2

by Vaibhav S

Introduction

What is Li-ion battery?

A lithium-ion (Li-ion) battery is a high-tech battery that employs lithium ions as an important component of its electrochemistry. Lithium atoms in the anode are ionized and separated from their electrons during a discharge cycle. The lithium ions go from the anode via the electrolyte to the cathode, where they recombine with their electrons and electrically neutralize. The lithium ions are tiny enough to pass through a micro-permeable separator that separates the anode and cathode. Because of the small size of lithium (third only to hydrogen and helium), Li-ion batteries can have very high voltage and charge storage per unit mass and unit volume. Electrodes in Li-ion batteries can be made of a variety of materials. The most frequent combination is lithium cobalt oxide (cathode) and graphite (anode), which is found in portable electronic devices like cellphones and laptop computers. Lithium manganese oxide (used in hybrid and electric vehicles) and lithium iron phosphate are two other cathode materials. As an electrolyte, ether (a class of organic chemicals) is commonly used in lithium-ion batteries.

Applications of Li-ion batteries

Li-ion batteries come in a variety of shapes and sizes. As a result, they are the ideal choice for power requirements regardless of system size. Furthermore, lithium-ion batteries provide power options across the board, from energy storage to portable energy. The following are some of the most prevalent applications for lithium-ion batteries:

- Electric Vehicles
- Mobile phones, laptop computers, and other regularly used consumer electronics.
- UPS (Uninterruptible Power Supply)
- Energy Storage Systems

Current scenario of Li-ion batteries

While lithium-ion batteries (LIBs) were initially created for portable electronics, they are now found in a wide range of applications, including electric cars, power tools, medical equipment, smart watches, drones, satellites, and utility-scale storage. As battery usage grows, so do the individual requirements, with a widening range of battery designs and sizes to meet each application. Current LIBs are suitable for frequency regulation, short-term storage, and micro-grid applications, but cost and mineral resource constraints preclude their broad use on the grid. There are numerous approaches, with no clear winners or preferred paths to the aim of producing a battery for widespread grid use. LIBs today are highly optimized, lasting months to years, with some projected to last decades. Given that several of the materials operate outside of their thermodynamic stability windows, this is a significant accomplishment. On charging, the anodes (negative electrodes) are litigated to near-Li metal potentials (0.08 V versus Li/Li^+), where no electrolytes are stable. The battery instead survives by generating a passivation layer, also known as solid-electrolyte interphase (SEI), which prevents further electrolyte deterioration. On the cathode, the breakdown of the electrolyte salts reduces Al current collector corrosion, resulting in a stable passivation layer. Cathode materials have been optimized to reduce oxygen loss at higher temperatures, so preventing 'thermal runaway,' and to endure the mechanical forces associated with Li removal and insertion. While some breakthroughs were coincidental, the majority were the result of significant and global research efforts, resulting in a highly optimized system suitable for a wide range of applications. As a result, our current commercial systems contain materials with energy densities that are approaching their fundamental limits, i.e., further lithium removal from the cathode results in irreversible structural transformations or oxygen loss, while no vacancies in the lattice remain to accommodate more Li ions on the anode. Surface coatings, electrolyte additives, and morphological optimization are allowing batteries to operate at greater voltages by making separators and current collectors thinner.

Problems in Li-ion Batteries

Ageing or degradation: One of the most significant lithium-ion battery disadvantages for consumer gadgets is that lithium-ion batteries age. This is dependent not only on the time or calendar, but also on the amount of charge discharge cycles that the battery has gone through. Often, batteries may only tolerate 500-1000 charge-discharge cycles before their capacity degrades. This figure is rising as lithium-ion technology advances, but batteries will eventually need to be replaced, which can be a problem if they are integrated in equipment.
Lithium-ion batteries age whether they are in use. Regardless of utilization, the decline in capacity has a time component.

Importance of AI in predicting the life of a Li-ion batteries

- Accurate prediction of remaining battery life: AI algorithms can analyze data from Li-ion batteries, such as current and voltage measurements, temperature, and other environmental factors, to predict how much longer the battery will last. This can help users avoid unexpected shutdowns, improve safety, and optimize battery performance.

- Personalized battery life prediction: AI can be used to develop personalized battery life prediction models that consider individual usage patterns, charging habits, and other factors that can impact battery life. This can help users better understand how to prolong the life of their batteries and make more informed decisions about their energy usage.

- Battery health monitoring: AI can be used to monitor the health of Li-ion batteries, detecting abnormalities such as cell degradation, temperature changes, and other factors that can affect battery performance. This can help identify potential issues before they become serious, allowing for preventative maintenance, and reducing the risk of failure.

- Optimization of battery management systems: AI can be used to optimize the performance of battery management systems (BMS), which are responsible for managing the charging and discharging of Li-ion batteries. By analyzing data from BMS sensors and other sources, AI algorithms can optimize charging rates, temperature control, and other factors that impact battery life.

Literature

Different types of Literature Datasets used:

NASA Dataset

This data collection was gathered at the NASA Ames Prognostics Centre of Excellence from a specially constructed battery prognostics testbed (PCoE). At various temperatures, Li-ion batteries were put through three different operational profiles (charge, discharge, and electrochemical impedance spectroscopy). The battery voltage was discharged at various current load levels until it reached certain voltage thresholds. Some of these thresholds were set lower in order to create the deep discharge ageing effects than the OEM's suggested threshold (2.7 V). Batteries age more quickly when they are repeatedly charged and discharged. After the batteries met the end-of-life (EOL) threshold of 30% fade in rated capacity, the trials were terminated (from 2 Ah to 1.4 Ah). The data sets can be used for many different things. The data can be used to create prognostic algorithms because they are essentially a lot of Run-to-Failure time series. In particular, no two cells have the same state-of-life (SOL) at the same cycle index because of variations in depth-of-discharge (DOD), the length of rest periods, and inherent variability. In order to accurately anticipate Remaining Useful Life (RUL) in both End-of-Discharge (EOD) and End-of-Life (EOL) situations, it is important to be able to handle this uncertainty, which is reflective of actual usage.

MIT Dataset

The Toyota Research Institute (TRI) has released two sizable and user-friendly high-throughput cycling datasets in collaboration with MIT and Stanford. These datasets together comprise data for 357 (= 124 + 233) commercial LFP/graphite cells with a rated capacity of 1.1 Ah produced by A123 Systems (APR18650M1A).

The purpose of the first of these datasets (124 cells) was to investigate how quick charging techniques affect cell ageing. A common CC-discharge protocol and one of 72 alternative profiles of one- or two-step fast charging protocols were alternated for each cell. In a controlled atmosphere with a temperature of 30 °C, the cells were cycled. Between 150 and 2300 cycles, or from cycle 2 until a cell attained its EOL (80% SOH), data were logged. The dataset includes per-cycle measurements of capacity, internal resistance, and charge time in addition to in-cycle observations of temperature, current, voltage, charge, and discharge rates. The data is divided into three batches, which correspond to three independent blocks of experiments. A feature-based model is created in the companion paper using data from the first 100 cycles to forecast the EOL. After the dataset's publication, many further papers utilising this data have been published.

The second of these datasets (233 cells), which develops a strategy to quickly optimise fast charging processes, builds on the first. Once more, cells were cycled using a standard methodology for discharging in a temperature-controlled environment (30 °C). The dataset is divided into five batches of between 45 and 48 cells each, and these batches were tested sequentially. For the first batch, one of 224 different six-step charging protocols was randomly selected for each cell, and the cells were tested for 100 cycles before being used to train a model to predict the EOL using the data. The choice of charging techniques for the following batch of cells was guided by this forecast. The final batch was tested up until after the EOL while being tested against a number of various charging methods, as was done with the first four batches. With the exception of IR readings, the dataset's 124 cells include the same readings as the previous dataset. A CNN model trained on the first dataset was used to predict the IR, and this projected IR data can be found online in an attempt to retrieve the missing data.

Different types of data-pre-processing techniques used:

Feature Scaling

It is technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing. There are two common techniques used for feature scaling:

- Min-Max Scaling: This technique scales the features to a fixed range, typically between 0 and 1. It is calculated by subtracting the minimum value of the feature and dividing by the range of the feature.
- Standardization: This technique scales the features to have zero mean and unit variance. It is calculated by subtracting the mean value of the feature and dividing by the standard deviation of the feature.

By using feature scaling, machine learning models can more accurately analyse the data and make better predictions.

Outliers Excluding

An outlier is a data point that differs significantly from other data points in a dataset. In other words, it is an observation that lies an abnormal distance away from other values in a random sample from a population. There are various causes of outliers such as Data entry errors, measurement error, experimental errors, intentional, data processing errors, sampling errors, natural outlier, etc. Outliers can be removed from the dataset using some techniques:

- Z-score Method: This method involves calculating the z-score for each data point. The z-score measures the number of standard deviations away from the mean a data point is. Data points with a z-score greater than a certain threshold (usually 2.5 or 3) are considered outliers and can be removed from the dataset.
- Interquartile Range (IQR) Method: This method involves calculating the IQR, which is the difference between the 75th percentile (Q3) and the 25th percentile (Q1) of the data. Any data point that falls below Q1 - 1.5IQR or above Q3 + 1.5IQR is considered an outlier and can be removed from the dataset.

Denoising

Nosie refers to random variations or errors in the data that can distort or obscure the underlying information. It is the process of removing noise from data using various techniques. AI algorithms are trained to recognize patterns in noisy data and then use these patterns to remove the noise while preserving the underlying information.

Different types of feature extraction and selection technique:

Feature Selection:

Correlation analysis

It is a statistical technique which determines the strength and direction of the relationship between two or more variables. It is used to determine whether there is a statistical association between variables, and if so, how strong and in what direction that associations is. In AI, correlation analysis can be used to identify patterns in data, which can then be used to train machine learning model that can predict outcomes or make decisions.

Sequential Feature Selection

It is a greedy method that adds and removes features from a dataset in a sequential manner. It assesses each feature individually and chooses M features from N features based on individual scores; this method is known as naive sequential feature selection. In other words, sequential feature selection is a machine learning and data science technique for selecting a subset of characteristics from a larger set of features to improve the performance of a machine learning model. The technique involves selecting and eliminating features from the initial collection of features iteratively until the optimal subset of features is found.

Feature Extraction

Time Series Feature

Feature extraction is the method of improving machine learning by identifying qualities in data that aid in the resolution of a certain problem. Several time series analysis and decomposition techniques can be used to extract features from time series data. Moreover, characteristics can be obtained using sequence comparison techniques like dynamic temporal warping and subsequence finding techniques like motif analysis.

Internal resistance, local time series of current, voltage, and temperature.

In ageing datasets of Li-Ion batteries, there are two sorts of data: "historical" data and "local" time series. The first is about the worldwide evolution of features computed at each cycle. The

SOH of different batteries from the MIT dataset is plotted as a function of cycle count. The darker the curve, the longer the battery's lifetime. The internal resistance (IR) of the same batteries is plotted as a function of cycle count, and the colour of the curves varies accordingly. There is just one value every cycle for both SOH and IR. Historical Characteristics include SOH and IR (HF). The second form of data is a local time series of current, voltage, and temperature that change with the battery's use mode and are expressed as a function of time inside a specific cycle. The time history of the charging current for numerous given cycles of the same battery is shown in Fig. 3 from the MIT dataset. Unlike HF, this time there is an entire vector for one cycle.

Different types of models used:

Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of artificial neural network commonly used in deep learning for image and video processing. It is designed to automatically learn features from input data in a hierarchical manner. CNNs are composed of multiple layers that are responsible for different operations. The first layer is typically a convolutional layer that applies a set of learnable filters to the input data. These filters detect specific features, such as edges or corners, in the input data. The output of the convolutional layer is then passed through a non-linear activation function, such as the rectified linear unit (ReLU), to introduce non-linearity into the model. This is followed by a pooling layer that reduces the spatial size of the input by down sampling the feature maps. The output of the pooling layer is then fed into one or more fully connected layers that perform the final classification or regression task. During the training process, the weights of the filters and fully connected layers are adjusted through backpropagation to minimize the difference between the predicted output and the true output.

Long Short Term Memory

It is a kind of recurrent neural network. The output of the previous step is fed into the current step in RNN. Hochreiter and Schmidhuber created LSTM. It addressed the issue of RNN long-term dependency, in which the RNN cannot predict words stored in long-term memory but can make more accurate predictions based on current input. RNN does not provide efficient performance as the gap length rises. By default, LSTM can save the information for a long time. It is used for time-series data processing, prediction, and classification. It is designed designed to handle sequential data such as time series, audio, and text. Because LSTM networks can learn long-term dependencies in sequential data, they are well suited for applications like language translation, speech recognition, and time series forecasting.

Gaussian Process Regression

GPR is a probabilistic approach to regression, which means that it provides a probability distribution over possible function values at each point in the input space, rather than a single point estimate. In GPR, a Gaussian process is used to model the underlying function that maps inputs to outputs. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. The Gaussian process is defined by a mean function and a covariance function, which specify the expected value and covariance of the function at any two input points, respectively. During training, the GPR algorithm learns the mean and covariance functions from the training data, which can then be used to make predictions for new inputs. Given a set of training inputs and corresponding outputs, the algorithm estimates the mean and covariance functions that best fit the data. The covariance function also provides an estimate of the uncertainty of the predictions, which can be used to quantify the confidence in the predictions. GPR has several advantages over other regression algorithms, including the ability to handle noisy and sparse data, the ability to incorporate prior knowledge into the model, and the ability to provide uncertainty estimates for the predictions. It is commonly used in various fields such as engineering, finance, and biology, where accurate and reliable predictions are required.

Multi-Layer Perceptron

MLP (multi-layer perceptron) are artificial neural networks with three or more layers of perceptron. These layers are as follows: a single input layer, one or more hidden layers, and a single perceptron output layer. The data flows in a single direction, forward, from the input layers to the hidden layer(s) and then to the output layer. Backpropagation is a process in which the multi-layer perceptron receives feedback on the error in its predictions and adjusts its weights to make more accurate predictions in the future. Several machine learning algorithms, such as classification and regression, make use of MLP. They have been demonstrated to produce very accurate results, particularly for categorization difficulties.

Support Vector Regression

Support Vector Regression (SVR) is a machine learning approach that is commonly used for regression tasks. It is a Support Vector Machines (SVM) variation used for classification tasks. SVR works by locating a hyperplane in a high-dimensional space that divides the training data into two classes with a margin of error. In the case of regression, the goal is to discover a hyperplane that minimises the gap between the projected and actual outputs while staying under a certain tolerance. The SVR algorithm uses a kernel function to transfer the input data into a higher-dimensional feature space separated by a hyperplane. The kernel function calculates the similarity of any two inputs in the feature space, which is used to construct the hyperplane.

Random Forest Regression

Random Forest is an ensemble technique that can handle both regression and classification tasks by combining many decision trees and a technique known as Bootstrap and Aggregation, or bagging. The core idea is to use numerous decision trees to determine the final output rather than depending on individual decision trees. Random Forest's foundation learning models are numerous decision trees. We randomly select rows and features from the dataset to create sample datasets for each model. This section is known as Bootstrap.

Table: Literature table for SOC Prediction.

Sr. No.	DataSet	Methodology	Pre Processing Technique	Feature Extraction	Feature Selection Technique	Model/ Algorithm	Accuracy
1	124 Lithium ion batteries	Pre processing of the data		Current	Correlation analysis	Convolutional Neural Networks (CNNs)	MAE. = 90 for current and MAE = 115 for remaining cycles
		Resampling data with 1000 equidistant voltage steps		Voltage			
		Set up the framework Tensorflow 2.0		Temperature			
		Train and tune the model		Cycle count			
		Make Prediction					
2	Battery lifespan prediction (data.matr)	Input the data		ConvLSTM		Adaptive feature separable convolution(AFSC)	MAPE = 0.25% for 100 cycles starting point
		Create subnet1 using AFSC algorithm		AFSC		Convolutional Long short term memory(CLSTM) network	MAPE = 0.74% for 20 cycles starting point
		Create subnet2 using ConvLSTM					
		Train the model					
		Make predictions and calculate MAPE					
3	124 lithium–iron-phosphate (LFP)/graphite LIBs.	Train CNN model on training and validation dataset		DEM as mean function for GPR		CNN for life cycle prediction preliminary	MAPE = 5.1 for train
		Use CNN to preliminary est. cycle life				GPR for prediction of RUL at given EOM cycle	MAPE = 10 for primary test

		Choose LIB from dataset and identify DEM					MAPE = 11.7 for secondary test
		Take DEM as mean function of GPR					
4	Camegie Mellon University Dataset	Extract the features		Charge related features	Random Forest Regression	SVR	MAPE = 0.02
		Select the features		Discharge related feature		RF Regression	MAE = 1.33
		Develop the model and train the model on training set		Temperature related features	Bayesian algorithm	XG Boost	MAE = 1.39
		Test the model using model evaluation on test set				GPR	MAE = 1.48
						MLP	RMSE = 7.49
5	NASA DataSet and MIT Dataset	Extract the TSF and HF	Feature scaling	Historical features as Internal Resistance and SOH	Sequential feature selection(SFS)	Long Short Term Memory	MAE for NASA batteries = 1.6*10^-2
		Train the data on window XLSTM model	Outlieer excluding	Time Series Features			MAE for MIT batteries = 1.07*10^-2
		Make predictions of SOH	Denoising				

Methodology

SOC prediction method based on Ensemble Learning:

Ensemble Learning is a popular and commonly used machine learning technique in which number of independent models, often termed as based models are combined to produce an effective optimal predictive model.

Bagging Technique:

Bagging, also known as Bootstrap aggregating, is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model. The code reads in a dataset of battery performance during a trip and selects two features, 'Time [s]' and 'Velocity [km/h]', to predict the 'SoC [%]' (State of Charge) of the battery. The dataset is split into training and testing sets using train_test_split() function.

Then, a Bagging ensemble with a Random Forest as the base estimator is created using Bagging Regressor() function with base estimator as Random Forest Regressor() and 10 estimators. The model is then trained on the training data using fit() function and predictions are made on the testing data using predict() function. The root mean squared error (RMSE) is calculated using mean_squared_error() function.

Similarly, a Random Forest ensemble with 20 trees is created using Random Forest Regressor() function. The model is trained on the training data and predictions are made on the testing data. RMSE and other performance metrics such as mean absolute error (MAE) and R-squared (R2) score are calculated using the respective functions.

Finally, the code plots the output for visualization purposes. A plot of 'Time [s]' vs 'SoC [%]' is created using plot() function. Then, a subplot of Time vs SoC, Motor Torque vs SoC, Battery Temperature vs SoC, and Throttle vs SoC is created using subplots() function.

Result

2

ORIGINALITY REPORT

47%	34%	26%	35%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

9%

★ Submitted to Liverpool John Moores University

Student Paper

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		