ion batteries is essential to ensure their safety and reliability [14]. On the other hand, Li-ion batteries are composed of complex chemical systems influenced by the environment, such as ambient temperature, resulting in complicated SoH calculations [15]. Also, because battery capacity decreases in a way that is not linear, it is harder to predict SoH and RUL [16].

Many research studies have been investigated to arrive at an accurate prediction of SoH. The types of studies are model-based methods (MBM) and data-driven methods (DDM) [17]. By simulating the battery and accounting for core degradation, MBM calculates the SoH of a battery. Autor [18] presents a stochastic model for predicting battery cell deterioration. The suggested model is inspired by a Markov chain-based method, although it cannot be compared to a Markov process because the transition probabilities change depending on how many cycles a cell has completed. The proposed model can simulate the cell's sudden loss of capability as they approach its end-of-life condition. Author [19] offers a SoC prediction method based on an Improved Extended Kalman Filter (IEKF) that is accurate and robust. An analogous circuit model is developed for offline parameter identification, and the Simulated Annealing-Particle Swarm Optimization (SA-PSO) algorithm is applied. Furthermore, noise adaptation, fading, and linear-nonlinear filtering based on the standard EKF method have

been improved, and the mathematical model has been conducted correspondingly. To deal with a model mismatch, a dual Kalman Filter (KF) is used for online parameter detection.

A study [20] used a state-space model; a method for predicting the battery SoH based on the discharge rate was proposed. A single advanced process with electrode material physics was used to assess a battery's SoH, which decided to consider internal mechanical and chemical battery drain. The study proposes a cell-level lack of consistency evaluation system based on real-world EV operation data. Although this MBM helps predict battery SoH, due to the battery's complex chemical reactions, designing a correct old model for Li-ion batteries is difficult. The state of Li-ion batteries is affected by working temperatures, anode materials, cathode materials, and other environmental factors. As a result, it's hard to make an exact old model for Li-ion batteries [21].

DDM has proven excellent merits in engineering applications in recent years compared to the above-mentioned traditional dependency analysis methods [22]. Modified algorithms are based on KF. Particle filters, support vector regression (SVR) [23], neural networks (NN) [24], and fuzzy logic (FL) were used in the SoH analysis of battery cells. In recent years, NN has been the most rapidly emerging of these methods and has several benefits. For illustration, [25] created a new predictive model for the Li-ion battery lifespan using a feed-forward neural network (FFNN). A neural network was used by [26] to predict a battery's short-term energy consumption. The most challenging part of expressing interactive degradation within battery packs is using multi-degradation data for system description. The most significant issue is high-dimensional (HD) data processing. In this regard, DL has several benefits [27, 28]. For example, the recurrent neural network (RNN) is a traditional time-series model that efficiently and successfully deploys time-dependent relationship models. The ability to screen and support data about the degradation of long-term datasets can be used by an RNN's historical data memory and ability to analyze relationships. This network [29] should achieve a model projection of the multi-degradation and an accurate SoH prediction with the support of DL. NN is being used increasingly to predict and measure the SoH of batteries.

Furthermore, [30] suggested a method for predicting SoH using a type of RNN known as long short-term memory (LSTM). The proposed work predicts the remaining useful life (RUL) of Li-ion batteries with only fewer data points. A dual extended Kalman filter (EKF) and fractional-order model (FOM)-based combined SoC and SoH prediction approach for Li-ion batteries is provided in [31]. An adaptive evolutionary method is used in their model to create a fractional 2nd-order RC model and to identify model limits offline. The SoC and SoH are both predicted using one of the dual filters, and the model

parameters are updated in real-time using the other dual filter. A DNN-based RUL prediction method for multi-battery RUL prediction is suggested in a systematic review [32]. The model is validated using data from Li-ion batteries. For Li-ion battery RUL prediction, [33] proposes a hybrid Elman + LSTM method. The RUL of the battery is predicted using the hybrid Elman + LSTM method based on the battery's capacity, cycle number, and discharge. They say that a simple way to figure out how much power a battery has is to use III-layer BNNs to figure out how much power it has.

Ohmic resistance, polarisation resistance, polarization capacity, and SoC are all inputs to the NN model. The SoH prediction is derived from the NN model's output. Batteries (LiFePo4s) were used to assess the model. The SoH prediction of the technique-based Artificial Neural Network (ANN) pattern recognition structure was presented by [23]. The system is assumed to have voltage, temperature, capacity, energy, and an estimated SoC. When similar patterns appear, the ANN's output predicts the SoH for Li-ion batteries.

Despite this impressive development, most research has improved classification accuracy by incorporating a detailed model and added parameters. As a result, even if a battery's electrochemical factors are unknown, these methods are simple to implement. Since models do not consider battery features, their primary restrictions are their inefficiency and inaccuracy in predicting SoH. [34], researchers propose an improved pre-processing method for improving the effectiveness and precision of SoH prediction in ML models, considering how the Battery's Energy Level (BEL) impacts the battery's features. The proposed pre-processing transforms time-domain data into SoC-domain data using a relative SoC conversion. This simple pre-processing improves accuracy and efficiency using the same ML model. But even such systems have excellent room for improvement. The proposed paper presents a hybrid learning model (HLM) in this purview that integrates auto-regressive integrated moving average (ARIMA) with convolutional neural network (CNN) and Gated Recurrent Unit GRU to predict SoH from time series and SoC re-engineered data. The idea is to infuse the advantages of temporal attributes and battery features. The model is being experimented with using NASA's RBUDS for performance metrics such as root mean square error (RMSE), mean absolute error (MAE), and Mean Absolute Percentage Error (MAPE).

Furthermore, the proposed work is more accurate than the other models. The comparison models included SoC-based and Time-series-based [35]; The results show that the proposed model performs better than the other models.

The main contributions of the proposed work are presented below:

Contribution #1 Make a practical pre-processing method to enhance the accuracy and speed of SoH prediction in ML models, considering that the battery's energy level affects the battery's properties.

Contribution #2 Build a prediction method to make monitoring platforms with limited computing power more accurate.

Contribution #3 To present a HLM that integrates ARIMA + CNN + GRU to predict SoH from time-series and SoC re-engineered data. The idea is to infuse the advantages of temporal attributes and battery features.

The organization of this article is as follows: Sect. 2 presents the methodologies involved with a problem statement and basic concepts. Section 3 discusses the proposed ARIMA + GRU + CNN + HLM. Section 4 describes the results and discussion of the tests, and Sect. 5 concludes this research.

2 Methodologies

2.1 Problem Definition

In the samples, ML aims to find the most suitable model. $\{(x_i, y_i)\}_{i=1}^N$, To determine the mapping from data to target output 'y', where 'x' represents the charging voltage curves, N represents the test and training sets, and 'y' represents the Li-ion battery's output SoH; as a result, the Non-Linear Mapping (NLM) $f(\bullet)$ found as follows, Eq. (1):

$$y_i = f(x_i) \quad (1)$$

All NN systems extract relevant data and derive long-term relationships from time-series data. The time series prediction problem has been challenged using GRU, LSTM, and CNN. Using the network learning model (ARIMA + GRU + CNN), as shown in Fig. 1, a new hybrid NN is proposed in this article. The model is based on time series and SoC data streams. The model employs extensive pre-processing, DL, and accurate prediction. The following section presents the dataset, explains each phase of the proposed model, and presents the evaluation summary.

2.2 Battery Test Dataset

The randomized battery usage repository (RBUR) provided the battery training dataset (BTD) used in this investigation at NASA's Ames Prognostics Center of Excellence. The data in this repository was first used to investigate when load profiles are randomized and capacity fades [36]. The LiFePo4 battery dataset was used in this study. An III-layer

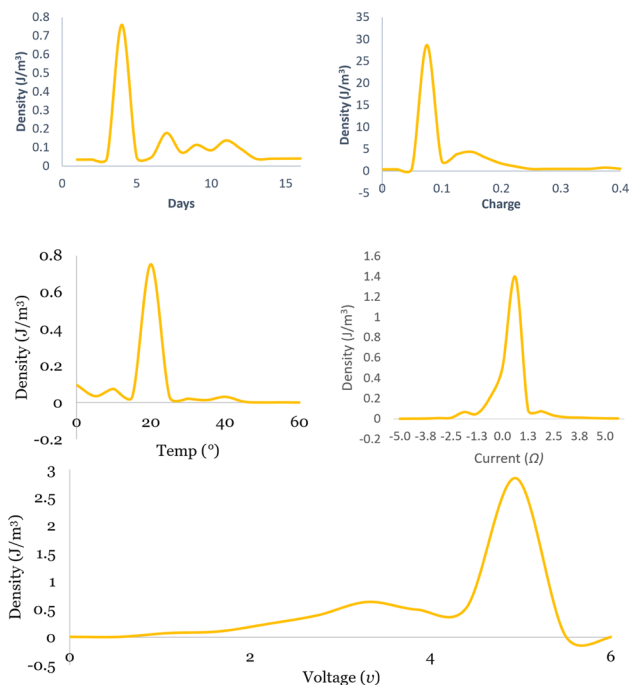


Fig. 1 Battery data smoothed histograms demonstration times, charge throughput, currents, voltages, and temperatures

Table 1 A summary of the dataset

Manufacturer	LG chem
Factor of from	18,650
Chemistry	Comparing lithium cobalt oxide and graphite
# Cells	26
#Q Trials	950
Q Array (Ah)	2,100.80
Period	7 regime-specific groups

BPNN is proposed as a simple approach to predicting the SoH of a battery. The ohmic resistance, polarization resistance, polarization capacity, and SoC are inputting into the neural network (NN) model.

The SoH prediction is generated by the model's output. The model is evaluated using LiFePo4 batteries as the power source. Table 1 below summarizes the battery dataset. A trade-off exists between the higher energy density and the cells' limited life span. The task of cycling and characterizing is described further below. The batteries were divided into 7 groups of 4, each starting a distinct randomized cycling operation. Groups 5–1 were cycled at ambient temperature throughout the experiment results, while groups 2–3 were cycled at 40 °C. Discharge curves were used to carefully consider capacity as a predictor of cell health in all

case scenarios during the regular interval tests performed, and a 2A charge–discharge cycle was implemented (Roughly 1 °C) between the cell voltage levels. All cells had 950 discharge curves, 30 per cell. We chose group 3 for this study experiment, including #RW9, #RW10, #RW11, and #RW12.

The column labeled “#Q-samples” denotes the estimated number of health transitions and capacity scales. Values for the maximum nominal and minimum final capacities are displayed in a row ‘Q-range’.

Four 18,650 Li-ion batteries (#RW9, #RW10, #RW11, and #RW12) were used indefinitely in their experiment, with charging and discharging currents ranging from –4.5 to 4.5 A. The Random Walk (RW) function is given to this charging and discharge method. After about 1500 charging and discharging cycles (5 days), this work did a scale charging and discharging cycle sequence to determine how the battery was doing.

As shown in Fig. 2, batteries had varying initial capacities, which led to SoH being used to evaluate their lives. The SoH is analysed using a mixture of methods, but the most common impedance and the battery’s usable capacity decide the battery’s usability [37]. Because it generally requires devices such as energy dispersive spectroscopy, online tests were performed using the described battery’s load resistance. As a result, this research study used the SoH of a battery based on its usable capacity [38]. Describes this as follows, Eq. (2):

Typically, C_{usable} is its maximum releasable capacity when fully charged, whereas C_{rated} , the capacity rating specified by the manufacturer, is the capacity used over time.

$$SoH = \frac{C_{Usable}}{C_{Rated}} \tag{2}$$

2.3 Preprocessing

Because batteries are measured at regular intervals, they are among the most often used methods for predicting SoH. In

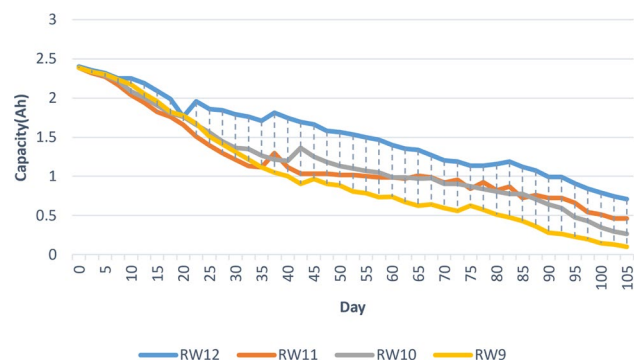


Fig. 2 Cell capacity measurement for group 3

terms of the internal energy of the battery rather than time intervals, SoC decides the battery’s characteristics. Data from the time domain to the SoC domain was changed with the help of a new pre-processing method that used relative SoC.

2.3.1 The Relative SoC

Calculating SoC is based on the battery’s design capacity and how much energy it has stored [39], Eq. (3).

$$SoC_{\text{typical}}(t) = \frac{C(t)}{C_{\text{Rated}}} \tag{3}$$

$C(t)$ is the current capacity at the time ‘ t ’, where C_{rated} is the design size. However, complicated sizes were needed to accurately predict the typical SoC by taking advantage of the correlation between battery features and BEL and data preprocessing to improve ML models for SoH prediction. This work didn’t need indicators for the BEL (or to use a standard SoC). As a result, this research developed a relative SoC, known as SoC, that was linked to the BEL. The usable capacity during the charging method was used to compute the relative SoC, Eq. (4)

$$SoC^k(t) = SoC^k(t_0) + \frac{1}{C_{Usable}^k} \int_{t_0}^t I_c dt \tag{4}$$

‘ t_0 ’ is the starting time, ‘ t ’ is the lifetime, and ‘ C_k ’ usable I_c ’ is the current charging at the ‘ k ’ cycle and the usable computation capacity. It is possible to compute the usable capacity by integrating the current as follows, Eq. (5) [40]:

$$C_{Usable}^k = \int_{t_0}^{t_{\text{cut-off}}} I_d dt \tag{5}$$

where t_0 is the discharging start time, t_{cutoff} occurs when the voltage drops in the battery below the cut-off voltage, ID is the discharge current from the battery at the ‘ k ’ cycle, and ‘ C_0 ’ usable is set to ‘ C_{rated} ’. The virtual SoC was formulated simply by charging between 0 and 100%, even if the battery was degraded. As a result, this research paper examined how to process information based on weighted SoC.

2.3.2 Time-Based Data Testing

Researchers have used time-based data testing to collect data over set periods. Because most equipment measures data at a constant time interval, their approach seemed reasonable. Based on time-based data testing, generated data using the same elements to compare with the proposed SoC-based method. t_{km} , V_{km} , I_{km} , T_{km} are the time, voltage, current, and temperature variance testing points at the ‘ k ’ cycle in the

‘*n*’ BTD. ‘*s*’ is the count of the *m* testing points during a charge cycle, and ‘*k*’ is the testing points for time, voltage, current, and temperature variance at the ‘*k*’ cycle in the ‘*n*’ BTD. The temperature change was determined as follows: Eq. (6) and Eq. (7)

$$D^k_{\text{Timebase } m} = \begin{pmatrix} t_1^k & t_2^k & \dots & t_s^k \\ V_1^k & V_2^k & \dots & V_s^k \\ I_1^k & I_2^k & \dots & I_s^k \\ \Delta T_1^k & \Delta T_2^k & \dots & \Delta T_s^k \\ k & k + \frac{1}{s-1} & \dots & k + 1 \end{pmatrix} \tag{6}$$

$$\Delta T_m^k = T_m^k - \text{Min}(T^k) \tag{7}$$

The cycle count is ‘*k*’, the ‘*m*’ testing points are ‘*m*’, and the temperature at the ‘*k*’ testing point is ‘*T_k*’.

During a single charge cycle of battery #RW12, 20 voltage, current, and temperature testing points were collected. Figure 3a shows that the older battery reached 4.2 v faster than the newer one. The battery’s voltage remained constant at 4.2 v after 9000 s, regardless of their health. As a result, nearly 66% of the voltage data had identical values. In less than 4000 s. or more than 14,000 s, there was no discernible change in the current value. The temperature variance between battery #RW12 after 3000 s was also slightly different. Based on this study’s results, time-domain features were not enough to help us figure out how long the batteries would last.

2.3.3 SoC-Based Data Testing

SoC-based trained data testing is a method that considers the BEL and collects data at a fixed relative SoC interval. The relative SoC is figured out by the battery’s energy level and changes the battery’s features. As a result, this work implemented training data testing based on SoC. A constant

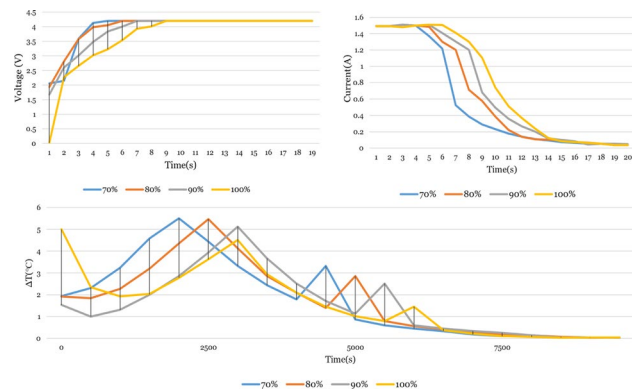


Fig. 3 Battery #RW12 charging data sampled in a constant time interval

relative SoC interval has experimented with variables such as temperature difference, current, voltage, and cyclic relative SoC, Eq. (8):

$$D^k_{\text{SoC_Base } kn} = \begin{pmatrix} SoC_1^k & SoC_2^k & \dots & SoC_s^k \\ V_1^k & V_2^k & \dots & V_s^k \\ I_1^k & I_2^k & \dots & I_s^k \\ \Delta T_1^k & \Delta T_2^k & \dots & \Delta T_s^k \\ k & k + \frac{1}{s-1} & \dots & k + 1 \end{pmatrix} \tag{8}$$

where SoC_{KM} , V_{km} , I_{km} , and T_{km} are the relative SoC, voltage, current, and temperature differences at the *k*th cycle of the ‘*n*’ BTD. ‘*M*’ points stand for them, and the number of facts during one cycle is given by ‘*s*’.

Figure 4 shows the data from battery #RW12 collected throughout a single charge cycle with a constant relative SoC interval and 20 testing points. The data for old and new batteries pre-processed by SoC differed from Fig. 3. For example, the voltage testing values of the battery varied significantly, ranging from 0 to 100%. Compared to time-based pre-processing, which reported a 45% electrical potential difference between old and new battery systems, it showed a 60% variation. Dissimilarities in pre-processing currents and battery temperatures were found to be linked. It was also true for SoC values.

Because unlabeled SoC series are raw, clustering methods divide them into two groups, each labeled ‘*1/0*’. As a result, SoC is clustered offline using the k-means algorithm to generate the labeled dataset required for supervised classification methods for online cell health checks. The criteria for aging and normal battery are measured using SoH. The SoH can be analysed using a mixture of methods, but the most common impedance and the battery’s usable capacity determine the battery’s usability. Because it requires devices such as energy dispersive spectroscopy, online tests were performed using the method described by the battery’s load resistance. As a result, this research study used the SoH of a

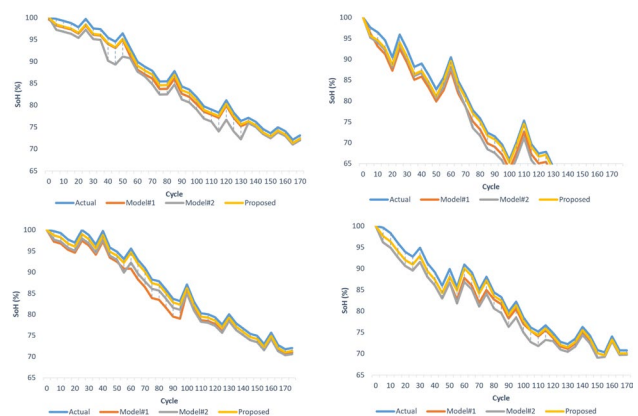


Fig. 4 Battery #RW12 charging data in constant SoC interval

battery based on its usable capacity. The SoH less than 50% is considered as old batteries. The work opted for k-means clustering for dividing.

On the other hand, a large volume of historical data can be objectively categorized by SoC series clustering, but this approach is computationally costly and unsuitable for online screening. In addition, the number of discharge voltage time series with a label of '0' is much less than those with a label of '1' when comparing real-time engineering processes. When a time series dataset is subjected to SoC-based data testing, this results in a class imbalance due to random oversampling issues. It's challenging to train classification algorithms from such unbalanced data, and a minority class of tests could be mistaken for the majority, lowering classification accuracy. An imbalanced SoC data hybrid resampling method is described below.

2.4 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a method of oversampling that creates synthetic tests for minimum classes. The method focuses on the complete set of features to create tests by interpolating positive and closer instances. The method helps to overcome the 'overfit' problem caused by random oversampling.

2.4.1 Procedure

First, calculate the number of observations required for oversampling 'N'. It is usually chosen to ensure a 1:1 binary class distribution, but it can be scaled back if necessary. Iteration starts with a selection of positive classes at random instances. In this case, the k-Nearest Neighbor (k-NN) value, which is 5 by default, is found. Finally, to create synthetic instances, the 'N' of the K occurrences was chosen for interpolation. The difference distance metric calculates the distance between the feature vector (FV) and its neighbours. It is then multiplied by any number between 0 and 1 before being added to the earlier FV. It can be visualized as shown below. Even though the method described above is beneficial, it has shortcomings.

- An artificial line connecting the diagonal and pointing in the same direction connects the synthetic examples. Few classifier algorithms produce a more complicated decision surface as a result
- SMOTE generates noisy data points in the feature space (FS).

2.5 Hybridized SMOTE

Hybridization methods combine undersampling and oversampling techniques. A classifier model was made to predict better how well it would do in tests made by these methods.

2.5.1 SMOTE+ENN

SMOTE+ENN, an alternative hybrid method used in this research study, removes different investigations from the test space. A majority class instance is eliminated if its nearest neighbors incorrectly label it. In this test case, an under-sampling method called ENN is used to find the majority class's nearest neighbours.

2.5.1.1 Edited Nearest Neighbor (ENN) The ENN method defines each observation's KNN and checks if its standard class matches its class. By default, the number of nearest neighbors in ENN is set to $K=3$.

The algorithm for the ENN is as follows:

- Assuming a set of N studies, calculate the number of closest neighbors K. K is assumed to be 3 if K cannot be determined
- Compute k-NN for the observation class in the remaining dataset study and return the standard class from k-NN
- If the study class and standard k-NN differ, the observations and k-NN are taken from the dataset
- Repeat Step 2 and Step 3 until every class's required part has been matched.

When the observation class and the observation's k-NNs differ, ENN eliminates the observation and its k-NN rather than just the observation and its nearest neighbor, as Tomek Links [41–43] does. As a result, ENN is expected to clean the data more thoroughly than Tomek Links.

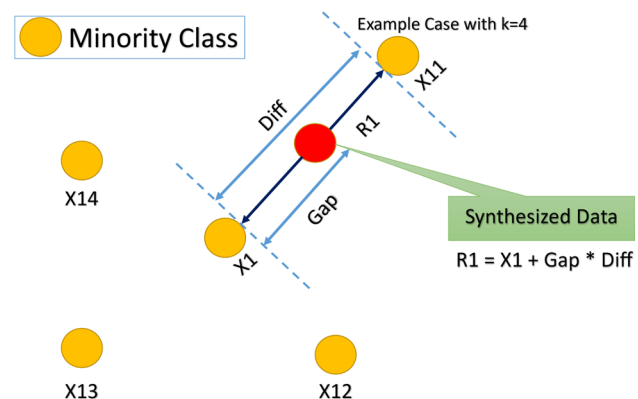


Fig. 5 SMOTE model

When used with SMOTE’s oversampled data, the above method produces significant data cleaning. Due to NN’s samples, the above two classes are free of misclassifications. As a result, the distinction between classes is more visible and briefer (Fig. 5).

The SMOTE + ENN process is explained as follows:

- (a) The class selects random minority data
- (b) Compute random data and k-NN distance
- (c) The minority class’s synthetic sample’s variance is multiplied by the random ranges of 0–1
- (d) Repeat Steps 2–3 until minority class proportion is desired
- (e) K is the closest neighbour. K = 3 if K cannot be computed
- (f) Compute k-NN for the observation class and return to the majority class
- (g) The dataset prevents findings and k-NN with different classes
- (h) Repeat steps 2 and 3 until every class achieves the proportion.

2.6 Feature Extraction Model

2.6.1 Kernel Principal Component Analysis (KPCA) Nonlinear Feature Extraction Theory

After the dataset has been rebalanced, the subsequent step is to extract features from it. Principal component analysis (PCA) is a technique used to extract features from HD data while performing linear dimensional reduction. It converts the input data from the primary HD space to the typical subspace, extracts the major FV from the input, and completes the data exploration destination. A set of data represented by second-order linearly varying correlations (or) generated using a Gaussian Distribution (GD) can usually be PCA’ quickly. Real-world data variations, on the other hand, are well-known to be non-linear and Gaussian, and second-order correlations cannot represent most data. As a result, using PCA will yield poor results. This work proposes ‘KPCA’, a non-linear modified PCA method based on kernel functions. It achieves a linear PCA in FS-F by inherently constructing *nonlinear transformations()* to map input space to nonlinear FS-F. Use the KF to calculate the dot products in the FS given below. It is possible to avoid NLM between the two input samples in the primary universe, say ‘x, y’, Eq. (9):

$$k(x, y) = \Phi(x) \cdot \Phi(y) \tag{9}$$

Figure 6 proves the KPCA discussion structural model. KFs come in a type of shapes and sizes. According to Mercer’s functional analysis theorem, a mapping exists between $\Phi()$ and a dot product space (F) if the KF is a positive

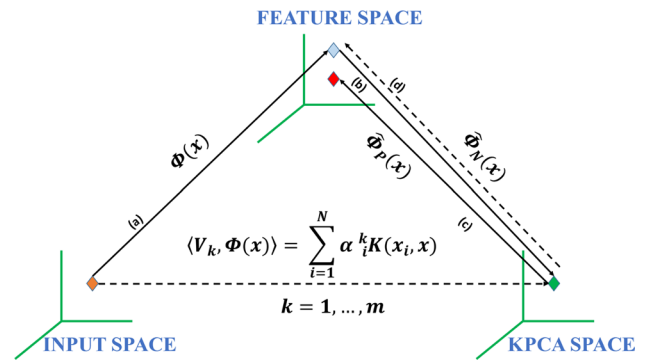


Fig. 6 KPCA framework

integral operator’s continuous kernel. If the KF satisfies Mercer’s Theorem and the best KF is selected, this can have a significant impact on dimensionality reduction.

Few examples of KF, Eq. (10) to (12): Polynomial kernel

$$k(x, y) = \langle x, y \rangle^d \tag{10}$$

Sigmoid kernel

$$k(x, y) = \tanh(\beta_0 \langle x, y \rangle + \beta_1) \tag{11}$$

Radial Basis Kernel Function (RBKF)

$$k(xy) = \text{Exp}\left(\frac{\|x - y\|^2}{c}\right) \tag{12}$$

where d, β_0, β_1 and c have been previously defined by the user.

The sigmoid kernel is only satisfied for specific β_0 and β_1 values, but Mercer’s theorem always satisfies the polynomial and RBKF. In many cases, RBKF is used as the KF of KPCA because of its superior performance; thus, RBKF is used as the KPCA-KF in this research study.

A linear FS-F is better than a non-linear input space because it has a ‘0’ mean $X(x_1, \dots, x_N) \in R_m$. This is because PCA and KPCA covariances, computed using input and the covariances determined using the PCA and KPCA algorithms, are better than the non-linear covariances, assessed using input and the non-linear covariances, Eq. (13–15).

$$C = \frac{1}{N} x_i x_i^T = \frac{1}{N} X X^T \tag{13}$$

$$C^F = \frac{1}{N} \sum_{j=1}^N \Phi_j(x) \Phi_j(x)^T \tag{14}$$

where it is implicit that

$$\sum_{k=1}^N \Phi(x_k) = 0, \tag{15}$$

$\Phi(\cdot)$ is an NLM function that maps input vectors from input space to F. It is noted that the dimension of the FS is high, if not infinite. In the FS, the ‘ λ ’ value problem is solved to compute the Covariance Matrix (CM), Eq. (16)

$$\lambda v = C^F v \tag{16}$$

Here, λ -values ≥ 0 ; λ -vector $v \in F$; λ vector ‘ v ’ for any $\lambda \neq 0$ can be expressed linearly by Eq. (17)

$$\Phi(x_i) : v = \sum_{i=1}^N a(i)\Phi(x_i) \tag{17}$$

Equation (18) is rewritten as the kernel ‘ λ ’ value problem:

$$N\lambda a = Ka, \tag{18}$$

where a $N * N$ matrix K is a kernel matrix, $K = k_{ij} = (\Phi(x_i) \cdot \Phi(x_j)) = k(x_i, x_j)$, and α is the FV of the kernel matrix. When reconstructing input data from FS, use Eq. (19):

$$y_k = v_k, \Phi(x) = \sum_i^N a_i^k \langle \Phi(x_i), \Phi(x) \rangle \tag{19}$$

2.7 Gated Recurrent Unit (GRU)

The proposed RU method captures sequence dependencies on a time-scale-varying basis for each recurrent unit. GRU’s gating model (Fig. 7) regulates its data flow.

The current input is denoted by ‘ x_t ’, and the previous step’s output is denoted by h_{t-1} , Eq. (20)

- **Update Gate (UG):** The UG ‘ z_t^j ’ is projected by
$$z_t^j = \sigma(W_z x_t + U_z h_{t-1}^j) \tag{20}$$

The unit uses UG ‘ z_j, t ’ to determine how much data it must update.

- **Reset Gate (RG):** This Eq. (21) assesses the
$$RG_j * t : r_t^j = \sigma(W_r x_t + U_r h_{t-1}^j) \tag{21}$$

The last stage is j data. RG determines the data that must be overlooked at the last moment. A value close to 0 is ignored in the memory content state, but a value close to 1 is retained in the memory content state when the value is close to 1. On the other hand, the Weight Matrix (WM) is not the same.

2.8 Determine the Current Memory Content

As in the previous gates, the Hadamard product is measured by multiplying the corresponding elements of r_t and h_{t-1} along with the WM. Because r_t is composed of vectors from 0 to 1, it is vital to use the Hadamard product to determine how much data is lost in the current memory content when r_t is used. During the RG, it was explained that a value close to 0 indicates lost information, while a value close to 1 indicates remembered information. To scale down the results to a scale of -1 to 1, the 2 bits of data are combined with the \tanh activation function, Eq. (22)

$$\tilde{h}_t^j = \tanh(W_h x_t + U(r_t \cdot h_{t-1}^j)) \tag{22}$$

where there is multiplication on an elemental basis and r_t is a set of RG. In effect, the RG causes the unit to read the initial symbol in input order when it is turned off (r_t^j , close to 0), thus agreeing to forget the earlier computed state.

2.9 Determine the Information Retained by the Hidden Layer at the Current SoC Cycle

To conclude, the network estimates the h_t vector, which maintains data from the current state and passes it to the next cell. When the 1st item of the formula is the UG, for

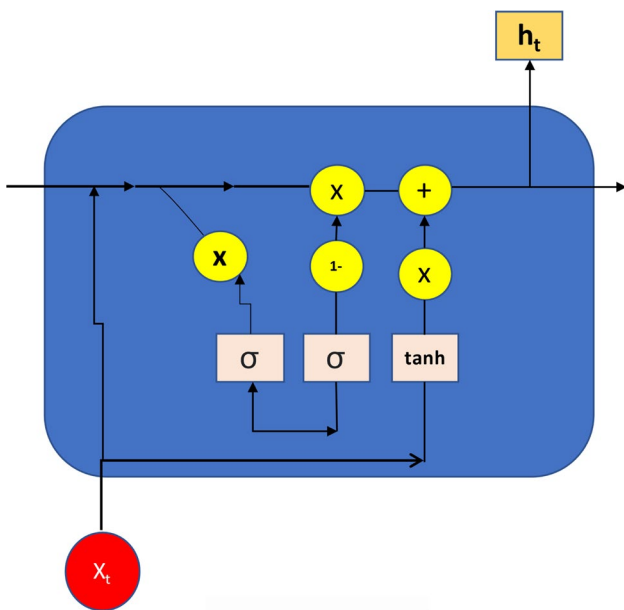


Fig. 7 GRU structure

example, it determines how much information from the previous moment is included in the formula h_{t-1} requirements to be stored in the remote unit h_t at this time. Likewise, information that ought to be forgotten can be summarized by $(I-z_t^*j)$ and updated by the current memory content as the 2nd portion of the formula. Because of this, the restructured door found what content to put in the current memory and hide the layer, Eq. (23)

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_t^j \tag{23}$$

The CNN module is now operational. CNN is an ANN that excels at processing data in multiple dimensions. Its applications include image and text recognition and text classification. In the ESS, there are many AI-embedded sensors and devices. The SM was created to preserve the spatial information of data collected in the ESS by smart sensors and devices, and the SM data is based on sensor location and time sequence. The SM is shown in the following format, Eq. (24)

$$X = \begin{bmatrix} X_1(1) & X_1(2) & \dots & X_1(n) \\ X_2(1) & X_2(2) & \dots & X_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ X_k(1) & X_k(2) & \dots & X_k(n) \end{bmatrix} \tag{24}$$

where ‘ k ’ stands for the ‘ k ’ smart sensor, ‘ n ’ stands for the ‘ n ’ time sequence, and $X_k(n)$ stands for the data collected by the ‘ k ’ smart sensor at ‘ n ’ time. However, because the NASA dataset was reengineered to represent SoC-based data features, this model reconstructed the matrix using SoC reengineered dataset, Eq. (25)

$$X_{SoC} = \begin{bmatrix} X_{SoC1}(1) & X_{SoC1}(2) & \dots & X_{SoC1}(n) \\ X_{SoC2}(1) & X_{SoC2}(2) & \dots & X_{SoC2}(n) \\ \vdots & \vdots & \ddots & \vdots \\ X_{SoCk}(1) & X_{SoCk}(2) & \dots & X_{SoCk}(n) \end{bmatrix} \tag{25}$$

To extract the SoH, CNN was used to develop the SoC Matrix (SCM). The model of CNN is depicted in Fig. 8. Many 2-D SCM were first stacked into 3-D matrix blocks and then used in convolution. The pooling operation combines the outputs of the convolution operation, which are used to create a highly abstract feature. As a result of the pooling operation, the depth of the input matrix does not change; however, the size of the matrices and the number of

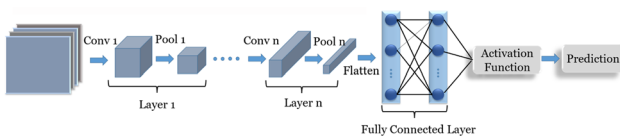


Fig. 8 CNN framework

nodes in the NN are reduced, lowering the parameters. After recurring convolution and pooling functions, the highly abstract feature was derived and flattened to a 1-D vector linked to the Fully Connected Layer (FCL). The weight and bias parameters of the FCL can then be computed iteratively. Finally, the activation function’s output is used to generate prediction results.

2.9.1 ARIMA Model

Box and Jenkins first introduced ARIMA in 1976 in a book that drew much interest from researchers working on prediction-related research at the time. As a result, this method is still one of the most reliable data processing and operational prediction models in many fields. ARIMA divides time series into three categories based on three key characteristics: using Auto-Regressive (AR) terms to model past process data. The differences that must be modelled for the process to be stationary are represented by integrated (I) terms. In the process, the Moving Average (MA) is a tool for controlling past noise information. The term AR provides a representation of the series based on its ‘ p ’ existing studies, Eq. (26):

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} = \sum_{j=1}^p \phi_j X_{t-j} \tag{26}$$

where $(\phi_j)j$ denotes the AR coefficients, and ‘ p ’ denotes the number of observations made previously required to predict the current value of the series. It’s not about looking back at what happened before. The MA component in a regression model (innovation processes) shows a rolling average of previous errors, Eq. (27)

$$X_t = \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} = \sum_{j=1}^q \theta_j \varepsilon_{t-j} \tag{27}$$

where $(\theta_j)j$ denotes MA coefficients, $(\varepsilon_{t-j})j$ denotes previous innovation processes at a time ‘ t ’, and ‘ q ’ denotes the number of times the MA has increased, used to forecast the current value of the series. The integrated part (the I in ARIMA) is defined by the parameter ‘ d ’, which is the order of differentiation, Eq. (28)

$$X_t^* = X_t - X_{t-1} - \dots - X_{t-d} \tag{28}$$

In most cases, using $d=1$ is sufficient. The ARMA process, for which $d=0$, is a specific method class. An ARIMA model is written using the ‘Backward’ operator $BX_t = X_{t-1}$, Eq. (29)

$$\left(I - \sum_{j=1}^p \phi_j B^j \right) (I - B)^d X_t = \left(I + \sum_{j=1}^q \theta_j B^j \right) \varepsilon_t \tag{29}$$

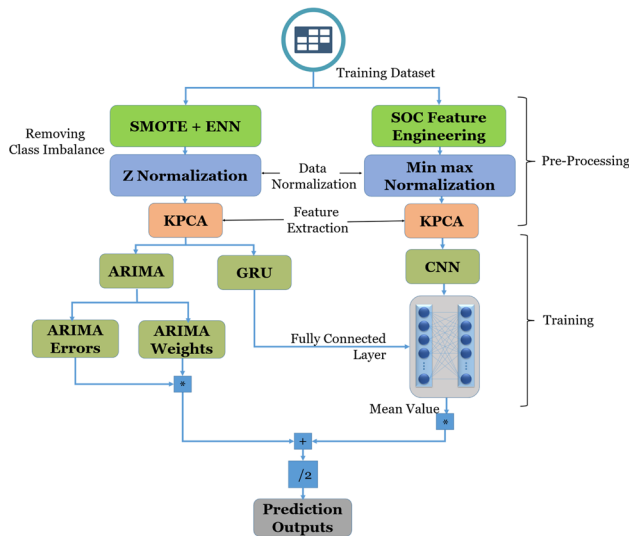


Fig. 9 Proposed ARIMA + GRU + CNN Deep Learning Model

3 Machine Learning Process

The proposed method for predicting a battery’s SoH is depicted in Fig. 9, which includes three stages: preprocessing of data, training of models, and evaluation of performance. The raw data of the batteries was normalised to eliminate missing and abnormal results. The time series and SoC-engineered data are subjected to the proposed ML. After the time-based data was cleaned, the current battery properties were extracted and transformed into SoC-based statistical information. The feature data was used to create training datasets. The data set for training was split into data for training and data for validation during the model stage.

Linear relationships (LR) and non-linear relationships (NLR) exist in different time-series models. In time series paradigms, the ARIMA model is beneficial when modelling LR. It is, however, insufficient to model NLR. The GRU model, on the other hand, can model LR and NLR, though not all datasets yield the same result. In order to overcome these challenges and reach the highest level of predictive accuracy, the HLM was invented, which relies on the differential modelling principle of the NLR and LR elements of the system.

Over time, multiple hybrid data time series assessment procedures have been presented and have made significant progress. Multiple-HLM algorithms are believed to provide better prediction and performance than creative learning algorithms. The concept of supervised learning algorithms is used to create these hybrid models. The main goal of this HLM is to increase the model’s diversity while improving prediction results. In numerous studies, many hybrid models have been reported. But there are no available sources to predict time-series data with SoC

data. Based on this motivation, this work has created the ARIMA + GRU + CNN + HLM for predicting the SoH of a battery using time series and SoC parameters. Our recommended ARIMA + GRU + CNN + HLM process has been outlined. GRU and LSTM are RNNs designed to capture long-term dependencies in sequential data. However, GRUs have a simpler architecture than LSTMs, with fewer parameters and less computational complexity.

In some cases, GRUs have been shown to perform similarly to LSTMs while requiring less training time and memory. In the context of the ARIMA + GRU + CNN + HLM for battery SoH prediction, this work used GRUs instead of LSTMs due to the simpler architecture and computational efficiency. Using GRUs instead of LSTMs, the model could process more data and achieve higher accuracy with less training time and resources. The trained model has been evaluated using the data set in the performance evaluation phase, and its output was determined by measuring the MAE, MAPE, and MSE. The validation data was used to fine-tune the hyperparameters of the model.

3.1 Data Preprocessing

To obtain relevant data for DL, data preprocessing had to be done. In these cases, the data was preprocessed to make it suitable for DL cleaning. After data cleaning, the input data was presented to the learning model in two formats: the actual time series dataset and the SoC-based data transformed from the actual time series data. The SMOTE + ENN + HLM addresses the class imbalance problem in the time series data. The application of MIN–MAX is flawed for time series data normalisation. Time series prediction using the MIN–MAX normalisation method is flawed because out-of-sample data sets have unclear MIN and MAX values. This research uses the *z-score* normalization model to get around this. They are normalizing all values in a dataset so that the mean is 0 and the standard deviation is done with the Z-score.

From Eq. (30) to perform z-score normalization on each value in a dataset:

$$X_{New} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{SD}(X)} \tag{30}$$

X_{New} = The new value from the normalized results; X = Old value; μ = Population mean; σ = Standard deviation value.

After data cleaning, SoC-based data was transformed into raw data based on time series, and data from SoC automatically extracted features. Equation (31) is used to express the feature format. The scale of the SoC-based dataset is not the same. As a result, this work used

MIN–MAX normalization to normalize the SoC-based dataset:

$$z_{nm}^k = \frac{x_{nm}^k - \text{Min}(x_n)}{\text{Max}(x_n) - \text{Min}(x_n)} \quad n \in \{1, \dots, 5\}, m \in \{1, \dots, s\} \quad (31)$$

Here, x_n is a collection of all charging cycles’ n rows in Eq. (31), where ‘ k ’ is the cycle count, and ‘ m ’ is the m testing point. As shown below in Eq. (32), used MIN–MAX normalization to normalize the capacity:

$$c^k = \frac{C^k - \text{Min}(C)}{\text{Max}(C) - \text{Min}(C)} \quad (32)$$

where ‘ k ’ stands for the number of cycles, and ‘ C ’ stands for the number of cycles—charging cycle capacities. After the data was normalized, it was split into two groups: a dataset for fitting model parameters and a test dataset for assessing the models that came out of it.

3.2 ARIMA + GRU + CNN

The ARIMA + GRU + CNN model is a HLM that combines the linear ARIMA, GRU, and CNN models into one. Researchers originally predicted CO₂ emission costs using the ARIMA model and determined them by calculating the ARIMA model’s residual. The ARIMA model’s residuals are then assumed to be spatially and temporarily dependent between observations. This paper extracts Spatio-temporal features from the ARIMA model’s residuals using the GRU model and CNN layers’ unique spatial and temporal modelling capabilities. It was proposed that the ARIMA + GRU + CNN modules, which are good at processing time-sequence and SoC data, be combined to make the hybrid NN.

The ARIMA + GRU + CNN modules comprise the proposed ARIMA + GRU + CNN + HLM (Fig. 4). Time series data and SoC data from NASA’s battery dataset are input variables, and the output signals are predictions of SoH. The CNN module is particularly adept at handling the SoC data format. The CNN module extracts local features from SoC data using shared weights and local connections, then

uses convolution and pooling layers to achieve adequate representation. Each CNN module convolution layer has convolution and a pooling function. After the pooling function, the SoC-HD data is converted to 1D data, and the CNN module’s output signals are linked to the FCL.

On the other hand, the ARIMA + GRU module primarily aims to capture long-term dependency, and it can use the memory cell to know and understand good data from historical information for an extended time, while the forget gate will forget the irrelevant information. The inputs to the ARIMA + GRU module are time-classification data; the ARIMA + GRU module includes many gated recurrent units, and their results are all connected with FCL. Finally, load predictions are determined by averaging the mean values of all neurons in the FCL. Figure 10 depicts the GRU + CNN method’s flow chart.

4 Experimental Analysis

The training and test validation data outnumber the validation data by two. The simulation was created with PyTorch 1.7 and then run on a GeForce GTX 1080Ti. The data used included voltage, current, and SoC values. All attained data is normalised to reduce training variability and test speed. The Bayesian optimization algorithm was implemented to optimize this proposed HLM, and the primary learning degree was set at 10⁻⁴.

4.1 Performance Metrics

This paper must compare the predicted SoH to the actual SoH results from the experiment to assess the predicted SoH results for the adopted models. As a result, performance metrics are evaluated using various metrics.

4.1.1 RMSE

This is the square root of all errors’ square meaning. However, RMSE is a good measure of precision. However, it is used for comparing model predictions to real-world data, not between variables, Eq. (33)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\text{SoH}_{\text{Predict}} - \text{SoH}_{\text{True}})^2} \quad (33)$$

4.1.2 MAPE

It normalises absolute error across data scales. MAPE is the absolute error/actual value. Equation (34).

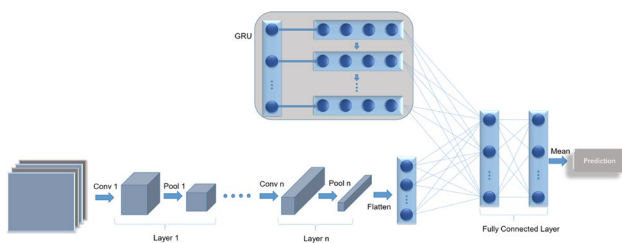


Fig. 10 GRU-CNN connection architecture

Table 2 Benchmark Models

Model	Processing type	Network	Reference
#1	SoC	CNN	[36]
#2	Time series	LSTM	[37]

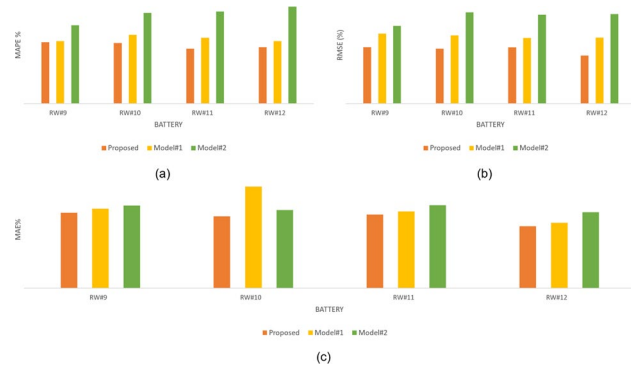


Fig. 11 a SoH prediction accuracy for batteries #RW9 to #RW12; b MAPE to compare a proposed model to other models; c RMSE compares the proposed model to other models

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{SoH_{Predict} - SoH_{True}}{SoH_{True}} \right| \quad (34)$$

4.1.3 MAE

Simple error measure MAE is standard. The MAE correlates with the results. Comparing data from different scales, Eq. (35).

$$MAE = \frac{1}{N} \sum_{i=1}^N | SoH_{Predict} - SoH_{True} | \quad (35)$$

4.2 Result and Discussion

The proposed model’s SoH prediction results for battery group 3 (RW #9 to #12) are compared against those of the proposed models (Table 2).

The HLM is trained for 50 cycles, and each battery dataset’s resultant SoH prediction accuracy is presented. The proposed HLM’s prediction accuracy for battery RW#9 is presented in Fig. 11a; the proposed ARIMA + CNN + GRU + HLM, which is based on SoC and Timeseries, has a consistent prediction accuracy after 50 cycles of training, while the Model#1 and Model#2 have equal accuracy rates. However, Model#1 outperforms Model#2 after 80 cycles. For the next battery, RW#10, the prediction accuracy of the proposed model is way better than the other two models: out of the

two models, Model#1 fares well against Model#1 right after 70 cycles of training (Figs. 11b, c).

This work assessed the MAE, RMSE, and MAPE based on the SoH prediction for a more straightforward comparison. Results proved that the proposed model was better at predicting SoH than Models #1 and #2, as shown in the following graphs. The proposed models with SoC + time-series-based processing had fewer prediction errors than other variants. According to these study results, the proposed ARIMA + CNN + GRU + HLM with SoC and time-series-based processing significantly outperforms complicated systems like CNNs (Model #1) and LSTMs (Model #2) in limited data. Figure 11 shows the MAPE performance of the proposed model against the other two models. The HLM based on SoC and Timeseries processing has the lowest MAPE score among other models, with the lowest score of 2.803 recorded for the battery 11 dataset. The CNN-trained SoC-based model (Model#1) had the next lowest MAPE of all battery data, with a minor average MAPE of 3.197. The LSTM (Model #2) using Timeseries-based sets had a minimum MAPE of 4.013.

Results of RMSE for all three models for the four batteries. The proposed HLM has the lowest RMSE in all batteries, with the lowest of 1025.46 for battery 12 and the highest of 1203.25 for battery 9. The CNN has the lowest value of 1398.35 for battery 11 and the highest value of 1492.34 for battery 9. The LSTM has the lowest score of 1658.21 for battery 9 and the highest RMSE score of 1946.76 for battery 10—the MAE results of all three models for the four battery datasets. The results show that the proposed model has a better MAE value for all the battery data, with the lowest score of 1.586 for battery 12. Model#1 has the lowest MAE of 1.673 for battery 12 and the highest of 2.601 for battery 10. Finally, Model#2 has the lowest MAE of 1.943 for battery 12 and 2.123 for battery 11. This research study revealed that compared with learning models like CNN and LSTM, a HLM with a SoC and time series-based data preprocessing phase could reliably predict the SOH.

5 Conclusion

This study proposes a novel learning model for forecasting the SoH of Li-ion batteries that integrates ARIMA + GRU + CNN. Numerous previous studies have concentrated on forecasting SoH using time series data. However, the limitation of such models is that they do not consider the battery’s inherent qualities. However, incorporating such attributes while ignoring the time attribute will result in inaccurate prediction. The proposed model can also bring time series and data from the SoC domain to the table for processing. The proposed preprocessing was based on

time series and the relative state of charge, which was computed using the current integration method during charging. The correlation coefficient among datasets and useable capacities was calculated to compare the proposed method to the typical preprocessing method based on consistent time intervals. The advantage of the proposed HLM is evident when compared to generic time-based datasets, SoC-based datasets processed using the suggested method showed a stronger connection with usable capacity, according to the results of the correlation analysis.

Additionally, this work compared the proposed model's efficiency to that of other developed models based on CNN + LSTM. The results indicate that the recommended model has a higher predictive accuracy with minor fever errors as measured by MAPE, RSME, and MAE compared to previous models. Based on our research findings, the proposed method might be used to increase the accuracy of devices with constrained computational power. Thus, SoH prediction on hardware platforms with limited resources can be made using the findings from this research study.

Author Contribution VM: Writing-Review & Editing; SS: Investigation, Writing-Original Draft, Conceptualization, Methodology, Formal analysis; RA: Validation, Software, Data Curation; MA: Supervision, Funding.

Funding The researchers would like to acknowledge the Deanship of Scientific Research at Taif University for funding this work.

Data Availability Not applicable.

Code Availability Not Applicable.

Declarations

Conflict of interest No Conflict of Interest.

References

1. Global energy and CO₂ emissions in 2020—global energy review 2020—analysis. IEA (2020). <https://www.iea.org/reports/global-energy-review-2020/global-energy-and-co2-emissions-in-2020>.
2. Martin (2019) Climate change. In: United Nations sustainable development. <https://www.un.org/sustainabledevelopment/climate-change/>. Accessed 23 Jan 2021
3. Alattar AH, Selem SI, Metwally HM, Ibrahim A, Aboelsaud R, Tolba MA, El-Rifaie AM (2019) Performance enhancement of micro grid system with SMES storage system based on mine blast optimization algorithm. *Energies* 12(16):1–23
4. Martin, Energy. In: United Nations sustainable development. <https://www.un.org/sustainabledevelopment/energy/>
5. Lipu MSH, Mamun AA, Ansari S, Miah MS, Hasan K, Meraj ST, Abdolrasol MGM, Rahman T, Maruf MH, Sarker MR, Aljanad A, Tan NML (2022) Battery management, key technologies, methods, issues, and future trends of electric vehicles: a pathway toward achieving sustainable development goals. *Batteries* 8(9):119
6. Tran MK, Fowler M (2020) A review of lithium-ion battery fault diagnostic algorithms: current progress and future challenges. *Algorithms* 13(3):62
7. Texas Instruments. bq27220 Single-Cell CEDV Fuel Gauge (2016)
8. Dai H, Jiang B, Hu X, Lin X, Wei X, Pecht M (2021) Advanced battery management strategies for a sustainable energy future: multilayer design concepts and research trends. *Renew Sustain Energy Rev* 138:110480
9. Li Y, Sheng H, Cheng Y, Stroe DI, Teodorescu R (2020) State-of-health estimation of lithium-ion batteries based on semi-supervised transfer component analysis. *Appl Energy* 277:115504
10. Tian H, Qin P, Li K, Zhao Z (2020) A review of the state of health for lithium-ion batteries: research status and suggestions. *J Clean Prod* 261:120813
11. Sun C, Lin H, Cai H, Gao M, Zhu C, He Z (2021) Improved parameter identification and state-of-charge estimation for lithium-ion battery with fixed memory recursive least squares and sigma-point Kalman filter. *Electrochim Acta* 387:138501
12. Zhang L, Fan W, Wang Z, Li W, Sauer DU (2020) Battery heating for lithium-ion batteries based on multi-stage alternative currents. *J Energy Storage* 32:101885
13. Shu X, Shen S, Shen J, Zhang Y, Li G, Chen Z, Liu Y (2021) State of health prediction of lithium-ion batteries based on machine learning: advances and perspectives. *iScience* 24(11):103265
14. Tian H, Qin P, Li K, Zhao Z (2020) A review of the state of health for lithium-ion batteries: research status and suggestions. *J Clean Prod* 261:120813
15. Qian K, Huang B, Ran A, He Y-B, Li B, Kang F (2019) “State-of-health (SOH) evaluation on lithium-ion battery by simulating the voltage relaxation curves. *Electrochim Acta* 303:183–191
16. Chen Y, Kang Y, Zhao Y, Wang L, Liu J, Li Y, Liang Z, He X, Li X, Tavajohi N, Li B (2021) A review of lithium-ion battery safety concerns: The issues, strategies, and testing standards. *J Energy Chem* 59:83–99
17. Zhang L, Fan W, Wang Z, Li W, Sauer DU (2020) Battery heating for lithium-ion batteries based on multi-stage alternative currents. *J Energy Storage* 32:101885
18. Jiang B, Dai H, Wei X (2020) Incremental capacity analysis based adaptive capacity estimation for lithium-ion battery considering charging condition. *Appl Energy* 269:115074
19. Tian H, Qin P, Li K, Zhao Z (2020) A review of the state of health for lithium-ion batteries: research status and suggestions. *J Clean Prod* 261:120813
20. Andrenacci N, Vellucci F, Sglavo V (2021) The battery life estimation of a battery under different stress conditions. *Batteries* 7:88
21. Yang S, Zhou S, Hua Y et al (2021) A parameter adaptive method for state of charge estimation of lithium-ion batteries with an improved extended Kalman filter. *Sci Rep* 11:5805
22. Wang D, Yang F, Zhao Y, Tsui KL (2017) Battery remaining useful life prediction at different discharge rates. *Microelectron Reliab* 78:212–219
23. Ren L, Zhao L, Hong S, Zhao S, Wang H, Zhang L (2018) Remaining useful life prediction for lithium-ion battery: a deep learning approach. *IEEE Access* 6:50587–50598

24. Vazquez-Canteli JR, Nagy Z (2019) Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Appl Energy* 235:1072–1089
25. Wei J, Dong G, Chen Z (2018) Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression. *IEEE Trans Industr Electron* 65:5634–5643
26. Zhang Y, Xiong R, He H, Pecht M (2018) Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Trans Veh Technol* 67:5695–5705
27. Vatani M, Vie PJ, Ulleberg Ø (2018) Cycling lifetime prediction model for lithium-ion batteries based on artificial neural networks. In: *Proceedings of IEEE PES innovative smart grid technologies conference Europe, Sarajevo, Bosnia and Herzegovina, Piscataway, NJ*, pp 1–6
28. Mirzapour F, Lakzaei M, Varamini G, Teimourian M, Ghadimi N (2019) A new prediction model of battery and wind-solar output in hybrid power system. *J Ambient Intell Hum Comput* 10:77–87
29. Titos M, Bueno A, Garcia L, Benitez MC, Ibanez J (2019) Detection and classification of continuous volcano-seismic signals with recurrent neural network. *IEEE Trans Geosci Remote Sens* 57:1936–1948
30. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444
31. Bektas O, Jones JA, Sankararaman S, Roychoudhury I, Goebel K (2019) A neural network filtering approach for similarity-based remaining useful life estimation. *Int J Adv Manuf Technol* 101:87–103
32. Zhang Y, Xiong R, He H, Pecht MG (2018) Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Trans Veh Technol* 67(7):5695–705
33. Ling L, Wei Y (2021) State-of-charge and state-of-health estimation for lithium-ion batteries based on dual fractional-order extended kalman filter and online parameter identification. *IEEE Access* 9:47588–47602
34. Li J, Landers RG, Park J (2020) A comprehensive single-particle-degradation model for battery state-of-health prediction. *J Power Sources* 456:227950
35. Bonfitto A, Ezemobi E, Amati N, Feraco S, Tonoli A, Hegde S (2019) State of health estimation of lithium batteries for automotive applications with artificial neural networks. In: *AEIT international conference of electrical and electronic technologies for automotive, Politecnico di Torino, Lingott*, pp 21–25
36. Jo S, Jung S, Roh T (2021) Battery state-of-health estimation using machine learning and preprocessing with relative state-of-charge. *Energies* 14(21):7206
37. Yayan U, Arslan AT, Yucel H (2021) A novel method for SOH prediction of batteries based on stacked LSTM with quick charge data. *Appl Art Intell* 35(6):421–439
38. Bole B, Kulkarni C, Daigle M (2014) Randomized battery usage data set, NASA AMES prognostics data repository. In: *Mountain view, CA*
39. Kim IS (2009) A technique for estimating the state of health of lithium batteries through a dual-sliding-mode observer. *IEEE Trans Power Electron* 25:1013–1022
40. Ng KS, Moo CS, Chen YP, Hsieh YC (2009) Enhanced coulomb counting method for estimating state-of-charge and state-of health of lithium-ion batteries. *Appl Energy* 86:1506–1511
41. Zhang J, Lee J (2011) A review on prognostics and health monitoring of li-ion battery. *J Power Sources* 196:6007–6014
42. Kirchev A (2015) *Electrochemical energy storage for renewable sources and grid balancing*. Elsevier, Amsterdam, pp 411–435
43. Tomek I (1976) Two modifications of CNN. *IEEE Trans Syst Man Commun* 6:769–772

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Varatharaj Myilsamy is an Associate Professor in the Department of Electrical and Electronics Engineering at V.S.B College of Engineering Technical Campus, Coimbatore, Tamil Nadu, India. He obtained his Ph.D. in 2020 from Anna University in Power Generation. He completed his M.E. in 2007 from Bannari Amman Institute of Technology with a specialization in Applied Electronics and obtained his B.E. in 2005 from Coimbatore Institute of Technology in Electrical and Electronics Engineering.

He has teaching experience of about 13 years and industrial experience of about 4 years. His area of interest includes Renewable Energy, Information and Communication Engineering, Power Electronics, and Power Systems.



Sudhakar Sengan (Member, IEEE) received the M.E. degree from the Faculty of Computer Science and Engineering, Anna University, Chennai, Tamil Nadu, India, in 2007 and the Ph.D. degree in information and communication engineering from Anna University, Chennai, Tamil Nadu, India, in 2014. He has 20 years of experience in teaching, research, and industry. He is a Professor and the Director of international relations at the Department of Computer Science and Engineering, PSN

College of Engineering and Technology (Autonomous), Tirunelveli, Tamil Nadu, India. He guided more than 100 Projects for the UG and PG streams. He is the Recognized Research Supervisor at Anna University, under the Information and Communication Engineering Faculty. He has published papers in 175 international journals, 20

international conferences, and 10 national conferences. He has published three textbooks for Anna University's, Chennai Syllabus. He has filed 20 Indian and 3 international patents in various fields of interest. His research interests include security, MANET, the IoT, cloud computing, and machine learning. He is a member of professional bodies like MISTE, MIEEE, MIAENG, MIACSIT, MICST, MIE, and MIEDRC. He received the Award of Honorary Doctorate (Doctor of Letters-D. Litt.) from International Economics University, SAARC Countries in Education and Students Empowerment, in April 2017.



Roobaea Alroobaea received bachelor's degree (Hons.) in computer science from King Abdulaziz University (KAU), Saudi Arabia, in 2008, and the master's degree in information systems and the Ph.D. degree in computer science from the University of East Anglia, U.K., in 2012 and 2016, respectively. He is an Associate Professor with the College of Computers and Information Technology at Taif University, Saudi Arabia. His research interests include

human-computer interaction, software engendering, cloud computing, the Internet of Things, artificial intelligence, and machine learning.



Majed Alsafyani received the bachelor's degree (Hons.) in computer science from the University of Hertfordshire (UH), U.K., in 2013 and the master's degree (Merit) in Advance computer science from the University of Hertfordshire (UH), U.K., in 2014, and the Ph.D. degree in computer science from the University of Hertfordshire, U.K., in 2015 and 2020, respectively. He is an Assistant Professor at the College of Computers and Information Technology, Taif University, Saudi Arabia. His research

interests include image processing, machine learning, Internet of Things applications, artificial intelligence, and software engineering.