Article

Lithium-Ion Battery Prognostics through Reinforcement Learning Based on Entropy Measures

Alireza Namdari [1], Maryam Asad Samani [2] and Tariq S. Durrani [3,*]

[1] Department of Industrial Engineering and Engineering Management, Western New England University, Springfield, MA 01119, USA
[2] Department of Electrical Engineering, Iran University of Science and Technology, Tehran 13114-16846, Iran
[3] Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, Scotland G1 1XW, UK
* Correspondence: t.durrani@strath.ac.uk; Tel.: +44-(0)-141-548-2540

Abstract: Lithium-ion is a progressive battery technology that has been used in vastly different electrical systems. Failure of the battery can lead to failure in the entire system where the battery is embedded and cause irreversible damage. To avoid probable damages, research is actively conducted, and data-driven methods are proposed, based on prognostics and health management (PHM) systems. PHM can use multiple time-scale data and stored information from battery capacities over several cycles to determine the battery state of health (SOH) and its remaining useful life (RUL). This results in battery safety, stability, reliability, and longer lifetime. In this paper, we propose different data-driven approaches to battery prognostics that rely on: Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), and Reinforcement Learning (RL) based on the permutation entropy of battery voltage sequences at each cycle, since they take into account vital information from past data and result in high accuracy.

Keywords: lithium-ion battery; prognostics; long short-term memory; ARIMA; reinforcement learning

1. Introduction

1.1. Lithium-Ion Batteries

Lithium-ion batteries, as the primary power source in electric vehicles, have attracted significant attention recently and have become a focus of research. It is assumed that lithium-ion batteries have the inherent potential for building future power sources for environmentally friendly vehicles [1].

Lithium-ion batteries are the best option for electrical vehicles due to their high-quality performance, capacity, small volume, light weight, low pollution, and rechargeability with no memory effect [2]. However, battery performance degrades when facing poor pavement conditions, temperature, and load changes. This leads to leakage, insulation damage, and partial short-circuits. Consequential situations can arise if these failures are not detected timeously [3,4]. As an example, several Boeing 787 aircraft caught fire because of lithium-ion battery failure in 2013, causing the airliners to be grounded [5]. Hence, it is necessary to detect performance degradations timeously and estimate future battery performance. This is where battery prognostics and health management (PHM) plays an important and vital role. PHM determines the battery state of health prediction (SOH) and battery remaining useful life prediction (RUL) of the product using possible failure information in the system, thus yielding improved system reliability and stability in the actual life-cycle of the battery.

Battery PHM and a battery management system (BMS) are important to ensure the reliable and safe functionality of energy storage units [6]. Battery RUL prediction, battery

SOH prediction, and battery capacity fade prediction are among the topics which have drawn more attention from researchers in the recent decade [7]. However, these tasks are very difficult, as battery degradation has a complex nature and numerous factors must be taken into consideration [8,9].

1.2. Entropy Measures

Entropy is a measurement metric for irregularities in time series data, and is used to quantify the stochastic process in data analyses [10]. It was first introduced in classical thermodynamics, and has applications in diverse fields such as chemistry and physics, biological systems, cosmology, economics, sociology, weather science, climate change research, and information systems. Entropy has expanded to far-ranging fields and systems. Shannon, Permutation, Renyi, Tsallis, Approximate, and Sample entropy measures are some of the conceptions of entropy regularly in use [11].

From the afore-mentioned entropies, permutation entropy (PE) is a simple and robust approach to calculating the complexity of a non-linear system using the order relations between values of a time series and assigning a probability to the ordinal patterns. The permutation entropy measure technique works flexibly; it is computationally efficient, and has a range of several thousand parameter values similar to Lyapunov exponents. PE is discussed in more detail in Reference [12]. In this study, PE of the discharge battery voltage sequences is calculated and used as an input to the proposed models.

1.3. ML and DL Techniques

Recently, Machine Learning (ML) and Deep Learning (DL) algorithms have found very significant and useful applications in research and practice. These concepts have been used to develop various models for predicting different characteristics in diverse fields. In general, ML and DL algorithms aim to capture information from past data, learn from that data, and apply what they have learned to make informed decisions. Therefore, the associated systems are not required to be broadly programmed in all aspects.

ML is used to synthesize the fundamental relationships between large amount of data to solve real-time problems such as big-data analytics and evolution of information [13]. DL, in turn, is able to process a large number of features and, hence, is preferred when computing huge datasets and unstructured data. DL facilitates analysis and extraction of important information from raw data by using computer systems. [14]. Different types of parameters with various quantities can be applied to the developed models as the input to obtain expected predictive variables as the output.

Deep Learning techniques, including Long Short-Term Memory (LSTM) [15] and Reinforcement Learning (RL) [16], can fit numerical dependent variables and have great generalization ability, and therefore, are applicable to battery data. The LSTM algorithm, a Deep Learning algorithm with multiple gates, performs on the basis of updating and storing key information in the time series data [15], and is applicable to battery prognostics. The RL algorithm, on the other hand—as one of the latest Deep Learning methods and tools—has the capability of creating a simulation of the whole system and making intelligent decisions (i.e., charge, replace, repair, etc.) after it is utilized to predict the battery RUL and SOH for the purpose of battery PHM and BMS [16].

1.4. Research Objective

In this study, the objective is to progress the study of lithium-ion battery performance based on battery SOH and RUL prognostics. To do so, we propose an entropy-based Reinforcement Learning model, predict the next-cycle battery capacity, and compare the numerical results from the proposed entropy-based RL models to those from two other data-driven methods—namely, ARIMA and LSTM—which are both constructed based on the same input variable (i.e., permutation entropy of voltage sequences at each cycle). Permutation entropy of the battery discharge voltage, as well as the previous battery capacities,

are given to these models as input variables. Finally, evaluation metrics such as MSE, MAE, and RMSE are applied to the proposed methods to compare the observed and predicted battery capacities.

Based on Figure 1, the remainder of this work consists of the following sections. First, battery data is prepared and provided for the study. The data is then analyzed from different points of view. Based on the data analysis, various models are proposed for lithium-ion battery performance using ML and DL techniques. We evaluate and compare the models in detail in the next sections. Finally, conclusions are presented in the last section.
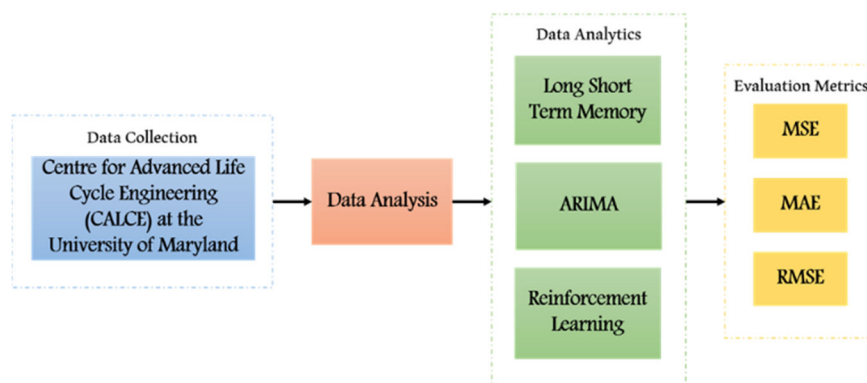

Figure 1. Prediction system for the lithium-ion batteries.

2. Related Work

In the current literature, entropy-based predictive models for battery prognostics, as well as other predictive models, have been researched and tested. Table 1 illustrates a brief overview of some of the most relevant and recently published papers that use data-driven methods for lithium-ion battery prognostics.

Table 1. An overview of different approaches to lithium-ion battery prognostics.

Ref.	Data	Methods	Results
[17]	NASA Ames Prognostics Center of Excellence (PCoE) database	Deep neural networks (DNN)	The proposed model successfully predicts the SOH and RUL of the lithium-ion battery but is less effective when real-time processing comes into play.
[18]	Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland	Deep neural networks (DNN)	The ANN predicts the battery State of Charge values with accuracy using only voltage, current, and charge/discharge time as inputs and achieves an MSE of 3.11×10^{-6}.
[19]	NASA Ames	Long short-term memory (LSTM)	The proposed model has a better performance for the time series problem of li-ion battery prognostics and a stronger learning ability of the degradation process when compared to other ANN algorithms.
[20]	NASA lithium-ion battery dataset	Long short-term memory (LSTM)	The method produces exceptional performances for RUL prediction under different loading and operating conditions.
[21]	Data repository of the NASA Ames Prognostics Center of Excellence (PCoE)	Autoregressive integrated moving average (ARIMA)	The RMSE of the model for the RUL prognostics varies in the range of 0.0026 to 0.1065.
[22]	Lithium-ion battery packs from forklifts in commercial operations	Autoregressive integrated moving average (ARIMA)	The ARIMA method can be used for SOH prognostics, but the loss function indicates further enhancement is needed for the environmental conditions.

[23]	NASA prognostic model library	Reinforcement Learning (RL)	RL model enables accurate calibration of the battery prognostics but has only been tested on simulated data and sim-to-real transfer needs to be made to test the proposed algorithm on real data.
[24]	SPMeT	Reinforcement Learning (RL)	The proposed method can extend the battery life effectively and ensure end-user convenience. However, experimental validation needs to be implemented for the optimal charging strategy.
[25]	Simulated datasets	Ensemble Learning	A data-driven method known as Ensemble Learning is presented for predicting degradation in a time-varying environment.
[26]	Experimental data from multiple lithium-ion battery cells at three different temperatures	Sparse Bayesian	The authors present a Sparse Bayesian model based on sample entropy of voltages for estimating SOH and RUL. It is shown that the Sparse Bayesian model outperforms the Polynomial model with the same input and target data.
[27]	Collected data through an experimental study	Unscented Particle Filter and Support Vector Regression	A hybrid model based on a combination of a data-driven method and a model-based approach is presented, which results in higher accuracy compared to each model individually.

The literature review reveals a research gap, which can be summarized as follows. Most of the research undertaken so far has relied on traditional Machine Learning and Deep Learning methods. However, the RL method is recognized as an area with room for exploration. Based on these findings, this paper is devoted to filling this gap in the research. LSTM and ARIMA methods are also studied as state-of-the-art models, which can be developed based on the entropy measures and compared with the RL method.

The main contribution of our study is the proposal of a Reinforcement Learning model based on the permutation entropy of the voltage sequences for predicting the next-cycle battery capacity. To the best of our knowledge, an RL model for lithium-ion battery prognostics, using entropy measures as the input, has not been previously tested in the literature. Additionally, we compare the numerical results from our proposed entropy-based RL model with the results from the state-of-the-art models (i.e., ARIMA and LSMT), which are built based on entropy measures for a fair and reliable comparison.

3. Data and Battery Specifications

The datasets used in this study were retrieved from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland [28]. The studied batteries are graphite/LiCoO2 pouch cells with a capacity rating of 1500 mAh, weight of 30.3 gm, and dimensions of 3.4 × 84.8 × 50.1 mm, labeled as PL19, PL11, and PL09. Table 2 shows the number of cycles in each dataset.

Table 2. Battery Cycles.

Batteries	# of Cycles
PL19	526
PL11	702
PL09	528

Figure 2 illustrates the battery capacities over the number of cycles and indicates the decrease in capacities as the number of cycles increases. It can also be observed that in PL09 and PL19 capacities are discrete, while in PL11, they differ continuously.
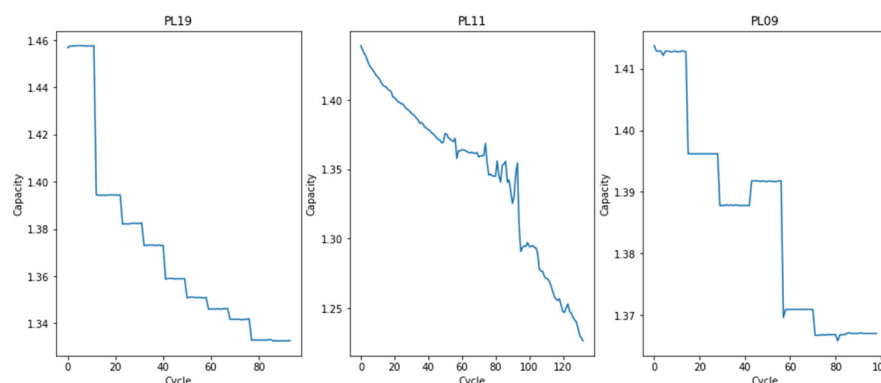
Figure 2. Capacity vs. Cycle for PL11, PL19, and PL09

Since the battery capacity and entropy were not observed in all cycles, we have estimated each unrecorded capacity value and its related entropy using the average of its previous and next known capacity and entropy value. By doing so, we have increased the number of data, and hence, the proposed models can be trained and tested more accurately.

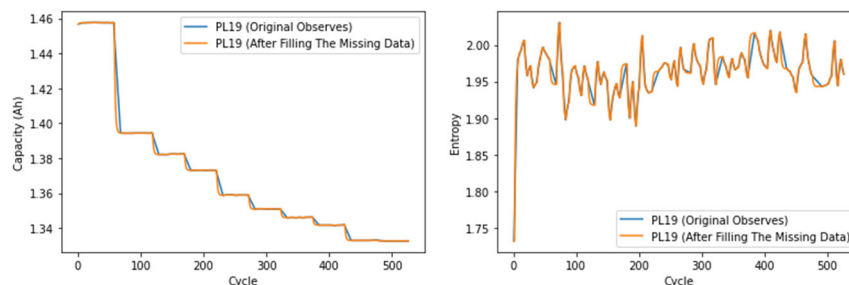Figures 3–5 indicate the resultant capacities and entropies after filling the missing data.

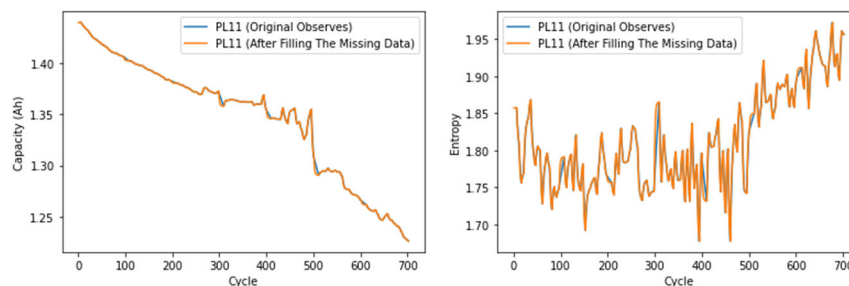**Figure 3.** Capacity vs. Cycle (left) and Entropy vs. Cycle (right) for PL19.

Figure 4. Capacity vs. Cycle (**left**) and Entropy vs. Cycle (**right**) for PL11.
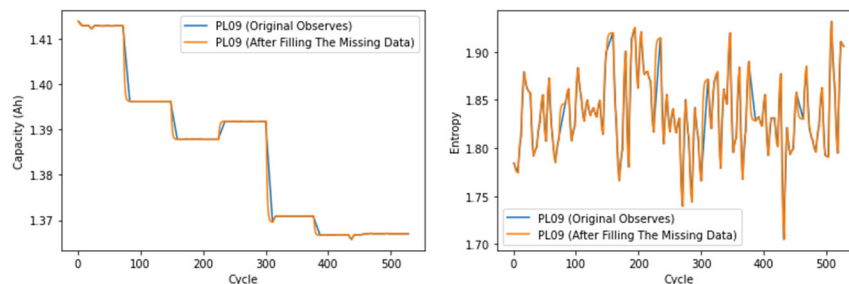

Figure 5. Capacity vs. Cycle (left) and Entropy vs. Cycle (right) for PL09.

4. Methodology

The mathematical notations used throughout this paper are summarized in Table 3.

Table 3. Glossary.

Indices	
n	Number of time series data
T	Number of times the permutation is found in time series data
Variables	
x_t	Input variable (permutation entropy of battery voltage) at step t
y_t	Observed battery capacity at step t
$\hat{y}_t$	Output variable (predicted battery capacity) at step t
h_t	Previous state at step t
c_t	Current state at step t
$\tilde{c}_t$	Intermediate cell state at step t
i_t	Input gate at step t
f_t	Forget gate at step t
o_t	Output gate at step t
p	Order of auto-regression
d	Order of difference
q	Order of moving average
s_t	State at step t
a_t	Action at step t
r_t	Reward at step t
R	Sum of the rewards
α	Learning rate
γ	Discount factor
$Q_{s_t.a_t}$	Q Table for states and actions at step t
Parameters	
PE	Permutation entropy
D	Order of permutation entropy
τ	Time delay in data series
V	Time series data matrix
l_i	Columns in V
π	Permutation pattern
P	Relative probability of each permutation
$W_i, W_f, W_o, W_c, U_i, U_f, U_o, U_c$	Weights in LSTM cells
b_i, b_f, b_o, b_c	bias vectors in LSTM cells
$\theta, \emptyset$	ARIMA coefficients
ε_t	Normal white noise with zero mean

In the following subsections, permutation entropy calculation and the proposed models will be discussed.

4.1. Permutation Entropy

To compute a D order permutation entropy for a one-dimensional set of time series data with n data points, the following steps are taken [29]. First, the data is partitioned into a matrix with D rows and $n - (D - 1)\tau$ columns, where τ is the delay time.

$$V = \begin{bmatrix} v(1) & v(1 + \tau) & \dots & v(1 + (D - 1)\tau) \\ v(2) & v(2 + \tau) & \dots & v(2 + (D - 1)\tau) \\ \vdots & \vdots & & \vdots \\ v(n) & v(n + \tau) & \dots & v(n + (D - 1)\tau) \end{bmatrix} \tag{1}$$

After rebuilding the data, π is defined as the permutation pattern for V columns:

$$\pi = \{l_0. l_1. \dots. l_{D-1}\} = \{0.1. \dots. D - 1\} \tag{2}$$

The relative probability of each permutation in π is calculated as below:

$$P(\pi) = \frac{T}{n - D + 1} \tag{3}$$

where T is the number of times the permutation is found in the time series. Finally, the relative probabilities are used to compute the permutation entropy:

$$PE = -\sum_{i=1}^{D!} P(\pi) \, log_2 \, P(\pi) \tag{4}$$

An algorithm for the permutation entropy computation is presented below.

Algorithm 1: Permutation Entropy

Step1 Reshape the data series into a matrix as in Equation (1)
Step2 Find the permutation patterns π
Step3 Calculate the probability of each permutation in π
Step4 Compute PE as in Equation (4)

Permutation entropy of the coarse-grained battery voltage is extracted, as in Figure 6. Despite the noise affecting the entropies, in PL11, the differences in the entropies are relatively small compared to the earlier cycles, while the deviations increase as the number of cycles increases. In PL19, the range of entropy is approximately constant over a different number of cycles; however, in PL09, they are completely random.
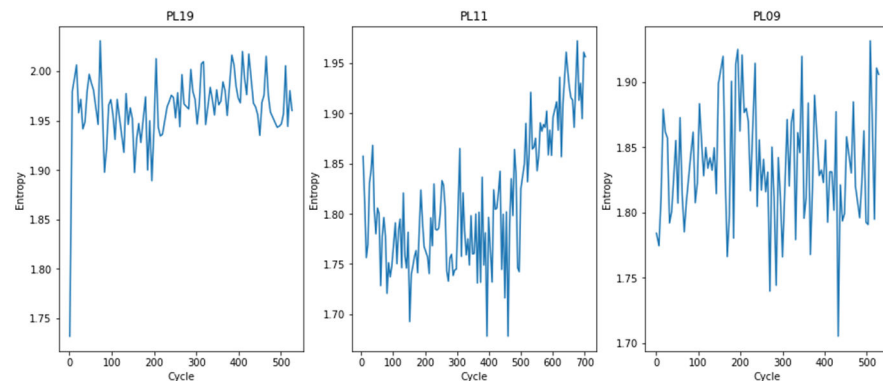

Figure 6. Entropy vs. Cycles for PL11, PL19, and PL09.

After data analysis, we split the data into train and test subsets. The proposed models utilize approximately 90% of the data for training purposes and take the rest for evaluation, as in Figure 7. The mechanism through which the training/test ration is selected is explained in the following sections.

Figure 7. Train–Test split schematic.

4.2. Predictive Models

The predictive models are presented in this section as follows.

4.2.1. LSTM

Long Short-Term Memory, known simply as LSTM, is a framework for a recurrent neural network (RNN) which avoids the problem of long-term dependency. Unlike standard feedforward neural networks, LSTM has feedback connections, and hence, it can update and store necessary information. It has been widely utilized in time series forecasting in different fields of science in recent years [30].

A unit LSTM cell consists of an input gate i_t, forget gate f_t, and an output gate o_t. Each gate receives the current input x_t, the previous state h_{t-1}, and the state c_{t-1} of the cell's internal memory. x_t, h_{t-1}, and c_{t-1} are passed through non-linear functions, which yield the updated c_t and h_t [31]. Considering W_i, W_f, W_o, W_c and U_i, U_f, U_o, U_c as the correspondig weights matrices and b_i, b_f, b_o, b_c as the bias vectors, each LSTM cell operates based on the following Equations.

$$i_t = \sigma(x_t U_i + h_{t-1} W_i + b_i) \tag{5}$$

$$\tilde{c}_t = tanh(x_t U_c + h_{t-1} W_c + b_c) \tag{6}$$

$$f_t = \sigma(x_t U_f + h_{t-1} W_f + b_f) \tag{7}$$

$$c_t = f_t * C_{t-1} + i_t * \tilde{c}_t \tag{8}$$

$$o_t = \sigma(x_t U_o + h_{t-1} W_o + b_o) \tag{9}$$

$$h_t = tanh(c_t) * o_t \tag{10}$$

In this study, all three gates take permutation entropy of the battery voltage at cycle t and the battery capacity at cycle $t-1$ as their input variables, x_t and c_{t-1}, and output the estimated battery capacity, $\hat{y}$, for the given inputs as shown in Figure 8. Furthermore, an algorithm is presented for the proposed LSTM model.
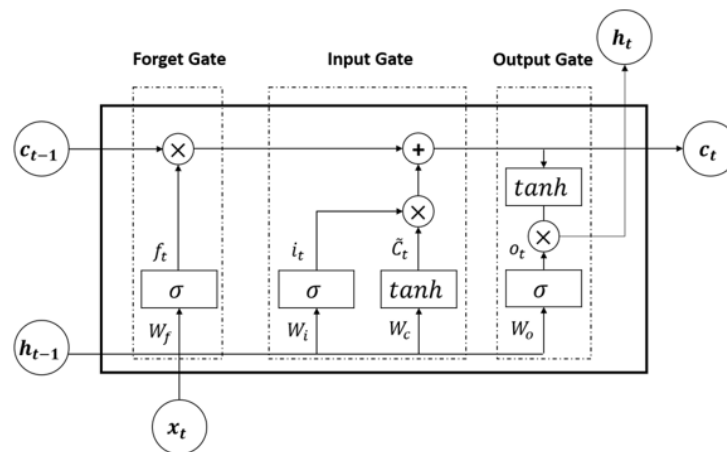

Figure 8. Schematic of a unit LSTM cell.

Algorithm 2: LSTM

Input: $x = \{PE_1. PE_2. \dots . PE_n\}$: Permutation Entropy of Battery Voltage and c_{t-1};
Output: $\hat{y} = \{Capacity_1. Capacity_2. \dots . Capacity_n\}$: Battery Capacity;
for t in range(epoch) do
Step1 Calculate i_t
Step2 Determine $\tilde{c}_t$
Step3 Calculate f_t
Step4 Update c_t
Step5 Calculate o_t
Step6 Update h_t
Step7 Determine the output $\hat{y} = LSTM_{forward}(x)$
Step8 Compute the loss function as Equations (20)–(22)
end

4.2.2. ARIMA

The Autoregressive Integrated Moving Average (ARIMA) method is proposed as a technique for statistical analysis in time series data. An ARIMA model is a combination of the autoregressive (AR) and moving average (MA) models. The ARIMA model can be explained according to three notations—p, d, and q—which define the type of the ARIMA model:

- p: order of auto-regression
- d: order of difference
- q: order of moving average

For AR(p), we have:

$$\hat{y}_t = \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \varepsilon_t \tag{11}$$

MA(q) can be described as follows:

$$\hat{y}_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{12}$$

ARMA($p.q$) is a combination of AR(p) and MA(q), and is described as below:

$$\hat{y}_t = \emptyset_1 y_{t-1} + \cdots + \emptyset_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q} \tag{13}$$

where y_t and $\hat{y}_t$, respectively, are the observed and estimated values; $\emptyset$ and θ, respectively, are coefficients; and ε_t is a normal white noise process with zero mean.

ARIMA is an advanced version of ARMA, which also works well for non-stationary time series data. To convert the non-stationary to stationary data, a data transformation is needed using a d-order difference equation [32]. Consequently, ARIMA $(p.d.q)$ can be described as Equation (14).

$$\hat{w}_t = \emptyset_1 w_{t-1} + \cdots + \emptyset_p w_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} \dots - \theta_q \varepsilon_{t-q} \tag{14}$$

where $w_t = \nabla^d y_t$ and ∇ is the gradient operator. When $d = 0$, Equation (14) is the same as Equation (13) and, thus, ARIMA acts the same as ARMA. p and q are initialized using the autocorrelation function (ACF) and partial autocorrelation function (PAFC).

AFC measures the average correlation between data points in a time series and previous values of the series measured for different lag lengths. PACF is the same as ACF, except that each correlation controls for any correlation between observations of a shorter lag length [32].

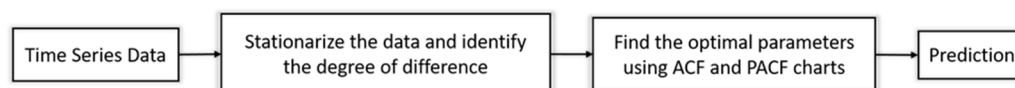Figure 9 demonstrates the ARIMA framework from the input data stage through the prediction stage.

Figure 9. ARIMA framework.

In this study, an ARIMA model is proposed to predict future battery capacities. Since we are working with a non-stationary time series, we have made a data transformation with $d = 1$. p and q, respectively, are set to 5 and 0, and thus, predictions were made with ARIMA(5.1.0). The rationale behind choosing the order of the ARIMA model is as follows. We compare the results from a range of non-negative integers, $p = [1,10]$ (extracted from the existing literature), and select the optimal number of time lags for the autoregressive model, which results in minimal errors compared to other orders in that range. The results from the optimal model are displayed and reported here.

There is a battery voltage sequence at each cycle (i.e., a time series of voltages at each cycle). We first compute the permutation entropy of each voltage sequence according to the corresponding algorithm; then, we use the time series of the permutation entropy measures (i.e., one entropy measure at each cycle) as an input in the ARIMA model, compare them with the deviations in the battery capacities, and predict the next-cycle battery capacity as an output of the model.

An algorithm for the ARIMA model is presented as follows.

Algorithm 3: ARIMA

Input: $x = \{PE_1. PE_2. \dots. PE_n\}$: Permutation Entropy of Battery Voltage Sequences at each Cycle;

Output: $\hat{y} = \{Capacity_1. Capacity_2. \dots. Capacity_n\}$: Battery Capacity;

- Make time series data stationary with appropriate d;
- Initialize p and q using ACP and PACF;
- Fit ARIMA($p. d. q$) to data;
- Predict the next-cycle capacity as Equation (14);
- Calculate the loss function using Equations (20)–(22).

4.2.3. Reinforcement Learning

Reinforcement Learning (RL) is a type of multi-layered neural network, and has become a focus of research in modern artificial intelligence. The concept is based on rewarding or punishing an agent's performance in a specific environment. A state is a description of the environment made to provide the necessary information for the agent to decide at each time step. For each and every state s, the agent has a number of selecting actions a to make decisions from. A policy is required, based on a cost function, to map each state to the optimal action with the consideration of maximizing its reward function during the episode [33].

Reinforcement Learning has real-life applications in various fields such as driving cars, landing rockets, trading and finance, diagnosing patients, and so on. This Deep Learning technique differs from supervised learning, as it does not require correct sets of actions and labeled input/output pairs [34]. Instead, the goal is to find a balance between exploration and exploitation. Figure 10 illustrates the schematic of a general Reinforcement Learning structure and its Equations are described as follows.
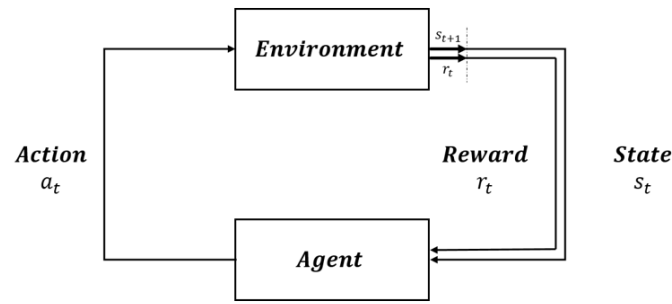
Figure 10. Reinforcement Learning Schematic.

$$a_t \sim \pi(a_t|s_t) \tag{15}$$

$$s_{t+1} \sim f_{state}(s_{t+1}|s_t.a_t) \tag{16}$$

$$r_{t+1} = f_{reward}(s_t.a_t.s_{t+1}) \tag{17}$$

$$R = \sum_{t=0}^{\infty} \gamma^t r_{t+1} \tag{18}$$

$$Q_{s_t.a_t}^{new} = Q_{s.a_t}^{old} + \alpha \left(\overbrace{r_t + \gamma max Q_{s_{t+1}.a}^{next\,s-a}}^{Target} - \overbrace{Q_{s_t.a_t}^Q}^{Prediction} \right) \tag{19}$$

In this study, we have considered the permutation entropy of the battery voltage as the states and the capacities as the actions, which should be taken at each state based on the given entropy. An algorithm for the RL model is presented in the following.

Algorithm 4: Reinforcement Learning

States: $s = \{PE_1.PE_2.....PE_n\}$: Permutation Entropy of Battery Voltage;
Actions: $a = \{Capacity_1.Capacity_2.....Capacity_n\}$: Battery Capacity;
Define the optimal policy;
Initialize the parameters α and γ;
for t in range (epoch) do
Calculate a_t using the optimal policy
Determine s_{t+1} as a function of the state and the previous state and action
Compute r_{t+1} and R
Update $Q_{s_t.a_t}$ using Equation (19)
Evaluate the estimation using the following loss function as in Equations (20)–(22)
end

The hyperparameters of the proposed models define how they are structured. Optimal hyperparameters are approximated so that the loss is reduced. In other words, we explore various model architectures and search for the optimal values in the hyperparameter space to minimize the resulting performance metrics; for instance, Mean Squared Error. For this purpose, in the three models, grid search is used for tuning the hyperparameters and achieving reliable comparisons between the numerical results from the models. A model is built for each possible combination of all of the hyperparameter values; next, the models are evaluated based on the performance metrics, and then the architecture which produces the best results is selected. The results and findings are reported in the following section.

5. Results and Findings

The numerical results and findings are presented in this section as follows.

5.1. Performance Measures

To evaluate the performance of the proposed models, we present the observed and predicted battery capacities for ARIMA and LSTM models and the reward and loss functions obtained from the RL model. Furthermore, we compare the observed and predicted battery capacities gained from each of these models using three performance metrics [35] as shown below:

Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{20}$$

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{21}$$

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{22}$$

where y_t and $\hat{y}_t$, respectively, are the observed and predicted capacity at cycle t, and n is the number of test data.

5.2. Numerical Results

The observed and predicted battery capacities results from ARIMA and LSTM models are shown in Figures 11–13. Based on the graphs obtained, it can be seen that in all three datasets the ARIMA model predictions are following the trends in the test data, and so, yields better results as compared to the LSTM model for predicting the time series of battery capacities.
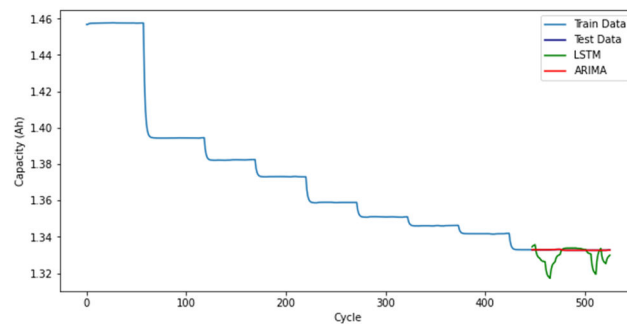

Figure 11. Train, test, and predicted data results from ARIMA and LSTM models for PL19.
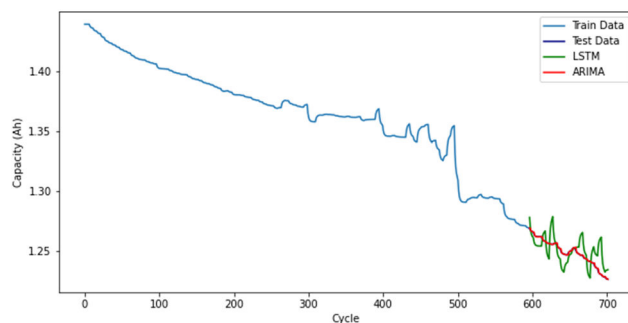
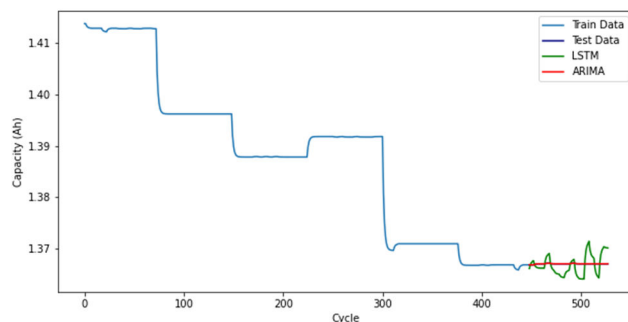**Figure 12.** Train, test, and predicted data results from ARIMA and LSTM models for PL11.

Figure 13. Train, test, and predicted data results from ARIMA and LSTM models for PL09.

The early battery-life prediction, which includes a prediction of the battery cycles at earlier cycles, is performed, and the results are displayed in Figures 14–16. It is observed that the deviation between the predicted capacities and the actual capacities are not significant, indicating that the proposed ARIMA and LSTM models are capable of predicting battery capacities at earlier cycles.
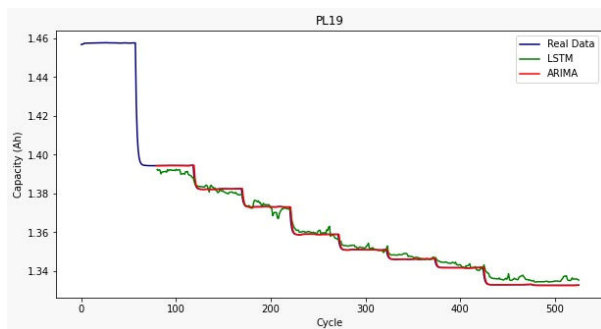

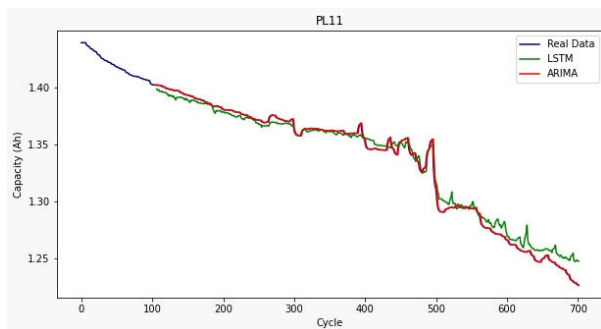**Figure 14.** Train, test, and predicted data results from ARIMA and LSTM models for PL19.

Figure 15. Train, test, and predicted data results from ARIMA and LSTM models for PL11.
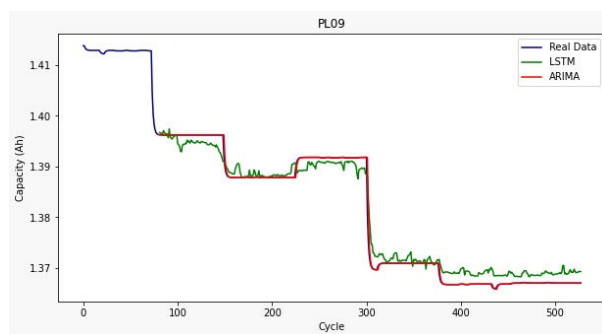
Figure 16. Train, test, and predicted data results from ARIMA and LSTM models for PL09.

In the RL model, as demonstrated in Figure 17, the reward values have an impressive increase and immediately become stable with some noise. The loss values increase at first; however, after approximately 250 epochs, they decline to 0, which verifies the procedure of Reinforcement Learning.
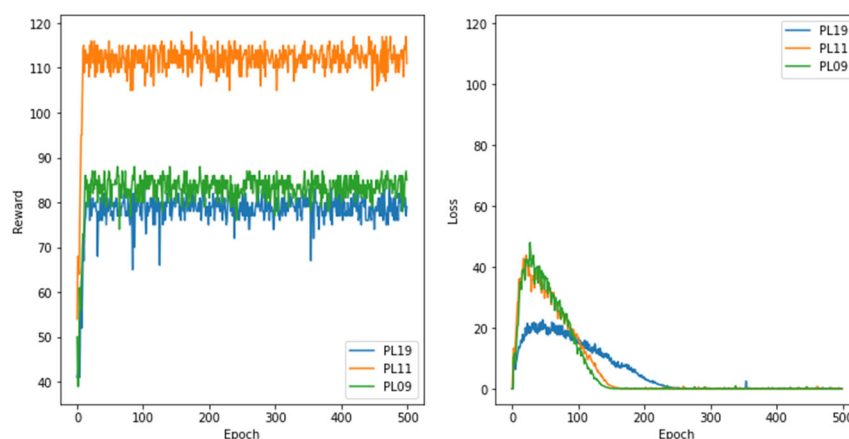

Figure 17. Reward and Loss Function (RL model).

To find the best data split ratio, our proposed RL approach is initially trained using shuffled datasets with five different training ratios (70%, 75%, 80%, 85%, and 90%). Afterwards, Mean Squared Error (MSE) is utilized as a loss function to evaluate the obtained results. Based on Table 4, the best accuracy is gained by using 90% of each dataset for training purposes and using the rest for the testing process (Figure 18). Finally, this ratio is applied to training the other two models (LSTM and ARIMA). To save space, the results from the LSTM and ARIMA models are not reported here. The results from the other two models are consistent with those from RL (i.e., the best training ratio of 10%).

Table 4. MSE Value for Different Training Ratios for the RL model.

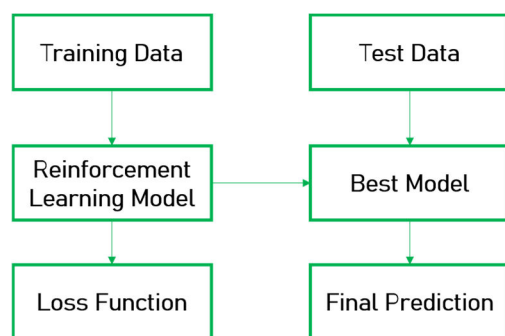| Battery	Training Ratio				
	70%	75%	80%	85%	90%
PL19	0.0422	0.0618	0.0179	0.0008	0.0002
PL11	0.0718	0.0465	0.0153	0.0156	0.0084
PL09	0.0209	0.0007	0.0006	0.0003	0.0003

Figure 18. Finding the best Train–Test Split.

5.3. Comparisons

Tables 5–7 represent a snapshot comparison of the aforesaid models for the PL19, PL11, and PL09 datasets, respectively. As the results show, in all datasets, ARIMA slightly surpasses the LSTM and RL models since it results in the smallest MSE, MAE, and RMSE values. However, the differences are not significant, and for PL19 and PL11, ARIMA and RL yield approximately the same values of performance measures. It is concluded that LSTM and RL also result in minor errors.

Table 5. MSE, MAE, and RMSE values for the predictive models (PL19).

Evaluation Metric	LSTM	ARIMA	RL
MSE	0.00003	0.00001	0.0002
MAE	0.00417	0.00001	0.00005
RMSE	0.00580	0.00003	0.00009

Table 6. MSE, MAE, and RMSE values for the predictive models (PL11).

Evaluation Metric	LSTM	ARIMA	RL
MSE	0.00011	0.00001	0.0084
MAE	0.00012	0.00026	0.00054
RMSE	0.01095	0.00066	0.00090

Table 7. MSE, MAE, and RMSE values for the predictive models (PL09).

Evaluation Metric	LSTM	ARIMA	RL
MSE	0.00001	0.00001	0.0003
MAE	0.00171	0.00001	0.03997
RMSE	0.00200	0.00002	0.05751

From Tables 5–7, it is observed that the ARIMA model yields smaller errors compared to the LSTM model. ARIMA, which is a mean-reverting process, has the ability to predict battery capacities with smaller deviations. However, the LSTM model—which is a recurrent network—attempts to avoid the long-term dependency by storing only necessary information, and thus, it is unable to probabilistically exclude the input (i.e., previous permutation entropy of battery voltage sequences) and the recurrent connections to the units of the network from the activation and weight updates while the model is being trained. Consequently, the deviations between the actual battery capacities and the predicted capacities resulting from the LSTM model are greater than those resulting from the ARIMA model. The results displayed on Figures 11–13 are consistent with the Tables.

6. Conclusions

In lithium-ion battery applications, failures in the system can be minimized by performing prognostics and health management. Data-driven methods are one way of doing so, and identify the optimal replacement intervals or the optimal time for changing the battery in an appropriate manner. This paper presents three different models (LSTM, ARIMA, and RL), which all are built based on the permutation entropies of the battery voltage sequences, for next-cycle battery capacity prediction using the status of the previous states. In various data conditions, different models may be required; having a collection of models, even for the same purpose, can be useful. In addition to accurate prediction of battery capacities based on the ARIMA model, it is shown that the LSTM and the proposed entropy-based RL models have similar performance and both result in small errors.

References

1. Shimamura, O.; Abe, T.; Watanabe, K.; Ohsawa, Y.; Horie, H. Research and development work on lithium-ion batteries for environmental vehicles. *World Electr. Veh. J.* **2007**, *1*, 251–257.
2. Jaguemont, J.; Boulon, L.; Dubé, Y. A comprehensive review of lithium-ion batteries used in hybrid and electric vehicles at cold temperatures. *Appl. Energy* **2016**, *164*, 99–114.
3. Han, X.; Lu, L.; Zheng, Y.; Feng, X.; Li, Z.; Li, J.; Ouyang, M. A review on the key issues of the lithium ion battery degradation among the whole life cycle. *eTransportation* **2019**, *1*, 100005.
4. Wu, L.; Fu, X.; Guan, Y. Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies. *Appl. Sci.* **2016**, *6*, 166.
5. Williard, N.; He, W.; Hendricks, C.; Pecht, M. Lessons learned from the 787 dreamliner issue on lithium-ion battery reliability. *Energies* **2013**, *6*, 4682–4695.
6. Ge, M.F.; Liu, Y.; Jiang, X.; Liu, J. A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. *Measurement* **2021**, *174*, 109057.
7. Li, X.; Zhang, L.; Wang, Z.; Dong, P. Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks. *J. Energy Storage* **2019**, *21*, 510–518.
8. Li, P.; Zhang, Z.; Xiong, Q.; Ding, B.; Hou, J.; Luo, D.; Rong, Y.; Li, S. State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network. *J. Power Sources* **2020**, *459*, 228069.
9. Tran, M.K.; Panchal, S.; Khang, T.D.; Panchal, K.; Fraser, R.; Fowler, M. Concept review of a cloud-based smart battery management system for lithium-ion batteries: Feasibility, logistics, and functionality. *Batteries* **2022**, *8*, 19.
10. Atkins, P. *The Laws of Thermodynamics: A Very Short Introduction*. OUP Oxford University Press: Oxford, UK, 2010.
11. Namdari, A.; Li, Z.S. A Multiscale Entropy-Based Long Short Term Memory Model for Lithium-Ion Battery Prognostics. In Proceedings of the 2021 IEEE International Conference on Prognostics and Health Management (ICPHM), Detroit, MI, USA, 7–9 June 2021; pp. 1–6.
12. Bandt, C.; Pompe, B. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.* **2002**, *88*, 174102.
13. Awad, M.; Khanna, R. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Springer Nature: Berlin/Heidelberg, Germany, 2015.
14. Singh, B.; Desai, R.; Ashar, H.; Tank, P.; Katre, N. A Trade-off between ML and DL Techniques in Natural Language Processing. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021; Volume 1831, p. 0120025.
15. Zhang, Y.; Xiong, R.; He, H.; Pecht, M.G. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Trans. Veh. Technol.* **2018**, *67*, 5695–5705.

16. Li, W.; Cui, H.; Nemeth, T.; Jansen, J.; Uenluebayir, C.; Wei, Z.; Zhang, L.; Wang, Z.; Ruan, J.; Dai, H.; et al. Deep reinforcement learning-based energy management of hybrid battery systems in electric vehicles. *J. Energy Storage* **2021**, *36*, 102355.

17. Khumprom, P.; Yodo, N. A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm. *Energies* **2019**, *12*, 660.

18. Almeida, G.; Souza, A.C.; Ribeiro, P.F. A Neural Network Application for a Lithium-Ion Battery Pack State-of-Charge Estimator with Enhanced Accuracy. In Multidisciplinary Digital Publishing Institute Proceedings. *Proceedings* **2020**, *58*, 33. https://doi.org/10.3390/WEF-06915.

19. Long, B.; Li, X.; Gao, X.; Liu, Z. Prognostics comparison of lithium-ion battery based on the shallow and deep neural networks model. *Energies* **2019**, *12*, 3271.

20. Hinchi, A.Z.; Tkiouat, M. A deep long-short-term-memory neural network for lithium-ion battery prognostics. In Proceedings of the International conference on industrial engineering and operations management, Paris, France, 26–27 July 2018; pp. 2162–2168.

21. Chen, L.; Xu, L.; Zhou, Y. Novel approach for lithium-ion battery on-line remaining useful life prediction based on permutation entropy. *Energies* **2018**, *11*, 820.

22. Huotari, M.; Arora, S.; Malhi, A.; Främling, K. A Dynamic Battery State-of-Health Forecasting Model for Electric Trucks: Li-Ion Batteries Case-Study. In Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Portland, OR, USA, 16–19 November 2020; Volume 84560, p. V008T08A021.

23. Unagar, A.; Tian, Y.; Chao, M.A.; Fink, O. Learning to Calibrate Battery Models in Real-Time with Deep Reinforcement Learning. *Energies* **2021**, *14*, 1361.

24. Kim, M.; Baek, J.; Han, S. Optimal Charging Method for Effective Li-ion Battery Life Extension Based on Reinforcement Learning. *arXiv* **2020**, arXiv:2005.08770.

25. Wang, L.; Lu, D.; Wang, X.; Pan, R.; Wang, Z. Ensemble learning for predicting degradation under time-varying environment. *Qual. Reliab. Eng. Int.* **2020**, *36*, 1205–1223.

26. Hu, X.; Jiang, J.; Cao, D.; Egardt, B. Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling. *IEEE Trans. Ind. Electron.* **2015**, *63*, 2645–2656.

27. Peng, X.; Zhang, C.; Yu, Y.; Zhou, Y. Battery remaining useful life prediction algorithm based on support vector regression and unscented particle filter. In Proceedings of the 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Ottawa, ON, Canada, 20–22 June 2016; pp. 1–6.

28. He, W.; Williard, N.; Osterman, M.; Pecht, M. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. *J. Power Sources* **2011**, *196*, 10314–10321.

29. Namdari, A.; Li, Z. A review of entropy measures for uncertainty quantification of stochastic processes. *Adv. Mech. Eng.* **2019**, *11*, 1687814019857350.

30. Sak, H.; Senior, A.; Beaufays, F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv* **2014**, arXiv:1402.1128.

31. Elsaraiti, M.; Merabet, A. Application of Long-Short-Term-Memory Recurrent Neural Networks to Forecast Wind Speed. *Appl. Sci.* **2021**, *11*, 2387.

32. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.

33. Abedi, S.; Yoon, S.W.; Kwon, S. Battery energy storage control using a reinforcement learning approach with cyclic time-dependent Markov process. *Int. J. Electr. Power Energy Syst.* **2022**, *134*, 107368.

34. Haney, B. Reinforcement Learning Patents: A Transatlantic Review. In *Transatlantic Technology Law Forum*; Working Paper Series; Stanford Law School: Stanford, CA, USA, 2020.

35. Namdari, A.; Li, Z.S. An Entropy-based Approach for Modeling Lithium-Ion Battery Capacity Fade. In Proceedings of the 2020 Annual Reliability and Maintainability Symposium (RAMS), Palm Springs, CA, USA, 27–30 January 2020; pp. 1–7.