processes

Article

Intelligent Model for Power Cells State of Charge Forecasting in EV

Víctor Caínzos López [1,*,†], Esteban Jove [1,†], Francisco Zayas Gato [1,†], Francisco Pinto-Santos [2,†], Andrés José Piñón-Pazos [1,†], Jose-Luis Casteleiro-Roca [1,†], Hector Quintian [1,†] and Jose Luis Calvo-Rolle [1,†]

[1] Department of Industrial Engineering, University of A Coruña, 15405 A Coruña, Spain; esteban.jove@udc.es (E.J.); f.zayas.gato@udc.es (F.Z.G.); andres.pinon@udc.es (A.J.P.-P.); jose.luis.casteleiro@udc.es (J.-L.C.-R.); hector.quintian@udc.es (H.Q.); jlcalvo@udc.es (J.L.C.-R.)
[2] Computer Science Faculty, Pontifical University of Salamanca, C/ Compañía, 5, 37002 Salamanca, Spain; fpintosa@upsa.es
* Correspondence: victor.cainzos.lopez@udc.es
† These authors contributed equally to this work.

Abstract: In electric vehicles and mobile electronic devices, batteries are one of the most critical components. They work by using electrochemical reactions that have been thoroughly investigated to identify their behavior and characteristics at each operating point. One of the fascinating aspects of batteries is their complicated behavior. The type of power cell reviewed in this study is a Lithium Iron Phosphate $LiFePO_4$ (LFP). The goal of this study is to develop an intelligent model that can forecast the power cell State of Charge (SOC). The dataset used to create the model comprises all the operating points measured from an actual system during a capacity confirmation test. Regression approaches based on Deep Learning (DL), such as Long Short-Term Memory networks (LSTM), were evaluated under different model configurations and forecasting horizons.

Keywords: LSTM; forecasting; battery

check for updates

Citation: Caínzos López, V; Jove, E.; Zayas Gato, F.; Pinto-Santos, F.; Piñón-Pazos, A.J.; Casteleiro-Roca, J.L.; Quintián, H.; Calvo-Rolle, J.L. Intelligent Model for Power Cells State of Charge Forecasting in EV. *Processes* **2022**, *10*, 1406. https://doi.org/10.3390/pr10071406

Academic Editor: Zhiwei Gao

Received: 19 May 2022
Accepted: 16 July 2022
Published: 19 July 2022

1. Introduction

The most often utilized solution to overcome the problem of intermittent energy generation in renewable energy systems is an electrical energy storage system [1]. Electric energy storage is utilized in renewable energy systems as a technique of storing energy when production exceeds demand and returning energy when demand exceeds production. These systems are also employed to power portable electronics applications or for substituting fossil fuels in automobiles, such as electric cars [1,2].

A growing amount of research is being conducted to enhance electrical energy storage devices. Some research is focusing on coping with intermittent energy generation in renewable energy systems such as solar and wind power systems, intending to use those technologies in Smart Grid systems [3]. Furthermore, the growing usage of portable devices such as tablets or smartphones, which increases the quality of life, needs greater autonomy, shorter charging times, and lighter energy storage components [4].

The driving range of an electric vehicle is now a significant issue tied to the storage system. Although inefficient, internal combustion engines give an excellent driving range due to the high energy density of fossil fuels [5].

Li-ion power cells are excellent prospects for use in electromobility and the renewable energy industry. This battery is suited for electric drive cars due to its high power density and cell voltage, minimal self-discharge, no memory effect, and extended life cycle [6]. Moreover, Li-ion batteries are available in various chemistry, each with its own set of characteristics, such as energy or power cells, to meet the various needs of the automobile sector [7].

Because of the widespread usage of these batteries in portable devices such as electric vehicles and the necessity for safety, failure analysis and modeling of these batteries are critical [8].

On the other hand, among the great number of machine learning applications [9–11], time series analysis can be used for clustering [12,13], classification [14], query by content [15], anomaly detection, as well as forecasting [16,17], which is the branch of the current study. Moreover, given the increasing availability of data and computing power in recent years, deep learning has become a critical component of the new generation of time series forecasting models. Overcoming the traditional machine learning disadvantages, being more robust to missing values, recognizing complex patterns in the data, and working well in long-term forecast [18–21].

Recurrent Neural Networks (RNN) are the most classical and used architecture for time series forecasting problems with variable-length sequences of inputs [22–24], such as unsegmented, connected handwriting recognition [25] or speech recognition [26,27]. LSTM [28] and Gated Recurrent Units (GRU) [29] are evolutions that were developed to deal with the vanishing gradient problem that involves traditional RNNs training.

This research aims to create an intelligent model that can forecast the SOC of a power cell. The aforementioned regression approaches, based on LSTM networks, were employed to take advantage of its capacity to predict long sequences and to test it on multiple study scenarios. The approach is chosen using the minimum mean squared error criterion.

The document is organized as follows. First, an introduction is provided. Then, the description of the system under study is presented, describing the test carried out. Next, the methods used in this study are detailed. After that, the model performance and scores are displayed in the results section, providing a statistical analysis of such results. Finally, future efforts and conclusions are shown.

2. Description of the System under Study

The SOC model of an LFP power cell utilized in electric car battery systems is presented in this study. The tests performed and the power cell are detailed in further depth in the subsections.

2.1. The Battery

Batteries are one alternative way of storing electric energy. These devices store energy in an electrochemical medium and are capable of storing and then providing electrical energy as a result of electrochemical processes [1].

Li-ion batteries are usually built with five distinctive layers: the negative collector (for negative current), a negative electrode (that is called "anode"), a separator (like a membrane), a positive electrode (called "cathode"), and the positive current collector. There are different types of positive electrode materials [30]: (1) a metal oxide with layered structure, like lithium cobalt oxide ($LiCoO_2$/LCO) [31]; (2) a metal oxide with a 3-dimensional spinel structure, like lithium manganese oxide ($LiMn_2O_4$) [32]; (3) lithium nickel manganese cobalt oxide ($LiNiMnCoO_2$/NMC); and (4) a metal phosphate with an olivine structure, such as lithium iron phosphate $LiFePO_4$ (LFP) [33]. The anode is generally made of graphite or metal oxide. Among carbonaceous materials, carbon nanotubes (CNTs) are considered the most promising materials being developed [34]. The electrolyte can be liquid, polymer or solid [35]. There are various types of lithium-ion batteries available such as cylindrical, coin, and prismatic. Cylindrical and coin batteries are used in small products such as wrist watches, laser pointers, and slide changers [36], and prismatic batteries are used for high-capacity applications such as automobiles [37].

The internal reactions in a lithium-ion battery are described in [38], and the specific reactions of $LiFePO_4$ battery cell are explained in [39]. Basically, during charge cycle, the electrons flow from cathode to anode using a power source to force this flow, and inside the battery, the reactions take place to increase the SOC. During discharge, the electrons flow in the opposite way in the electric circuit; the current flow in electric diagrams is the opposite of the real flow of electrons, from the positive electrode to the negative one. As it is described in [40], the membrane separator is a key component in a liquid-electrolyte battery for electrically separating the cathode and the anode, ensuring ionic transport between them. Besides these basic requirements, endowing the separator with specific beneficial functions is now being paid great attention to because it provides an important alternative approach for the development of batteries, particularly next-generation high-energy rechargeable batteries.

One reason that decreases the capacity of batteries is the temperature, among other phenomena. For example, the battery degradation in electric vehicles can be optimized by reducing the power used to charge it; with a low power charger, the battery does not increase its temperature [41]. One of the problems in high-power chargers is that the battery increases its temperature and it reduces its capacity [42]. In [43], a study of the dependency between the battery's performance and the temperature is done; in [44], a comprehensive mathematical model is used to analyze the degradation of the battery. In both papers, it is possible to conclude that the temperature is one of the reasons that reduces the battery's capacity. Also, in [45] an analysis of the depth of discharge (DOD) is performed (and other analyses of the degradation among the whole life cycle of Li-ion batteries), and one of the conclusions about the DOD in electric vehicles is that they should work in range from 30% to 80% to prevent degradation.

Lithium-Ion batteries are the most often used batteries in portable gadgets. They are often seen in electronic products and, more lately, electric automobiles [46]. Li-ion cells are distinguished by their high energy density, owing to the high voltage they can be produced compared to other battery types while weighing less [1]. They have a long life cycle, a low self-discharge rate, and no memory effect [46,47].

2.2. Capacity Confirmation Test

The capacity confirmation test, done on the battery, measures the capacity of the battery in ampere-hour [48]. The measurement of the battery capacity is done at a constant current. The test begins with the battery fully charged, so the SOC is at its highest. Then, the battery produces a steady current until it achieves the discharge voltage set by the manufacturer [48]. Then, the battery is put to rest until the voltage is restored, and the charging process may continue. Once the voltage has returned to a certain level, the battery is charged to its maximum voltage, as set by the manufacturer [48]. These operations are carried out with a continuous current, and the current SOC is determined at each moment.

The capacity confirmation test was performed using a battery tester. This gadget can charge and discharge the cell within the test settings, in this case employing a constant current and taking the necessary measurements. The battery voltage, the current provided or absorbed by the battery, the temperature at two different times, and the length of the test are all measured.

A current source i(t) is the testing equipment that delivers or absorbs the current.

The power cell employed in this study carbon/$LiFePO_4$ (material of the anode/cathode) cell. It has a nominal capacity of 8000 mAh and a typical voltage of 3.3 V; its datasheet [49] mention also the maximum charging voltage (3.65 V), the cut-off discharge voltage (2.0 V), and maximum charge and discharge current, 32 A and 200 A respectively. This battery has a cylindrical shape with 38 mm diameter and a height of 123 mm.

2.3. Dataset

As previously stated, the whole dataset utilized in this study was gathered via testing a power cell during its capacity confirmation test. Table 1 displays the total samples collected with the sample time and the scope of values for each magnitude. Different devices such as an ammeter, voltmeter, and temperature sensors were engaged in measuring the intensity (A), voltage (V), and temperature (T_1, T_2) variables, respectively.

Table 1. Measured variables and range of values.

		Variables Values			
Samples	**Sample-Time (s)**	**Amps (I)**	**Volts (V)**	**T_1 (°C)**	**T_2 (°C)**
16,370	1	[−32.04, 32.05]	[2, 3.65]	[37.43, 41.80]	[36.15, 40.56]

While the test is ongoing, four different stages can be distinguished: Charge: by the time the initial voltage is 3 V and it is increased up to 3.65 V. Rest after the charging: without current flowing, the battery voltage reaches its nominal value, 3.3 V. Discharge: the voltage is decreased from 3.3 V to 2 V. And rest after the discharging: having no current flowing, the voltage of the battery goes from 2 V to 3 V, thus letting a new cycle be ready to begin anew.

Hence, the SOC of the battery is calculated at each stage using current and time. While temperature sensors record measurements in two different points (T_1, T_2) in order to identify a battery failure during the test. Thereby, a problem can be discovered if readings are significantly outside of a usual range. Besides, their values and the voltage change cyclically depending on whether the battery is charging, discharging or resting. Thus, all the data collected during the capacity confirmation test was labeled to determine the SOC. The measured data was recorded with a sampling frequency of 1 Hertz.

3. Methods

In this section, the technical procedures used across the study are exposed, such as the feature engineering to preprocess collected data, the window sliding technique to build up a suitable dataset of sequences, and the LSTM units.

3.1. Feature Engineering

Data has not been randomly shuffled before splitting but after. This ensures that chopping the data into windows of consecutive samples is still possible, and the test results are more realistic, being evaluated on the data collected after the model was trained. A proportion of 80–20% for training and test set, respectively, has been selected (see Figure 1).

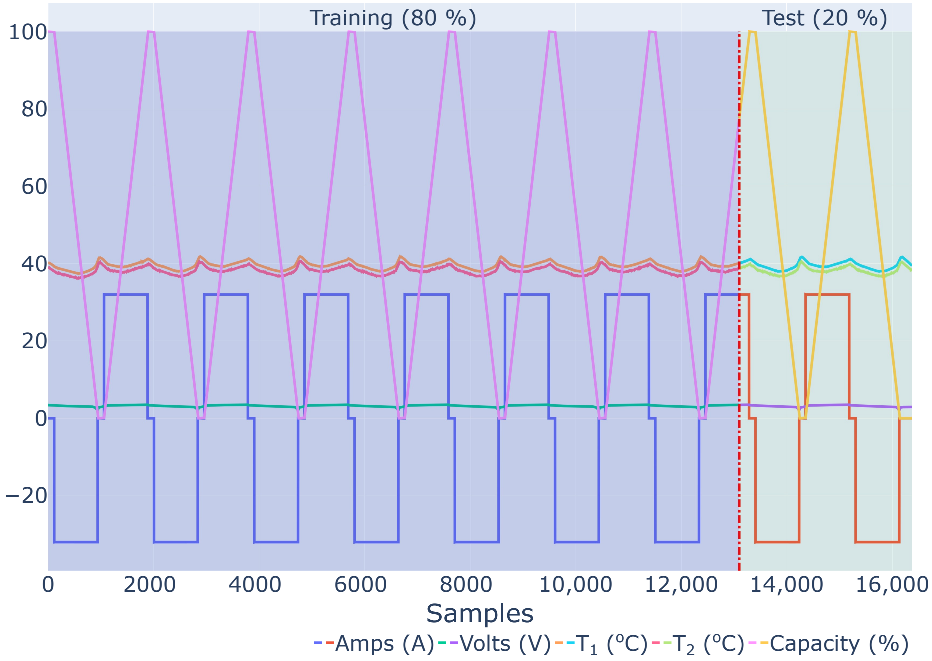

Figure 1. Traces and unit ranges for metrics measured during the capacity test: Intensity (Amps), Voltage (Volts), T_1 (°C), T_2 (°C) and Capacity (%).

Normalization is a common way of scaling features: subtract the mean and divide by the standard deviation of each feature.

$$\overline{x_i} = \frac{x_i - \mu_x}{\sigma_x} \quad \forall i, \ldots, n \tag{1}$$

where $\overline{x_i}$ intends to the value i normalized, being μ_x the mean and σ_x the standard deviation of the feature x, whose number of samples is inferred as n.

The mean and standard deviation have only been computed using the training data so that the models cannot access the values in test sets. Thereby, the test splits are normalized using this mean and standard deviation likewise.

3.2. Data Windowing

The models trained in this article will make predictions based on a window of consecutive samples from the data. Various data windows were generated to build and compare models in different scenarios.

The number of time steps of the input and label sequences is attached to the window size and horizon of prediction, respectively. Several widths were considered according to the sample time of 1 Hertz. Therefore, models could be trained on different window configurations (Figure 2).

The overlapping of the window slides was fixed to the label width.

The models developed make *single-output* and *multi-time-step* predictions using every feature previously explained as inputs and the capacity as the label.

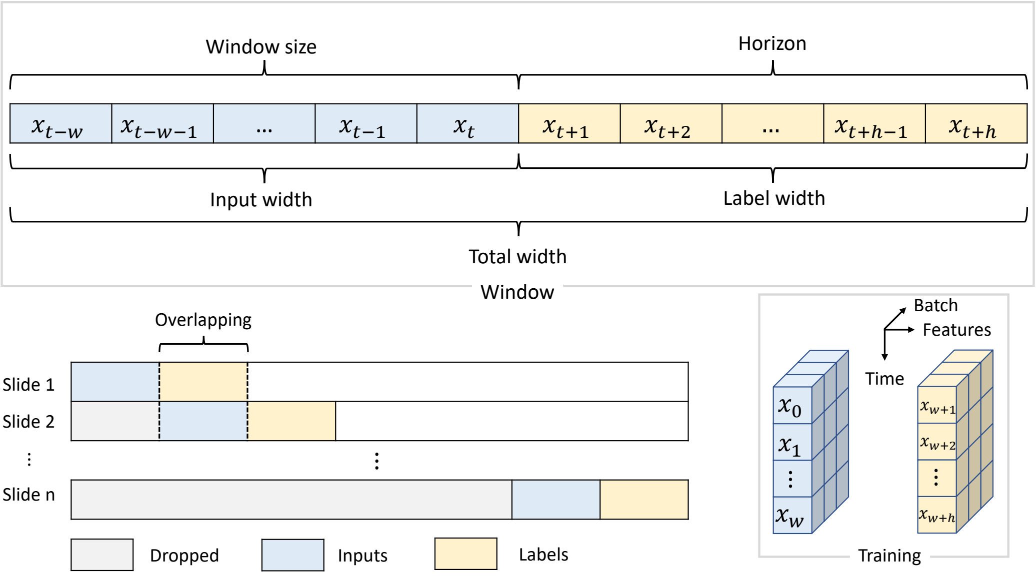

Figure 2. A fixed window slides on the available time series data in agreement with the overlapping, to build a dataset of input and label sequences, according to window size and forecasting horizon, respectively.

3.3. LSTM

LSTM Networks have been developed to overcome the vanishing gradient problem in the standard RNN by improving the gradient flow within the network [28]. This was achieved by using a LSTM unit in place of the hidden layer. As shown in Figure 3, a LSTM unit is composed of [28,50]:

Cell State: brings information along the entire sequence and represents the memory of the network.

$$C(t) = \sigma(f(t) \odot C(t-1) + i(t)) \tag{2}$$

Referring to $C(t)$ as the cell state at current time-step t, whereabouts σ alludes to sigmoid activation function, $f(t)$ the output of the forget gate, $C(t-1)$ the cell state at previous time-step $t-1$ and $i(t)$ the output of the input gate. The operator $\odot$ refers to Hadamard product [51].

Forget Gate: decides what is relevant to keep from previous time steps.

$$f(t) = \sigma(x(t)U_f + h(t-1)W_f) \tag{3}$$

Being $f(t)$ the output of the forget gate at current time-step t, $x(t)$ the inputs to the cell at current time-step t, $h(t-1)$ the hidden state at previous time-step $t-1$ whereupon the recurrent connection is fed back into the cell. U and W are the weights of the inputs and recursive state each.

Input Gate $\boldsymbol{i(t)}$: manages what information is relevant to add from the current time step.

$$i_1(t) = \sigma(x(t)U_i + h(t-1)W_i) \tag{4}$$

$$i_2(t) = tanh(x(t)U_g + h(t-1)W_g) \tag{5}$$

$$i(t) = i_1(t) \odot i_2(t) \tag{6}$$

where $i_1(t)$ and $i_2(t)$ are hereby the main components of the input gate $i(t)$ output at current time-step t. *tanh* infers to hyperbolic tangent activation function.

Output Gate: computes the value of the output at current time step.

$$o(t) = \sigma(x(t)U_o + h(t-1)\ W_o) \tag{7}$$

$$h(t) = tanh(C_t) \odot o(t) \tag{8}$$

Here, $o(t)$ at current time-step t, derives from merging the inputs $x(t)$ with the hidden state $h(t-1)$. Concluding $h(t)$ as the output of the cell.

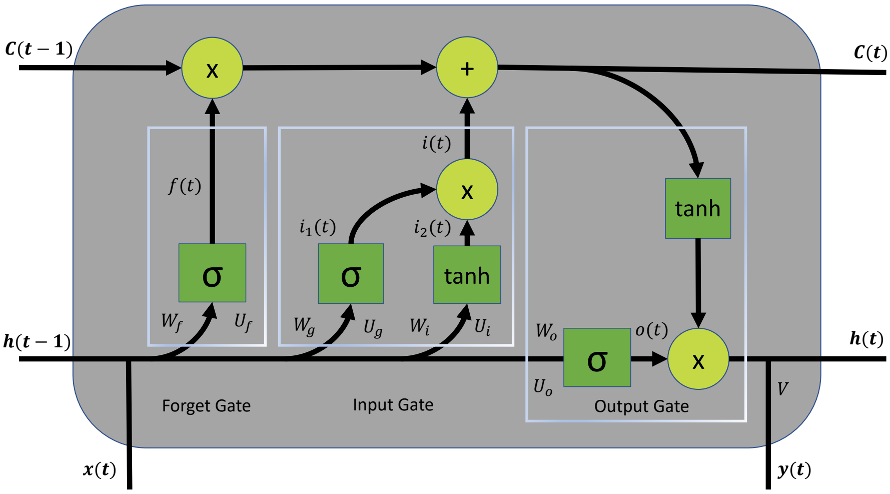

Figure 3. A LSTM unit scheme. It is composed of the cell state, forget gate, input gate and output gate. U and W represent the weights of inputs and recurrent connections for the internal layers within each gate. σ and *tanh* alludes to sigmoid and hyperbolic tangent activation functions each.

As a type of RNN, LSTM networks have loops, allowing information to persist [22–24]. Figure 4 considers how it flows between time steps by looking at some input x_t and computing an output value y_t, feeding back the internal state ($C(t)$, $h(t)$) of the unit from one step to the next.

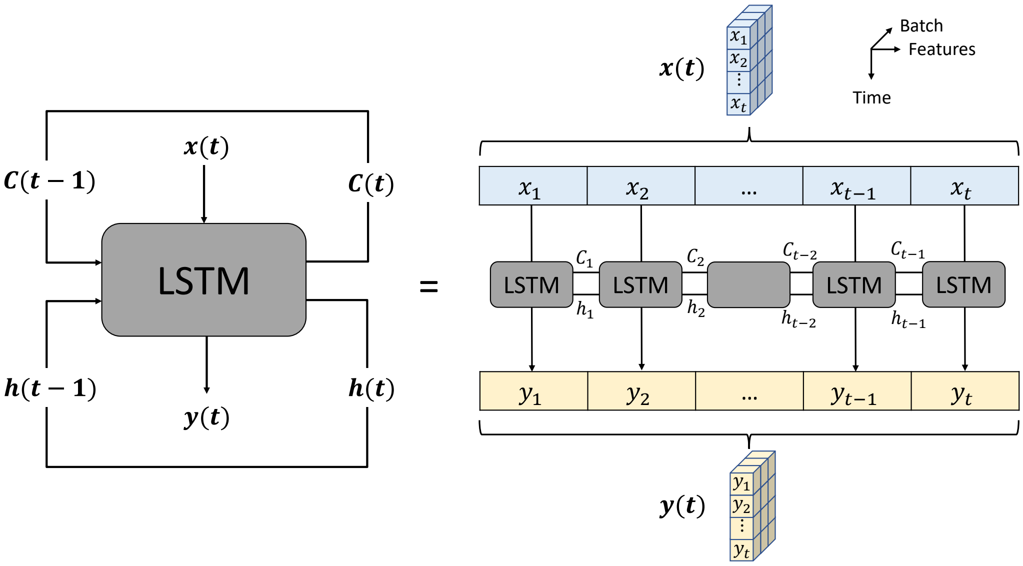

Figure 4. An unrolled LSTM network. The LSTM unit places the hidden layer and computes inputs recursively for each time step.

4. Experiments and Results

Throughout this point, it describes all the experiments proposed and the results obtained for each simulation carried out.

4.1. Experiments

As introduced in Section 3.2, the proposed models will make *single-output* predictions of the capacity of the battery and *multi-time-step* forecasting according to two different approaches as shown in Figure 5: Single-shot (SS), making predictions all at once and Autoregressive (AR), making one prediction at a time and feeding the output back to the model [52,53].

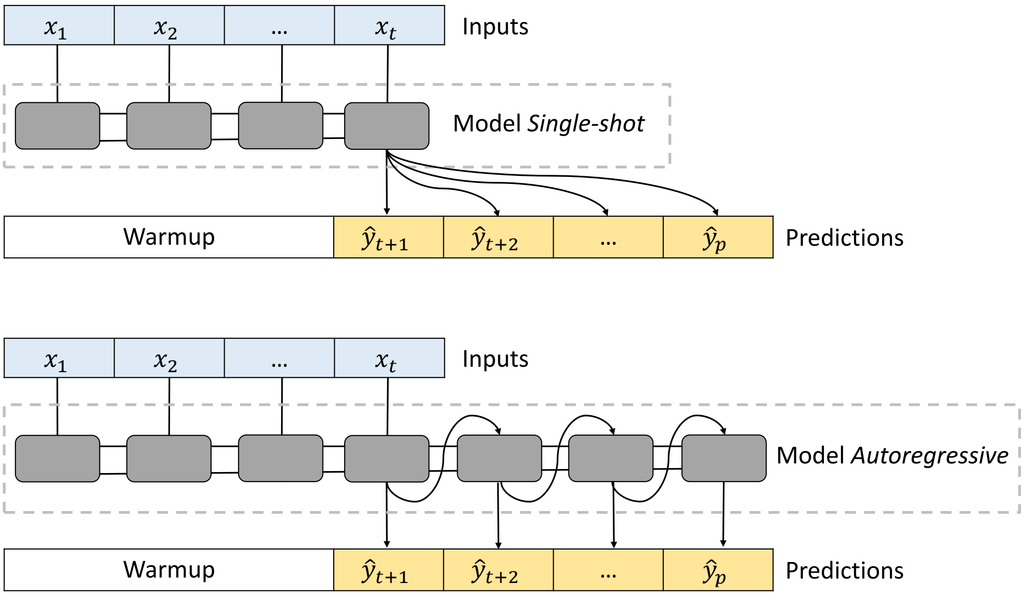

Figure 5. Model configurations are considered to forecast multiple time steps: Single-shot (SS) and Autoregressive (AR).

The sequence of steps involved in creating the intelligent models has been conducted through an Application Programming Interface (API) specifically developed for this purpose. Mainly written in Python language [54], it deals with all of the functionalities required for launching the experiments proposed throughout the workflow. From constructing the datasets of windowed samples to the training and evaluation stages. Therefore, celebrated software libraries for machine learning and artificial intelligence, such as TensorFlow and Keras were used [55] to build the models according to the architecture based on LSTM units. With this, a specific class integrated into TensorFlow through a Keras module deals with the layer's behavior but also for the optimization stage. After that, the KerasTuner optimization framework is employed.

Besides, different window widths were used to build up datasets (as stated in Figure 2) employed to train the models to output predictions in varied scenarios, using the previous time step feature values to forecast the capacity across the horizon of prediction.

The pipeline followed to obtain models steadied to forecast incoming data is summarized in Figure 6. During the optimization stage, the performance of different versions of model configurations (Figure 5) was evaluated using the Cross-Validation (CV) resampling method to estimate the generalization at long therm and Mean Squared Error (MSE) metric as the objective function to minimize.

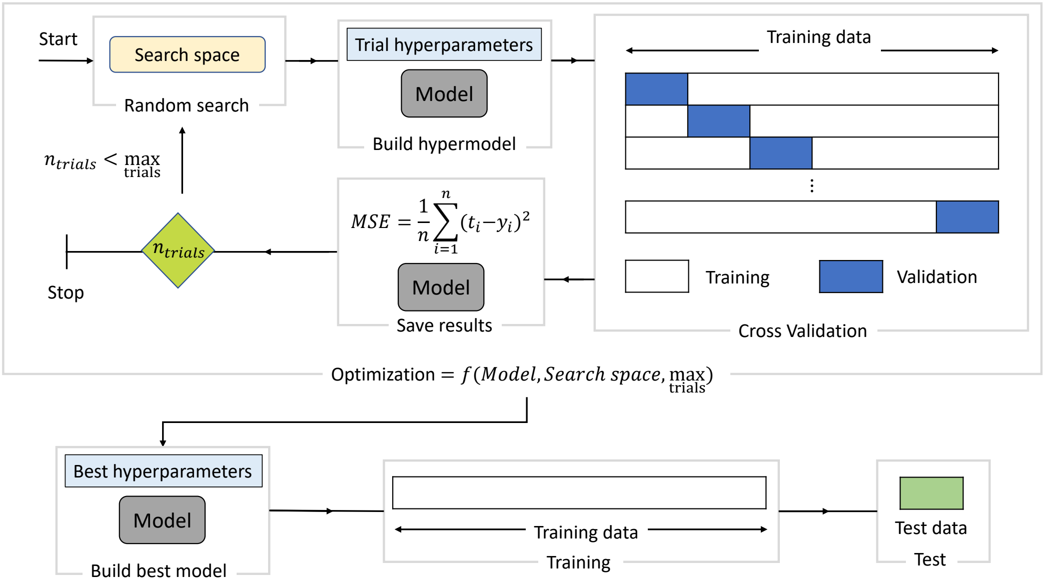

Figure 6. Optimization, Training and performance stage procedure for the models evaluated for each configuration and case of study.

The best hyperparameters obtained were used to build up the final models and fit them on the whole training set. Then, the performance was evaluated on the hold-out unseen test set simulating new incoming data. The search space design by the hyperparameters range of the models and the training settings are presented in Table 2.

Table 2. Hyperparameters space and Training parameter settings.

Hyperparameters Space			**Training Parameters**	
LSTM Units	**Neurons**	**Batch Size**	**K-Folds**	**Epochs**
[1–3]	[32–40]	[32–40]	10	50

4.2. Results

Several simulations were carried out, with the considerations mentioned above, for the window widths representing each scenario. These are 15, 30, 60, and 120 samples, meaning in terms of real-time, lookup historical data and forecasting horizons of 15 s, 30 s, 1 min., and 2 min. according to the time step stated by the sampling time of 1 Hertz.

Table 3 shows the metric scores for each model configuration in every case of study described by different window sizes and forecasting horizons. Several metrics were necessary to deal with potentially biased comparisons between models that can mislead meaningful interpretations.

Table 3. Metric results for model and window case of study.

		Windows			
Model	**Metrics**	**15**	**30**	**60**	**120**
SS	MAE (%)	0.148933	0.250346	0.366923	0.766663
	MSE	0.000299	0.000600	0.001315	0.005677
	MAPE (%)	7.259039	9.656219	9.977581	54.853806
	R^2	0.999738	0.999473	0.998836	0.994919
AR	MAE (%)	0.152125	0.251448	0.446577	1.281911
	MSE	0.000248	0.000666	0.001812	0.013689
	MAPE (%)	4.797170	5.755136	11.760392	77.755592
	R^2	0.999782	0.999416	0.998395	0.987729

Observing Figure 7, the SOC value for confirming the battery capacity can be predicted in real-time. The goodness of the fit in terms of the coefficient of determination (R^2) appears to be quite regular even for the largest window settings, despite the invaluable fact that the robustness and oscillation around real values are considerably much more stable upon conservative horizons.

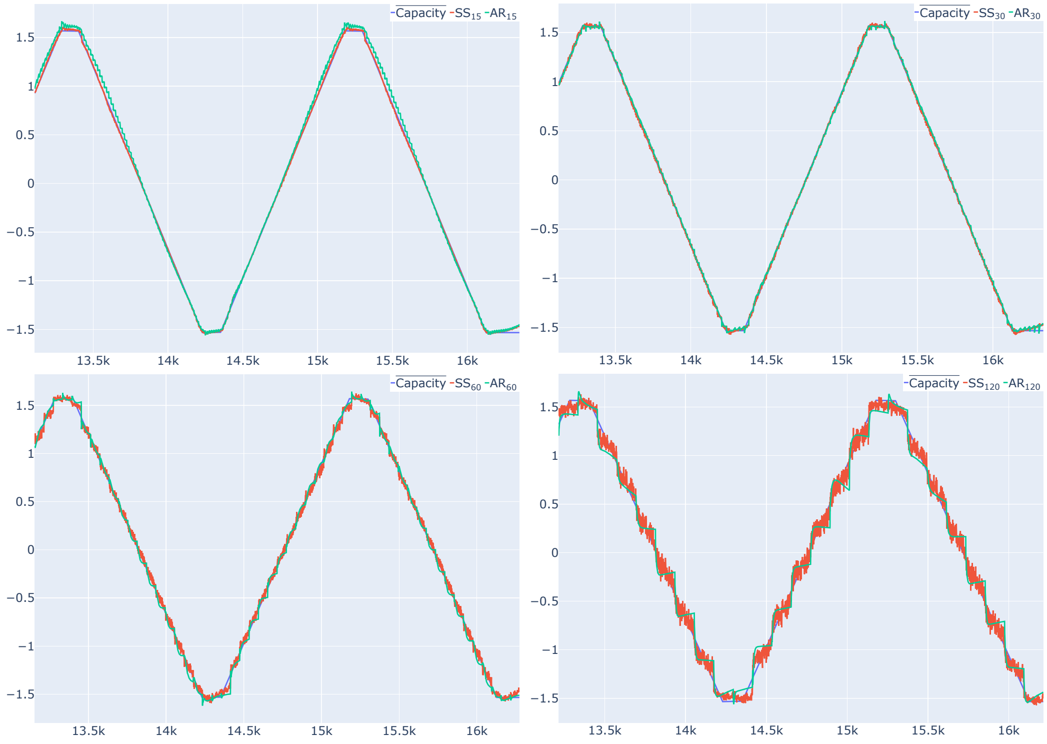

Figure 7. Predicted scaled values for the Capacity of the battery over the hold-out test set for the SS and AR model configurations in each window scenario.

As well, regarding Table 3, promising scores of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) are attained, assimilating other related studies [56]. MAE values vary from 0.1489 to 0.7667 with an increment of 68, 47, and 109% between window gaps. And 0.1521 to 1.2819 with 65, 78, and 187% for the SS and AR models. At the same time, MSE exhibits a range of scores from 0.0003 to 0.0056 with a growth of 100, 119, and 232% amid prediction horizons. This MSE goes from 0.0002 to 0.0137 by 168, 172, and 655% for the SS and AR models. Likewise, the MAPE scores follow a similar trend, presenting ©54.8538 for the SS configuration and 4.7971 to 77.7556 for the AR, with the apex at 2 min away in both cases. From the standpoint of the R^2, the results are considerably steady, through 0.9997 and 0.9949 for the SS model and in like manner through 0.9997 to 0.9877 for the AR.

With this exposed, it may be noticeable that the SS configuration achieves better results than the AR approach, especially for the widest windows, intuiting an elbow around 60 s width (Figure 8). Indeed, the consistency of such predictions vanishes, conforming to the window size, and the forecasting horizon increases. In addition, The assessment of the MAE metric along with the MSE, MAPE, and R^2 may hint that the SS model works better than AR, particularly for extremist widows. Even though for the most conservative cases, no evidence may assure the same.

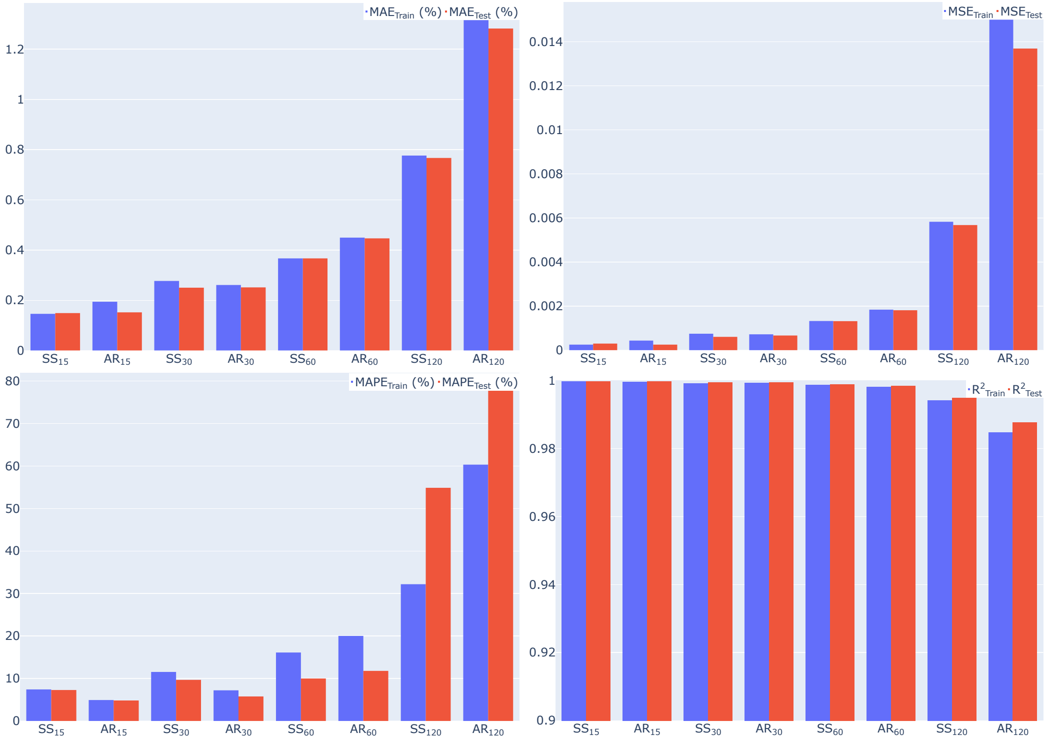

Figure 8. Bar chart representing metric results for model and window case of study.

4.3. Statistical Analysis

Table 3 provides a lead to reckon the models' performance expected facing new data. These results were obtained by fitting the models over the full training set and evaluating them on the hold-out test set following the track described in Figure 6.

Nonetheless, to be able to determine if eventually there is any evidence of which model configuration and window settings work significantly better than the rest, statistical discrimination is needed.

In line with the aforementioned, Figure 9 shows the boxplot taking advantage of the metric results obtained for the best models in each fold of the CV applied during the optimization stage. Variance analysis is carried out to settle if there are significant differences in the score means. Methods applied such as Kruskall-Wallis [57], and ANOVA [57] ascertained that the null hypothesis, which states that all models have the same metric scores, can be rejected with a confident interval of 95%.

In case the null hypothesis is rejected, a step further must be taken to identify which models are different using multiple comparative analyses, as methods of Tukey [58] and

Holm-Bonferroni [59], which compare differences between each pair of means with the proper adjustment for the multiple comparisons.

All the resulting *p*-values of the multiple comparative analysis are collected in Table 4. Those under 0.05 and thus, outside the confident interval of 95% are marked in bold, bespeaking than certainly there is a significant difference between that pair of models.

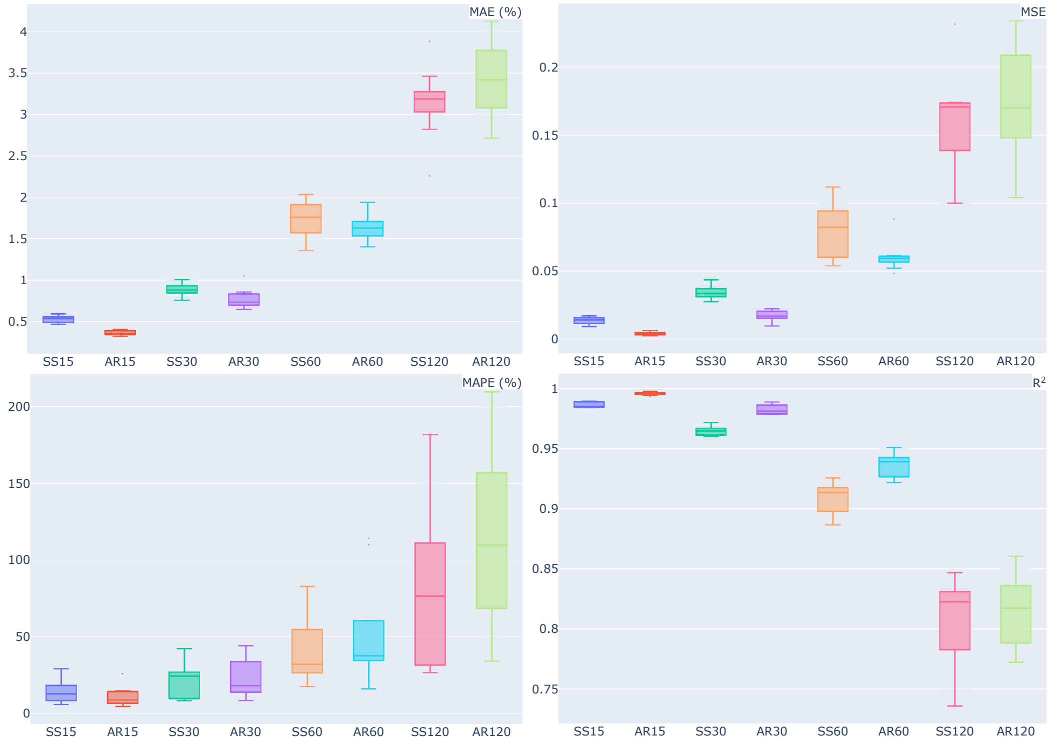

Figure 9. Boxplots of the metric results obtained for the best models cross-validated in each window width.

Table 4. *p*-Values determined by the muti-comparison methods of Tukey and Holm-Bonferroni for each metric and pair of models.

		***p*-Values**							
		MAE		**MSE**		**MAPE**		**R^2**	
Models		**Holm**	**Tukey**	**Holm**	**Tukey**	**Holm**	**Tukey**	**Holm**	**Tukey**
AR$_{120}$	AR$_{15}$	**0**	**0.001**	**0**	**0.001**	**0.0347**	**0.001**	**0.0002**	**0.001**
AR$_{120}$	AR$_{30}$	**0**	**0.001**	**0**	**0.001**	**0.049**	**0.001**	**0.0002**	**0.001**
AR$_{120}$	AR$_{60}$	**0**	**0.001**	**0.0001**	**0.001**	0.3487	**0.001**	**0.0007**	**0.001**
AR$_{120}$	SS$_{120}$	0.2262	0.0906	0.7203	0.9	0.6894	0.1168	0.436	0.6574
AR$_{120}$	SS$_{15}$	**0**	**0.001**	**0**	**0.001**	**0.049**	**0.001**	**0.0002**	**0.001**
AR$_{120}$	SS$_{30}$	**0**	**0.001**	**0**	**0.001**	0.0578	**0.001**	**0.0004**	**0.001**
AR$_{120}$	SS$_{60}$	**0**	**0.001**	**0.0004**	**0.001**	0.1697	**0.001**	**0.0032**	**0.001**
AR$_{15}$	AR$_{30}$	**0**	0.0988	**0.0001**	0.9	0.6894	0.9	**0.0002**	0.9
AR$_{15}$	AR$_{60}$	**0**	**0.001**	**0**	**0.0071**	0.2934	0.4679	**0**	**0.0027**
AR$_{15}$	SS$_{120}$	**0**	**0.001**	**0.0002**	**0.001**	0.0514	**0.0011**	**0**	**0.001**
AR$_{15}$	SS$_{15}$	**0.0012**	0.9	**0.0004**	0.9	1	0.9	**0.0002**	0.9
AR$_{15}$	SS$_{30}$	**0**	**0.0042**	**0**	0.1262	0.6123	0.9	**0**	0.0777
AR$_{15}$	SS$_{60}$	**0**	**0.001**	**0**	**0.001**	**0.049**	0.8011	**0**	**0.001**
AR$_{30}$	AR$_{60}$	**0**	**0.001**	**0**	0.1233	0.3478	0.7034	**0**	0.0728
AR$_{30}$	SS$_{120}$	**0**	**0.001**	**0.0004**	**0.001**	**0.046**	**0.0044**	**0**	**0.001**
AR$_{30}$	SS$_{15}$	**0**	0.5479	**0.0282**	0.9	0.6894	0.9	**0.016**	0.9
AR$_{30}$	SS$_{30}$	**0.0092**	0.9	**0**	0.6601	1	0.9	**0**	0.5819
AR$_{30}$	SS$_{60}$	**0**	**0.001**	**0**	**0.001**	0.3478	0.9	**0**	**0.001**

Table 4. *Cont.*

		***p*-Values**							
		MAE		**MSE**		**MAPE**		**R^2**	
Models		**Holm**	**Tukey**	**Holm**	**Tukey**	**Holm**	**Tukey**	**Holm**	**Tukey**
AR_{60}	SS_{120}	**0.0001**	**0.001**	**0.0009**	**0.001**	0.6599	0.3039	**0**	**0.001**
AR_{60}	SS_{15}	**0**	**0.001**	**0**	0.0523	0.0903	0.5032	**0**	**0.0235**
AR_{60}	SS_{30}	**0**	**0.0041**	**0.0282**	0.9	0.6123	0.8048	**0.016**	0.9
AR_{60}	SS_{60}	**0.0215**	0.7539	**0.0021**	0.3463	1	0.9	**0.0023**	0.2844
SS_{120}	SS_{15}	**0**	**0.001**	**0.0004**	**0.001**	**0.046**	**0.0014**	**0**	**0.001**
SS_{120}	SS_{30}	**0**	**0.001**	**0.0008**	**0.001**	0.0651	**0.0077**	**0**	**0.001**
SS_{120}	SS_{60}	**0.0002**	**0.001**	**0.0066**	**0.001**	0.3487	0.0922	**0**	**0.001**
SS_{15}	SS_{30}	**0**	0.0683	**0**	0.4552	0.6894	0.9	**0**	0.3292
SS_{15}	SS_{60}	**0**	**0.001**	**0**	**0.001**	0.0903	0.8349	**0**	**0.001**
SS_{30}	SS_{60}	**0**	**0.001**	**0**	**0.0326**	0.6894	0.9	**0**	**0.0184**

p-vlaues under 0.05, and thus outside the confidence interval of 95%, are marked in bold.

5. Conclusions and Future Works

The obtained results in this study report a system able to predict in advance and with enough accuracy the SOC of a battery to perform corrective actions to improve the energy management in batteries (charge, discharge) and its potential applications in different fields such as an electric vehicle, smart grids, . . .

The simulations launched in this study may present remarkable results like in other akin studies conducted [56]. Nevertheless, the performance of the model configurations can differ significantly in agreement with the window settings considered.

In this line, as far as the statistical analysis is concerned, according to Kruskall-Wallis and ANOVA variance analysis, there are expected differences between the model scores, which are specifically denoted by Tukay and Holm-Bonferroni multiple comparative analysis. With the help of these methods, it can be stated that statistical variation between models arises when the window width contrast is emphasized. As long as there are systematically significant differences among both SS and AR configurations with a sameness of forecasting horizon, choosing the one with the best scores seems to be a wise approach.

The SOC value for the confirmation of the battery capacity can be predicted in real-time by both models. When the power cell test gives incorrect results, the model can be used to verify that it is working properly.

Applications of interest may be fulfilled by these models, supported by the metric results. These models can provide a beneficial way to optimize the energy management in power cells during charge or discharge cycles.

Testing its capacity to deal with large sequences throw varied window widths is the main reason which made LSTM networks the object of study in this article. Even though GRUs have been shown to exhibit better performance on specific smaller and less frequent datasets [60,61], there is no consensus on the analogy between LSTM or GRU units in time series forecasting, leaving the computational cost out. It may open a door for future multi-comparison analysis employing different model configurations based on these units.

As future works, it is possible to mention that we will try to develop models for different types of batteries to check if the specific internal battery components affect the performance of the model.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: CITIC, as a Research Center of the University System of Galicia, is funded by Consellería de Educación, Universidade e Formación Profesional of the Xunta de Galicia through the European Regional Development Fund (ERDF) and the Secretaría Xeral de Universidades (Ref. ED431G 2019/01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chukwuka, C.; Folly, K.A. Batteries and super-capacitors. In Proceedings of the IEEE Power and Energy Society Conference and Exposition in Africa: Intelligent Grid Integration of Renewable Energy Resources, PowerAfrica, Johannesburg, South Africa, 9–13 July 2012; pp. 1–6.
2. Atawi, I.E.; Hendawi, E.; Zaid, S.A. Analysis and Design of a Standalone Electric Vehicle Charging Station Supplied by Photovoltaic Energy. *Processes* **2021**, *9*, 1246. [CrossRef]
3. Qian, H.; Zhang, J.; Lai, J.-S. A grid-tie battery energy storage system. In Proceedings of the 2010 IEEE 12th Workshop on Control and Modeling for Power Electronics, COMPEL, Boulder, CO, USA, 28–30 June 2010; pp. 1–5.
4. Chaturvedi, N.; Klein, R.; Christensen, J.; Ahmed, J.; Kojic, A. Modeling, estimation, and control challenges for lithium-ion batteries. In Proceedings of the American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 1997–2002.
5. Vukosavic, S. *Electrical Machines, Power Electronics and Power*; Springer: New York, NY, USA, 2012.
6. Ismail, N.H.F.; Toha, S.F. State of charge estimation of a Lithium-ion battery for electric vehicle based on particle swarm optimization. In Proceedings of the 2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2013, Kuala Lumpur, Malaysia, 25–27 November, 2013; pp. 1–4.
7. Swierczynski, M.; Stroe, D.-I.; Stan, A.-I.; Teodorescu, R.; Kær, S.K. Lifetime estimation of the nanophosphate lifepo4 battery chemistry used in fully electric vehicles. *IEEE Trans. Ind. Appl.* **2015**, *51*, 3453–3461. [CrossRef]
8. Ant'on, J.C.A.; Nieto, P.J.G.; Juez, F.J.d.; Lasheras, F.S.; Viejo, C.B.; Gutierrez, N.R. Battery state-of-charge estimator using the MARS technique. *IEEE Trans. Power Electron.* **2013**, *28*, 3798–3805. [CrossRef]
9. Zhang, D.; Du, L.; Gao, Z. Real-Time Parameter Identification for Forging Machine Using Reinforcement Learning. *Processes* **2021**, *9*, 1848. [CrossRef]
10. Zhang, D.; Gao, Z. Improvement of Refrigeration Efficiency by Combining Reinforcement Learning with a Coarse Model. *Processes* **2019**, *7*, 967. [CrossRef]
11. Jove, E.; Gonzalez-Cava, J.M.; Casteleiro-Roca, J.L.; Quintián, H.; Méndez Pérez, J.A.; Vega Vega, R.; Zayas-Gato, F.; de Cos Juez, F.J.; León, A.; Martón, M.; et al. Hybrid Intelligent Model to Predict the Remifentanil Infusion Rate in Patients Under General Anesthesia. *Log. J. IGPL* **2020**, *29*, 193–206. [CrossRef]
12. Liao, T.W. Clustering of time series data—A survey. *Pattern Recognit.* **2005**, *38*, 1857–1874. [CrossRef]
13. Aghabozorgi, S.; Shirkhorshidi, A.S.; Wah, T.Y. Time-series clustering—A decade review. *Inf. Syst.* **2015**, *53*, 16–38. [CrossRef]
14. Keogh, E.J. On the need for time series data mining benchmarks. *Data Min. Knowl. Discov.* **2003**, *7*, 349–371. [CrossRef]
15. Agrawal, R.; Faloutsos, C.; Swami, A. Efficient Similarity Search In Sequence Databases. In Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms, Chicago, IL, USA, 13–15 October 1993; Springer: Berlin/Heidelberg, Germany, 1993; Volume 730, pp. 69–84.
16. Chen, C.W.; Chiu, L.M. Ordinal Time Series Forecasting of the Air Quality Index. *Entropy* **2021**, *23*, 1167. [CrossRef]
17. Tang, Z.; Tang, L.; Zhang, G.; Xie, Y.; Liu, J. Intelligent Setting Method of Reagent Dosage Based on Time Series Froth Image in Zinc Flotation Process. *Processes* **2020**, *8*, 536. [CrossRef]
18. Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14413–14423. [CrossRef]
19. Ciresan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June, 2012; pp. 3642–3649.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the NIPS 2012: Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012.
21. Google's AlphaGo AI wins Three-Match Series against the World's Best Go Player. 2018. Available online: https://www.bbc.com/news/technology-40042581 (accessed on 5 July 2022).
22. Dupond, S. A thorough review on the current advance of neural network structures. *Annu. Rev. Control.* **2019**, *14*, 200–230.
23. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef] [PubMed]
24. Tealab, A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Comput. Inform. J.* **2018**, *3*, 334–340. [CrossRef]
25. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 855–868. [CrossRef]
26. Sak, H.; Senior, A.; Beaufays, F. *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*; Interspeech: Singapore, 2014.

27. Li, X.; Wu, X. Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015.
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
29. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
30. Yeow, K.; Teng, H.; Thelliez, M.; Tan, E. Thermal analysis of a li-ion battery system with indirect liquid cooling using finite element analysis approach. *Sae Int. J. Altern. Powertrains* **2012**, *1*, 65–78. [CrossRef]
31. Shao-Horn, Y.; Croguennec, L.; Delmas, C.; Nelson, E.C.; O'Keefe, M.A. Atomic resolution of lithium ions in licoo2. *Nat. Mater.* **2003**, *2*, 464–467. [CrossRef]
32. Julien, C. Local structure of lithiated manganese oxides. *Solid State Ion.* **2006**, *177*, 11–19. [CrossRef]
33. Chung, S.-Y.; Bloking, J.T.; Chiang, Y.-M. Electronically conductive phospho-olivines as lithium storage electrodes. *Nat. Mater.* **2002**, *1*, 123–128. [CrossRef] [PubMed]
34. Cakici, M.; Kakarla, R.R.; Alonso-Marroquin, F. Advanced electrochemical energy storage supercapacitors based on the flexible carbon fiber fabric-coated with uniform coral-like mno2 structured electrodes. *Chem. Eng. J.* **2017**, *309*, 151–158. [CrossRef]
35. Ma, Y.; Chang, H.; Zhang, M.; Chen, Y. Graphene-based materials for lithium-ion hybrid supercapacitors. *Adv. Mater.* **2015**, *27*, 5296–5308. [CrossRef]
36. Beard, K.W. *Linden's Handbook of Batteries*; McGraw-Hill Education: New York, NY, USA, 2019.
37. Yi, J.; Kim, U.S.; Shin, C.B.; Han, T.; Park, S. Three-dimensional thermal modeling of a lithium-ion battery considering the combined effects of the electrical and thermal contact resistances between current collecting tab and lead wire. *J. Electrochem. Soc.* **2013**, *160*, A437. [CrossRef]
38. Arora, P.; White, R.E.; Doyle, M. Capacity fade mechanisms and side reactions in lithium-ion batteries. *J. Electrochem. Soc.* **1998**, *145*, 3647. [CrossRef]
39. Takahashi, M.; Tobishima, S.I.; Takei, K.; Sakurai, Y. Reaction behavior of lifepo4 as a cathode material for rechargeable lithium batteries. *Solid State Ion.* **2002**, *148*, 283–289. [CrossRef]
40. Pan, Y.; Chou, S.; Liu, H.K.; Dou, S.X. Functional membrane separators for next-generation high-energy rechargeable batteries. *Natl. Sci. Rev.* **2017**, *4*, 917–933. [CrossRef]
41. Hoke, A.; Brissette, A.; Smith, K.; Pratt, A.; Maksimovic, D. Accounting for lithium-ion battery degradation in electric vehicle charging optimization. *IEEE J. Emerg. Sel. Top. Power Electron.* **2014**, *2*, 691–700. [CrossRef]
42. Neubauer, J.; Wood, E. *Will Your Battery Survive a World with Fast Chargers*? Technical Report; National Renewable Energy Lab.(NREL): Golden, CO, USA, 2015.
43. Lu, Z.; Yu, X.; Wei, L.; Cao, F.; Zhang, L.; Meng, X.; Jin, L. A comprehensive experimental study on temperature-dependent performance of lithium-ion battery. *Appl. Therm. Eng.* **2019**, *158*, 113800. [CrossRef]
44. Zhou, F.; Bao, C. Analysis of the lithium-ion battery capacity degradation behavior with a comprehensive mathematical model. *J. Power Sources* **2021**, *515*, 230630. [CrossRef]
45. 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. [CrossRef]
46. Sparacino, A.R.; Reed, G.F.; Kerestes, R.J.; Grainger, B.M.; Smith, Z.T. Survey of battery energy storage systems and modeling techniques. In Proceedings of the 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 22–26 July 2012; pp. 1–8.
47. Mauger, A.; Julien, C.M. Critical review on lithium-ion batteries: Are they safe? Sustainable? *Ionics* **2017**, *23*, 1933–1947. [CrossRef]
48. PNGV Battery Test Manual (3 2001). Available online: https://avt.inl.gov/sites/default/files/pdf/battery/pngv_manual_rev3b.pdf (accessed on 18 May 2022).
49. Report, T. *Lifebatt x-1p 8ah 38123 Cell*; Technical Report; LiFeBATT Ltd.: Eastwoodbury Lane, UK, 2009.
50. Sparacino, A.R.; Reed, G.F.; Kerestes, R.J.; Grainger, B.M.; Smith, Z.T. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471.
51. Million, T. The Hadamard Product. 2007. Available online: http://buzzard.ups.edu/courses/2007spring/projects/million-paper.pdf (accessed on 22 June 2022).
52. Shahid, F.; Zameer, A.; Muneeb, M. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos Solitons Fractals* **2020**, *140*, 110212. [CrossRef] [PubMed]
53. Salman, A.G.; Heryadi, Y.; Abdurahman, E.; Suparta, W. Single Layer and Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting. *Procedia Comput. Sci.* **2018**, *135*, 89–98. [CrossRef]
54. Python Software Foundation. Python Language Reference, Version 2.7. Available online: http://www.python.org (accessed on 25 June 2022).
55. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 25 June 2022).
56. Casteleiro-Roca, J.L.; Jove, E.; Sánchez-Lasheras, F.; Méndez-Pérez, J.A.; Calvo-Rolle, J.L.; de Cos Juez, F.J. Power Cell SOC Modelling for Intelligent Virtual Sensor Implementation. *J. Sens.* **2017**, *2017*, 9640546. [CrossRef]

57. Kruskal–Wallis H Test Using SPSS Statistics, Laerd Statistics. Available online: https://statistics.laerd.com/spss-tutorials/kruskal-wallis-h-test-using-spss-statistics.php (accessed on 25 June 2022).
58. Lowry, R. One Way ANOVA—Independent Samples. Available online: http://facultysites.vassar.edu/lowry/PDF/c14p1.pdf (accessed on 25 June 2022).
59. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.
60. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
61. Gruber, N.; Jockisch, A. Are GRU cells more specific and LSTM cells more sensitive in motive classification of text? *Front. Artif. Intell.* **2020**, *3*, 40. [CrossRef]