A selective hybrid system for State-of-Charge forecasting of Lithium-ion batteries

Marie Chantelle Cruz Medina (mccm@ecomp.poli.br)
Universidade de Pernambuco
João Fausto Lorenzato de Oliveira
Universidade de Pernambuco

Research Article

Keywords: State-of-charge, Lithium-ion battery, Machine learning; Time series forecasting; Hybrid systems

Posted Date: October 21st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2167906/v1

A selective hybrid system for State-of-Charge forecasting of Lithium-ion batteries

Marie Chantelle Cruz Medina[1*] and João Fausto L. de Oliveira[1]

[1*]Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, 50720-001, Pernambuco, Brazil.

*Corresponding author(s). E-mail(s): mccm@ecomp.poli.br; Contributing authors: fausto.lorenzato@upe.br;

Abstract

As research in Lithium-ion batteries field has extended, the need for better management systems also increases. An important part of them is the proper estimation of battery status over time with indirect metrics such as State-of-Charge (SoC). In the machine learning environment, different simple techniques have been tested showing good performance and being surpassed by hybrid systems. In this study, a static selection model is proposed to choose the best non-linear predictor to work with and ARIMA model and combination function for a specific database considering how the perform in a validation set. This architecture allows to consider linear and non-linear components of the time series using residual forecasting and a selection step to reduce the chances of choosing the wrong combination in each specific database. SoC values for five different databases where forecasted, allowing to compare the results of six models relevant in literature to the proposed one. The results showed a superior performance for the proposed model in four out of the five databases, with gains of 5.27%, 13.51%, 56.67%, and 38.71%. There was only one database where the proposed model scored in second place. To determine whether the obtained results were statistically significant enough to make a conclusion, a Nemenyi test was conducted using MSE and MAE values to rank the performance of all models in all databases. The critical distance and rank achieved by the proposed model allowed to conclude that this, in fact, delivered the best performance amongst all models tested.

Keywords: State-of-charge, Lithium-ion battery, Machine learning; Time series forecasting; Hybrid systems

1 Introduction

Lithium-ion (Li-ion) batteries currently stand as one of the most vital energy storage strategies, given their applicability as power sources for mobiles, laptops, everyday electronics, power backups, and electric vehicles (EVs), [1–4] among others. Furthermore, storage-focused systems use battery packs made of numerous lithium-ion batteries, which require close monitoring of the battery states to maintain safe and efficient operation [5]. Thus, battery management requires estimating the battery's states, such as the state-of-charge (SoC), state-of-health (SoH), state-of-power (SoP), and state-of-life (SoL).

The State-of-Charge is known as the remaining usable percentage of the capacity of a battery. Therefore, a 100% SoC indicates the availability of use for the maximum capacity, while a 0% SoC suggests the opposite. Furthermore, SoC also provides information on a system's reliability, efficiency, and safety [6]. Accurate SoC estimations can help battery management systems with cell balancing in packs, avoiding cell over-charging and over-discharging, which can result in battery damage [7]. However, the State-of-Charge is not directly measurable since it depends on other factors within the battery and requires separate calculation. The development of data-driven algorithms has taken a significant step in recent years to enhance SoC measurements by improving the generalization performance, learning capabilities for high accuracy, and convergence [8]. In addition, recent work's conventional standalone machine learning techniques suffer from an accuracy standpoint and thus have been replaced by high-fidelity hybrid machine learning techniques [9].

Furthermore, SoC time series are often composed of linear and nonlinear patterns, which may not be dealt with equally well by a single forecasting method [10]. In this sense, traditional forecasting methods such as the ARIMA have been explored in SoC forecasting [9, 11] due to their well-established Box-Jenkins [12] methodology for model selection. However, despite the forecasting capabilities of the ARIMA model for nonstationary time series, the presence of nonlinear patterns can reduce its accuracy. On the other hand, nonlinear models such as artificial neural networks (ANNs) can learn nonlinear patterns. However, their accuracy is dependent on the selected hyperparameters, which lead to overfitted or under-fitted models when misspecified [10].

The ability to learn the data is often referred to as model bias, whereas the stability of the model when forecasting new samples is referred to as variance. Therefore, Data-driven models such as ANNs may learn well from data and present a slight *bias*. However, it may excessively rely on some samples, producing high variance [13]. Thus, a suitable trade-off between model bias and variance is desirable in forecasting systems since low-bias models could reflect high variance.

Considering the limitations of stand-alone models in forecasting tasks, several hybrid systems that decompose the time series into its linear and nonlinear components have been proposed to overcome the limitations of linear and nonlinear models and improve forecasting accuracy by mitigating the risk of selecting an inappropriate model. In general, the architecture of such hybrid systems is composed of three main steps: I) Time series forecasting, II) Residual Forecasting, and III) Combination. Usually, in the first step, a linear model such as ARIMA is employed to perform time series forecasts. In the second step, the residual from the ARIMA forecasts is calculated and used as input for a nonlinear model. Residual forecasting is a challenging task due to the presence of heteroscedastic patterns and may be affected by noise. The third and last step combines the forecasts obtained by the linear and nonlinear models.

Traditional studies in the literature assume a linear relation between linear and nonlinear forecasts; however, in several scenarios, this relation is better represented by nonlinear functions [14, 15]. Furthermore, other architectures have been proposed to improve prediction stability, reduce variance, and find the best model combinations. For example, Cruz and Oliveira[16] proposed a model with a weighted average combination of an ARIMA+SVR and a LSTM using R2 as a base metric to compute static weights to predict current and voltage time series. Models like this have been shown to perform adequately in SoC prediction. Nonetheless, introducing a selection process to adapt models and combinations to the architecture would greatly benefit overall performance, as this ensures the best adaptation to the characteristics of each time series.

This research proposes a novel hybrid system architecture to improve the forecasting accuracy of the hybrid systems presented in the SoC forecasting literature. The proposed hybrid system improvements can be achieved by the employment of two levels of selection. The first level is performed by selecting the non-linear model to be combined with linear forecasts according to the best-performing option in the validation set. The second level selects the best combination function for the outputs obtained from linear and selected nonlinear forecasts. As a result, the proposed system has the following advantages:

- Reduces the risk of selecting an inappropriate model by combining different forecasting systems.
- Increases model performance for each predicted dataset as the combination function also adapts to specific time series characteristics.
- Its versatility allows different models to be used in each step.

The remainder of this work is organized as follows: Section 2 presents the related studies and state of the art in the literature. In section 3, the proposed method is described. Then, the results are given in section 5. Finally, section 6 presents the conclusion and future studies.

2 Related Work

A relevant approach used to overcome the limitations of linear and nonlinear models is to decompose the time series into their linear and nonlinear components through a hybrid system. Zhang's [13] considered a time series Z_t to be a linear relation between its linear L_t and nonlinear N_t components as described in eq. 1

$$Z_t = L_t + N_t. \tag{1}$$

The first stage in the model proposed in [13] is to predict the forecasts of the linear component $\hat{L}_t$ using an ARIMA model. The residual (error) series is computed as the difference between the time series Z_t and linear forecasts $\hat{L}_t$ as presented in eq. 2.

$$E_t = Z_t - \hat{L}_t \tag{2}$$

Second, the residual series generated in last step is introduced nonlinear function f which considers k past residuals in order to perform nonlinear forecasts $\hat{N}_t$ as demonstrated in eq. 3.

$$\hat{N}_t = f(e_{t-1}, e_{t-2}, ..., e_{t-k}). \tag{3}$$

In the nonlinear function f, the number past information k represents the length of the time window. In the work of Zhang [13] a Multilayer Perceptron (MLP) neural network was considered as nonlinear model to perform the residual forecasts. The last step performs the combination of the linear and nonlinear forecasts, producing the final forecast $\hat{Z}_t$, as described in eq. 4.

$$\hat{Z}_t = \hat{L}_t + \hat{N}_t \tag{4}$$

For the sake of simplicity the architecture described previously, will be henceforward referred to as ARIMA+MLP, representing the summation of forecasts achieved by the ARIMA and MLP as linear and nonlinear models respectively.

Subsequent to Zhang's contribution, several models proposed in literature have follow the same architecture. Panigrahi and Behera [17] proposed an Exponential smoothing (ETS) to forecast the time series and a MLP to forecast residuals and a summation operator for the combination (ETS+MLP). Pai and Lin [18] proposed an ARIMA and Support Vector regression (ARIMA + SVR) to forecast stock pricing, Holanda and de Oliveira [19] employed a combination of models in the residuals using an ARIMA+PSO-SVR hybrid system. De Oliveira [20] proposed an alternative to Zhang's model using dynamic residual forecasting (DReF) aiming to reduce the uncertainty of model selection and avoiding deterioration of the time series forecast by finding the most suitable machine learning model from a pool to predict a specific pattern of the residual series and deciding whether it is a fitted option to increase the accuracy of the linear combination stage. The model is proved to perform satisfactory in several databases, which opens the opportunity for it to be implemented in an application-specific approach.

Despite the importance of such architecture to time series forecasting tasks, the assumption of linearity between linear and nonlinear forecasts can reduce the accuracy of the system, since the real combination function is unknown. In this sense, several other studies in the literature have investigated the nonlinear combination of linear $\hat{L}_t$ and nonlinear $\hat{N}_t$ forecasts. Lucena [21] proposed an ARIMA model to perform time series forecasting, a SVR model to perform residual forecasting, and the combination between them being performed by either an SVR or a MLP. In this sense, to simplify the notation, this architecture will be referred to as MLP(ARIMA,SVR) and SVR(ARIMA, SVR) representing the combination of ARIMA and SVR forecasts by a MLP ensemble. Moreover, Santos *et al.* [15] used a similar architecture, but incorporating past forecasting values as input for the combination function.

In the specific application of SoC forecasting, some models also follow a similar idea, as is the case of Khalid [9], combining a Minimized Akaike Information Criterion tuned ARIMA, and a unified Multilayer Perceptron (MLP) and then with Nonlinear Autoregressive Neural Network with external input (NARX). Cruz and Oliveira[16] with a weighted average combination of an ARIMA+SVR and a LSTM using R2 as a base metric to compute both static weights. In this model, current and voltage time series are first predicted and then coulomb counting technique is used to compute SoC values through time by integration. Moreover, the coulombs counting method could be implemented in a more sophisticated form as it is presented as the simple integration of current values in eq. 5.

$$SoC(t) = SoC_{t_o} + \int_{t_o}^{t_o+\Delta t} \frac{\alpha I_c + (\alpha - 1) I_d dt}{C_p}, \tag{5}$$

where C_p is the capacity of the battery, I_c is the charging current, and I_d is the discharging current. Also, this definition considers that each cycle of the battery registered is formed by two semi-cycles: charge (where only I_c exists) and discharge (where only I_d exists). The existence of each type of current in a given semi-cycle is determined in (8) as α, a coefficient equal to:

$$\alpha = \begin{cases} 0 & in\ discharge\ semi-cycle \\ 1 & in\ charge\ semi-cycle \end{cases}, \tag{6}$$

using α, the integral is simplified in both charge and discharge semi-cycles.

An alternative to improve eq. 5 is proposed by Mohammadi[22] with an improved coulomb counting algorithm (refered to as iCC) for lithium-ion batteries. Here, uncertainties from current measurement and integration are considered to shape eq. 7 for a one-hour analysis:

$$SoC(t) = SoC_{t_o} + \frac{\Delta t}{3600C}[i(t) + \epsilon_M(t) + \epsilon_I(t)], \tag{7}$$

Where some new variables appear in comparison to eq. 5, being Δt the period of observation to be computed, $i(t)$ representing current values without uncertainties, $\epsilon_M(t)$ representing the measurement uncertainty over time, and $\epsilon_I(t)$ the integration uncertainty over time. Typical values for $\epsilon_M(t)$ and $\epsilon_I(t)$ are also presented in [22] as functions of the battery capacity value with eqs. and :

$$\epsilon_M(t) \approx 0.018C \tag{8}$$

$$\epsilon_I(t) \approx 0.092C \tag{9}$$

The proposed model differs significantly from previous approaches proposed in the literature through the employment of two forecasting steps and introducing a selection approach to the non-linear model and combination function between linear and nonlinear forecasts. This step is important to mitigate problems related to uncertainty in model selection.

3 Proposed model

The proposed method follows the architecture of the proposed studies in the literature, and it is composed of three main stages:

- Linear modelling stage
- Nonlinear modelling stage
- Combination stage.

The stages are required to perform modelling of the time series, residual series and the combination of forecasts. The architecture of training process for the proposed model is presented in figure 1 whereas for the testing process is given in figure 2.

The main objective is to select an appropriate model for the forecast of residuals and for the combination of forecasts, which are fixed in several studies in the literature. Moreover, for each dataset under study, a new selection process is carried, resulting in a different forecasting system.

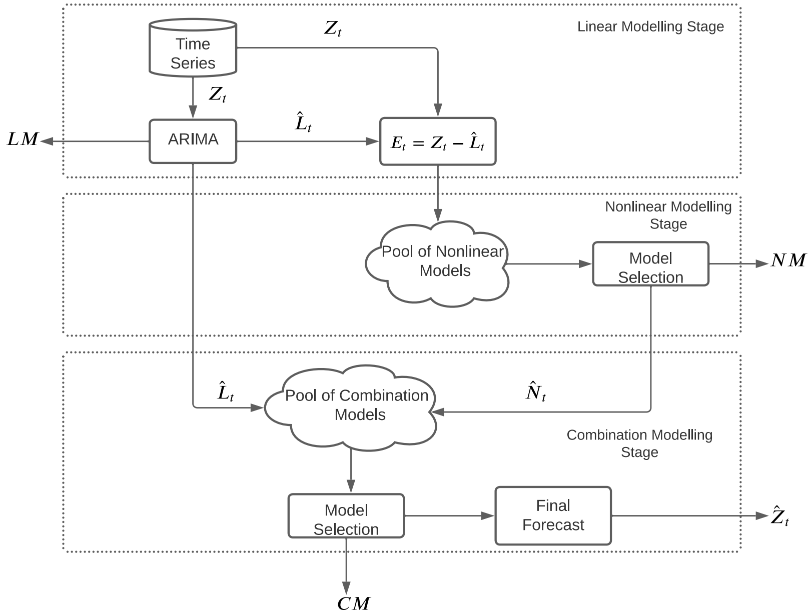


Fig. 1 Training process for the proposed model

3.1 Linear modelling stage

The first step in the pipeline of the architecture is to implement a linear model (LM) to perform a forecast $\hat{L}_t$ of the series. In this stage, an ARIMA model is employed to perform linear predictions in the time series, the training of the model is performed through a stepwise procedure [23] to select its most suitable parameters.

The residual calculation is then performed as presented in eq. 2 to be used as input in the next stage.

3.2 Nonlinear modelling stage

The residual forecasting process presented in eq.3 uses a single non-linear model to perform a forecast $\hat{N}_t$ of the residuals of the linear stage E_t. However, a single model may not be the best in every situation [24] and a proper selection procedure could be employed to enhance the forecasting accuracy of the system. In the proposed model, this step is expanded using a pool of non-linear models: Multi-Layer Perceptron (MLP), Decision Tree Regressor (DTR), and Support Vector Regressor (SVR). The training process of each models is performed individually, with the parameter selection performed through a grid search process.

The most suitable non-linear model (NM) for each time series prediction is then selected by evaluating all models in pool using a validation set, as this data is not contained in training nor testing sets.

3.3 Combination stage

The last stage performs the combinations of forecasts of linear ($\hat{L}_t$) and nonlinear $\hat{N}_t$ modeling stages. It is important to highlight that this stage performs a regression task in order to perform the nonlinear combination, that is, there are two inputs (linear and nonlinear forecasts) and one output which is the target value at time t. Therefore, we can consider that the combination model will represent a function CM as described in eq. 10.

$$\hat{Z}_t = CM(\hat{L}_t, \hat{N}_t). \tag{10}$$

Unlike other work in the literature [14, 15, 20], the proposed model selects the most suitable model to perform the combination of forecasts for a given data set. Furthermore, the pool of models in this stage is composed by the summation operator representing a linear combination function and three machine learning models SVR, DTR, and MLP to perform nonlinear combinations. It is important to mention that the selection process is performed on the validation set, which enables to select models with better generalization capacity.

3.4 Testing process

After training the ARIMA model LM and selecting appropriate models to forecast the residuals and perform the combination (NM and CM, respectively), they are used to perform the forecasting and combination of new patterns. Thus, the same stages performed in the training process are also required in the testing.

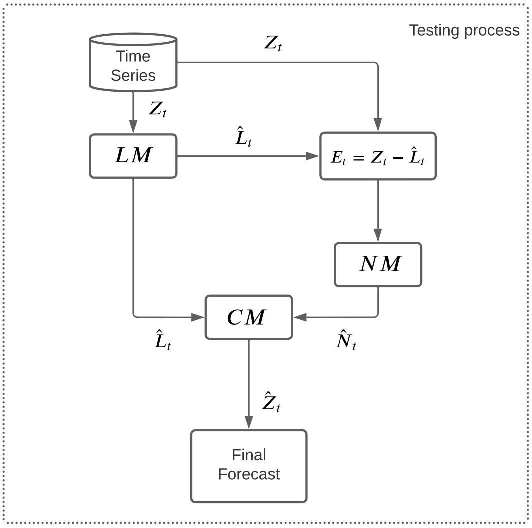


Fig. 2 Testing process for proposed model

Linear forecasting stage is performed through the employment of the linear model LM, which produces linear forecasts $\hat{L}_t$ and then calculates the residual $E_t = Z_t - \hat{L}_t$. In the nonlinear forecasting stage, the residual series E_t is used by the nonlinear model NM to produce nonlinear forecasts $\hat{N}_t$. The combination stage employed the combination model CM taking into consideration previous forecasts $\hat{L}_t$ and $\hat{N}_t$ to perform the final forecast $\hat{Z}_t$.

The testing procedure described in fig. 2 shows the employment of the selected models. It is important to mention that the CM model in the combination can be represented by linear nor nonlinear functions.

4 Experiments

To perform comparisons of different algorithms presented in the literature, different data sets concerning SoC of batteries with different characteristics are employed. The models are compared using several error metrics, and finally a hypothesis testing is employed to perform the final analysis.

4.1 Data sets

In order to test performance of the proposed model, tests were conducted on four time series representing charge and discharge cycles of Lithium-ion batteries with different characteristics. The data was all accessed through the Battery Archive repository [25], which allows to visualize, download, compare and filter battery data shared by five different institutions. The specific databases used in these experiments are:

1. Underwriters Laboratories Inc. – Purdue University (UL-PUR) [26].
2. Center for Advanced Life Cycle Engineering (CALCE) set.
3. Hawaii Natural Energy Institute (HNEI) [27].
4. Oxford University (OX) set.
5. Sandia National Laboratories (SNL) [28].

From each of these sets, one time series was selected to test the ability of the proposed model to forecast the state of charge of Lithium-ion batteries with graphite anode and different characteristics as summarized in Table 1, where C is the capacity of the battery in ampere hours, T is the temperature in Celsius degrees, Ch C-Rate is the speed at which the battery is fully charged, and Dis C-Rate is the speed at which the battery is fully discharged.

Table 1 Database characteristics

Dataset	**Cycles**	**Cathode**	**C [Ah]**	**Type**	**T[C]**	**Ch C-Rate**	**Dis C-Rate**
UL-PUR	301	NCA	3.40	18650	23	0.50	0.50
CALCE	2016	LCO	1.35	prismatic	25	0.50	0.50
HNEI	1113	NMC-LCO	2.80	18650	25	0.50	1.50
OX	8200	LCO	0.74	pouch	40	2.00	1.84
SNL	3038	LFP	1.10	18650	25	0.50	1.00

Cathode chemistry is specified for each battery recorded. Their abbreviations correspond to:

1. NCA: Lithium nickel cobalt aluminum oxide, used to enhance battery effectiveness and efficiency, enabling high energy density batteries.
2. LCO: Lithium cobalt oxide, is one of the first used in Lithium-ion batteries and one the most common.
3. NMC: Lithium Nickel Cobalt Manganese Oxide, strong overall performance and the most used in automotive applications.
4. LFP: Lithium Iron Phosphate, enviromentally safer option with low cost and good performance.

While chemistry is a battery characteristic to be taken into consideration when analyzing the prediction of each time series, given that the proposed model is of data-driven nature (and not chemical/electrical modeling), in this research work the most important values in Table 1 along with the number of cycles are the Charging C-Rate and Discharging C-Rate, which represent the time steps necessary to repeat each recorded cycle.

Table 2 Database segmenting

Segment	CALCE	OX	UL-PUR	SNL	HNEI
Training	1210	1200	2389	1500	3256
Validation	403	400	796	500	1085
Testing	403	400	797	500	1085
Total	2016	2000	3982	2500	5426

4.2 Preprocessing

Data normalization was applied to the input features. All of them are transformed into the range [0,1] according to eq. 11, where Z^{Norm} is the normalized series and the maximum and minimum values of the series are represented by $min(Z_t)$ and $max(Z_t)$ respectively.

$$Z_t^{Norm} = \frac{Z_t - min(Z_t)}{max(Z_t) - min(Z_t)} \tag{11}$$

Each time series was split into three subsets: training (60%), validation (20%), and testing (20%). This is specified for all databases in Table 2. The training set is destined for linear and non-linear models training processes, the validation set is used for best parameters, non-linear model and combination function of the model, and testing set is used for evaluation stage.

4.3 Parameter selection

ARIMA has three parameters to be adjusted, where p represent the autorregressive order, d is the number of differences in the series in order to make it stationary, and q represents the order of the moving average. The ARIMA model was generated using ”forecast” package in R language [29], through the auto.arima function which searches automatically the best parameter according to either AIC (Akaike Information Criterion), AICc (second order AIC) or BIC (Bayesian Information Criterion).

For the rest of the models (non-linear forecasting and combination), parameter selection process was based on a grid search over a set of possible configurations as described in Table N. For the SVR, hyperparameters were tuned under the definitions:

1. ε: The sparsity can be induced through the employment of a ε-insensitive loss function which creates a ε-tube.
2. C represents the regularization factor, which will perform the trade-off between complexity and generalization capacity of the model.
3. Kernels perform nonlinear mappings of the datasets into higher dimensional space.
4. γ defines the influence of each training example.

Next, for MLP:

1. Hidden layer sizes as the amount of neurons in each given hidden layer.
2. Activation function for the hidden layers.

3. Solver for weight optimization.
4. Learning rate for weight update.
5. α representing the strength of the L2 regularization term.

And, lastly, for DTR:

1. Splitter as the strategy selected to split the tree in each node.
2. Maximum depth of the tree.
3. Minimum amount of samples at each leaf node.
4. Minimum weighted fraction of the sum total of weights at each leaf node.
5. Maximum number of features to look for when splitting.
6. Maximum leaf nodes to grow the tree.

Table 3 Parameters search space per model

Model	Search space

MLP	Hidden layer sizes: {(50,50,50), (50,100,50), (100,100)}
	Activation: {relu, tanh}
	Solver: {adam, sgd}
	Learning rate: {constant, adaptive}
	Alpha: {0.001, 0.01, 0.05}

DTR	Splitter:{best, random}
	Max depth: {1,3,5,7,9,11,12}
	Min samples_leaf: {1,2,3,4,5,6,7,8,9,10}
	Min weight_fraction_leaf: {0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9}
	Max feature: {auto, log2, sqrt,None}
	Max leaf nodes: {None,10,20,30,40,50,60,70,80,90}

SVR	Kernel: {rbf, sigmoid}
	C: {0.1, 1, 10, 100, 1000, 10000}
	ε: {0.1, 0.01, 0.001}
	γ: {1, 0.1, 0.01, 0.001}

ARIMA	p: {1, 2, 3, 4, 5}
	d: {1, 2}
	q: {1, 2, 3, 4, 5}

4.3.1 Evaluation Metrics

All methods were evaluated using mean squared error (MSE) and mean absolute error (MAE), both described in equations 12 and 13 respectively. The MSE performs a quadratic evaluation of errors which applies heavier penalties to higher forecasting errors, it is a scale dependent metric, thus its not advisable to perform comparisons over multiple datasets. The MAE metric calculates the average error as output, using the same scale as the time series being evaluated.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Z_{t_i} - \hat{Z}_{t_i})^2 \tag{12}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Z_{t_i} - \hat{Z}_{t_i}| \tag{13}$$

[30] These measures, besides allowing to make some conclusions on the model performance, also allow to conduct Friedman and Nemenyi tests as a mean to determine whether or not there is a statistically significant difference in performance of all evaluated models across all datasets. This would bring conclusive evidence to backup the idea of the proposed model performing better than other models.

5 Results

Following the grid search process mentioned in section 4 for each model in all databases, the experiments were conducted on the 5 aforementioned data sets. In the proposed model, different models and architectures may be selected for each dataset, as presented in table 4. In CALCE data set a hybrid system based on a summation between ARIMA and DTR models was selected (ARIMA+DTR) whereas in the other datasets nonlinear combinations were the ones with the lowest validation errors.

Table 4 Hybrid model selected per database.

Database	**Selected model**
CALCE	ARIMA+DTR
OX	MLP(ARIMA,SVR)
UL-PUR	SVR(ARIMA,MLP)
SNL	DTR(ARIMA,MLP)
HNEI	DTR(ARIMA,SVR)

Table 5 Results of SoC forecasting based on the MSE metric.

Model	**CALCE**	**OX**	**UL-PUR**	**SNL**	**HNEI**
ARIMA	0.002676	0.008333	0.102552	0.014669	0.091894
MLP	0.003618	0.007771	0.088216	0.013377	0.082900
SVR	0.002638	0.009180	0.455908	0.013576	0.104277
DTR	0.004723	0.009212	0.121527	0.013881	0.089333
ARIMA+MLP	0.002581	0.007838	0.091630	0.012553	0.088479
ARIMA+SVR	0.002630	0.007946	0.142413	0.009227	0.101341
SVR(ARIMA, SVR)	0.002574	0.007291	**0.081474**	0.013638	0.095487
Proposed model	**0.002535**	**0.007207**	0.085143	**0.006356**	**0.056319**

For evaluation and comparison purposes, seven relevant methods found in related literature were implemented using the five datasets discussed in subsection 4.1. Mentioned models were: ARIMA, MLP, SVR, DTR, ARIMA+MLP, ARIMA+SVR, SVR(ARIMA, SVR). Model performance was evaluated first through MSE and MAE values presented in Tables 5 and 6, respectively.

In Table 5 for MSE, the worst performances across datasets were not all attributed to a unique model. In some cases DTR delivered the highest errors (as in CALCE), in others it was ARIMA (as in SNL), and in others it was SVR (as in UL-PUR). This points to the fact that datasets have different

Table 6 Results of SoC forecasting based on the MAE metric.

Model	**CALCE**	**OX**	**UL-PUR**	**SNL**	**HNEI**
ARIMA	0.025707	0.032881	0.242681	0.100858	0.212315
MLP	0.032743	0.026559	0.185550	0.054275	0.209012
SVR	0.020768	0.019573	0.423436	0.051230	0.225515
DTR	0.040050	0.041265	0.272541	0.0564085	0.209250
ARIMA+MLP	0.024822	0.019712	0.193115	0.061284	0.209460
ARIMA+SVR	0.023663	0.030410	0.254781	0.048978	0.220599
SVR(ARIMA, SVR)	0.023365	0.037330	**0.164220**	0.060612	0.214669
Proposed model	**0.020488**	**0.019461**	0.172206	**0.047297**	**0.199115**

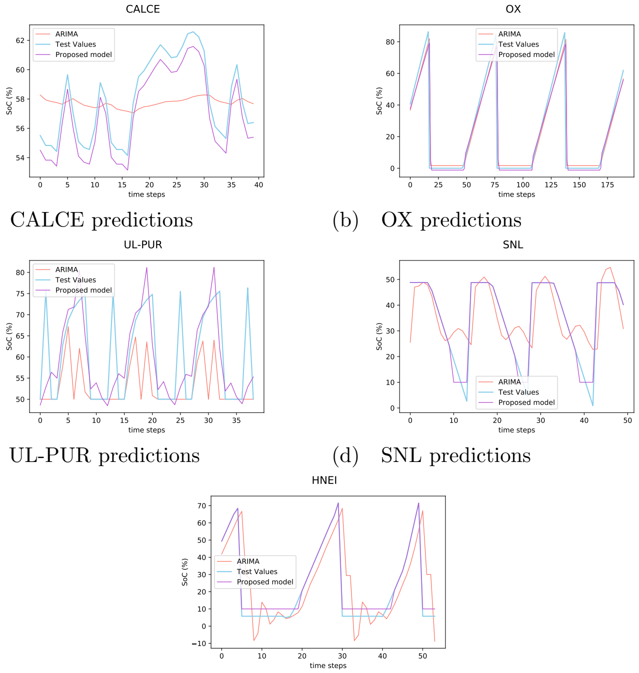
Table 7 Performance gain per model with ARIMA as reference.

Model	**CALCE**	**OX**	**UL-PUR**	**SNL**	**HNEI**
ARIMA	0.00	0.00	0.00	0.00	0.00
MLP	-35.20	6.74	13.98	8.81	9.79
SVR	1.42	-10.16	-344.56	7.45	-13.48
DTR	-76.49	-10.55	-18.50	5.37	2.79
ARIMA+MLP	3.55	5.94	10.65	14.42	3.72
ARIMA+SVR	1.72	4.64	-38.87	37.10	-10.28
SVR(ARIMA,MLP)	3.81	12.50	**20.55**	7.03	-3.91
Proposed model	**5.27**	**13.51**	16.98	**56.67**	**38.71**

characteristics that are not always perfectly predicted by a specific type of model, in some of them even ARIMA might perform better than a multi-layer perceptron (the series could have a strong influence of linear patterns and not so much non-linear ones). This only adds to the idea of combination and model selections allowing to improve overall performance. Thus, in CALCE dataset, the DTR model achieved lowest performance, however, when used in a hybrid way (ARIMA+DTR), the proposed model was able to outperform both methods when used individually.

MSE and MAE results also support the statement of the proposed model being better than the ones found in literature for datasets CALCE, OX, SNL, and HNEI by achieving the lowest error values in both metrics against simple and complex models. This is best represented in Table7, where performance gain reached 5.27%, 13.51%, 56.67%, and 38.71% respectively. For UL-PUR this was not the case as SVR(ARIMA, SVR) performed better (20.55% performance gain), leaving the proposed model to the second place in performance (16.98% gain). Still, this does not discard the proposed model as the best option among all options compared as it was confirmed by Nemenyi test later on.

In error analysis, the proposed model showed evidence of being probably the best method to predict the datasets. However, it is important to determine if the values obtain allow to make the conclusion that the model is statistically better than the others. To achieve such certainty, the post-hoc Nemenyi test [31] was implemented to reject the hypothesis of model performances being similar. This allows to examine the ranking of several methods over several datasets determining a minimal critical distance at which any performance would be considered to be different than the other.



(a) CALCE predictions

(b) OX predictions

(c) UL-PUR predictions

(d) SNL predictions

(e) HNEI predictions

Fig. 3 Predictions per base.

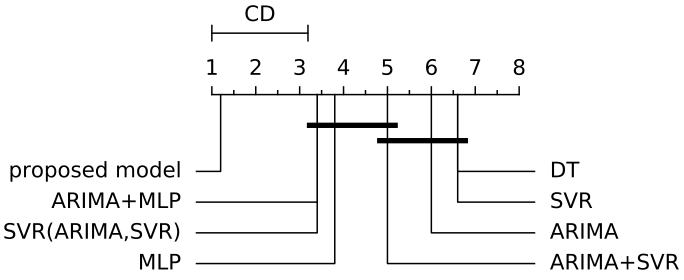


Fig. 4 Nemenyi test critical distance representation.

Using error values, model average ranks were approximated corresponding the highest one (rank 1.2) to the proposed model, and the lowest one (rank 6.6) to DTR. All of the others are placed in between these values, as represented in 4. The test was conducted with a significance level (α) of 0.05, meaning that the probability of the test failing on giving the correct inference is 95%. Also, calculated critical distance for available models and datasets was 2.189250 meaning that if two models find themselves separated by a greater distance, they would be considered to be different. This is specially important for the proposed model case, as it is important to discover if performance observed in error metrics allows to make favorable conclusions. The result of Nemenyi test is represented in Figure 4, with models ordered in average ranks and bold horizontal lines represent models that find themselves within a smaller range than the critical distance. ARIMA, DTR, and SVR are placed in the lowest ranks, as it was to be expected by observing their error values. MLP, ARIMA+MLP, and SVR(ARIMA, SVR) are found in the middle range being statistically similar but different from the proposed model, as there is no horizontal line joining them. This confirms the idea of the proposed model being the best performing one in the dataset group as it is the one with highest score and all of the other models are ranked farther than the critical distance from it.

6 Conclusions

In this study, a hybrid system with model selection in the residual forecasting stage and combination selection is presented in order to approximate future values of SoC in Lithium-ion batteries. The proposed model is formed by an ARIMA model and a nonlinear predictor, which is selected from a pool of nonlinear models according to its performance on validation set. It also selects the best combination function over a pool of options to combine the forecasts of the linear and non-linear models. The experiments were conducted on five univariate time series containing Lithium-ion batteries records were computed to predict their State-of-Charge values.

The results of the experiments highlight that the models with lowest performance were ARIMA, SVR, and DTR. This is supported by error values and Nemenyi test ranks. However, none of the three models mentioned was the worst performer on all datasets. This suggest the variety of linear and nonlinear components in all time series, which would not allow for a single model to predict efficiently all of them.

The ARIMA+SVR and MLP models performed in the middle ranks for all datasets, implying that neural network-based and hybrid models would ensure performance improvement. ARIMA+MLP and SVR(ARIMA, SVR) also help to support this idea and add to the conclusion that combination functions other than aggregation can help predictions to adapt best to a specific time series characteristics and, therefore, achieve the best predictions.

The proposed model achieved the lowest error values for both MSE and MAE in four of the five datasets used. In the case in which it did not get the lowest value, it was only displaced to second best rank by SVR(ARIMA,SVR) model. To determine if error values were statistically significant enough to conclude that the proposed model performs better than the other seven methods, a Nemenyi test was conducted. The results of such process demonstrated that the proposed model can be declared the best performing model amongst all as the rank space between them is greater than the critical distance given by the analysis (2.189250). This affirmation is made with 95% confidence level.

The accuracy gain of the proposed model is due to the ability it has to select the best non-linear model to be combined with the linear model and the combination function to do so. This allows to reduce greatly the chances of the model designer choosing the wrong options in either one.

This study is limited to static selection of non-linear model and combination function, as both are picked over performance in validation set and no longer updated over the testing stage. Future work may consider exploration of dynamic selection analyzing specific data features over time.

7 Declarations

Ethical Approval. Not applicable.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Authors' contributions. All authors contributed to the study conception and design. Data collection and experiments were performed by Marie Chantelle Cruz Medina. Analysis and the draft of the manuscript was written by Marie Chantelle Cruz Medina and João Fausto Lorenzato de Oliveira. All authors read and approved the final manuscript.

Funding. This research work was supported by the Foundation for Science and Technology Support from Pernambuco (FACEPE) - Process No. APQ-1252-1.03/21, and the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Financing Code 001 for the financial support.

Availability of data and materials. The datasets generated and analysed during the current study are available in the "BatteryArchive.org" repository, https://www.batteryarchive.org/

References

[1] Etacheri, V., Marom, R., Elazari, R., Salitra, G., Aurbach, D.: Challenges in the development of advanced li-ion batteries: a review. Energy & Environmental Science **4**(9), 3243–3262 (2011)

[2] Wang, Q., Wang, Z., Zhang, L., Liu, P., Zhang, Z.: A novel consistency evaluation method for series-connected battery systems based on real-world operation data. IEEE Transactions on Transportation Electrification **7**(2), 437–451 (2020)

[3] Sathyamurthy, R., Kabeel, A., Chamkha, A., Karthick, A., Manokar, A.M., Sumithra, M.: Experimental investigation on cooling the photovoltaic panel using hybrid nanofluids. Applied Nanoscience **11**(2), 363–374 (2021)

[4] Eddine, M.D., Shen, Y.: A deep learning based approach for predicting the demand of electric vehicle charge. The Journal of Supercomputing, 1–24 (2022)

[5] Zhang, X.-F., Zhao, Y., Patel, Y., Zhang, T., Liu, W.-M., Chen, M., Offer, G.J., Yan, Y.: Potentiometric measurement of entropy change for lithium batteries. Physical Chemistry Chemical Physics **19**(15), 9833–9842 (2017)

[6] Xiong, R., Cao, J., Yu, Q., He, H., Sun, F.: Critical review on the battery state of charge estimation methods for electric vehicles. Ieee Access **6**, 1832–1843 (2017)

[7] Wu, Z., Ling, R., Tang, R.: Dynamic battery equalization with energy and time efficiency for electric vehicles. Energy **141**, 937–948 (2017)

[8] Liu, K., Shang, Y., Ouyang, Q., Widanage, W.D.: A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery. IEEE Transactions on Industrial Electronics **68**(4), 3170–3180 (2020)

[9] Khalid, A., Sarwat, A.I.: Unified univariate-neural network models for lithium-ion battery state-of-charge forecasting using minimized akaike information criterion algorithm. Ieee Access **9**, 39154–39170 (2021)

[10] Taskaya-Temizel, T., Casey, M.C.: A comparative study of autoregressive neural network hybrids. Neural Networks **18**(5), 781–789 (2005)

[11] Amini, M.H., Kargarian, A., Karabasoglu, O.: Arima-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. Electric Power Systems Research **140**, 378–390 (2016)

[12] Makridakis, S., Hibon, M.: Arma models and the box–jenkins methodology. Journal of forecasting **16**(3), 147–163 (1997)

[13] Zhang, G.P.: Time series forecasting using a hybrid arima and neural network model. Neurocomputing **50**, 159–175 (2003)

[14] de Mattos Neto, P.S.G., Cavalcanti, G.D.C., Madeiro, F.: Nonlinear combination method of forecasters applied to pm time series. Pattern Recognition Letters **95**, 65–72 (2017)

[15] de O. Santos Júnior, D.S., de Oliveira, J.F.L., de Mattos Neto, P.S.G.: An intelligent hybridization of arima with machine learning models for time series forecasting. Knowledge-Based Systems **175**, 72–86 (2019)

[16] Medina, M.C.C., de Oliveira, J.F.L.: A hybrid system for lithium-ion

battery state-of-charge univariate forecasting. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 991–995 (2021). IEEE

[17] Panigrahi, S., Behera, H.S.: A hybrid ets–ann model for time series forecasting. Engineering Applications of Artificial Intelligence **66**, 49–59 (2017)

[18] Pai, P.-F., Lin, C.-S.: A hybrid arima and support vector machines model in stock price forecasting. Omega **33**(6), 497–505 (2005)

[19] de Holanda, R.B.B., de Oliveira, J.F.L.: Swarm based ensembles for time series residual forecasting. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 595–602 (2020)

[20] de Oliveira, J.F., Silva, E.G., de Mattos Neto, P.S.: A hybrid system based on dynamic selection for time series forecasting. IEEE Transactions on Neural Networks and Learning Systems (2021)

[21] de Curso, T.d.C.: Uma abordagem híbrida para a predição de séries temporais. PhD thesis, Universidade Federal de Pernambuco (2016)

[22] Mohammadi, F.: Lithium-ion battery state-of-charge estimation based on an improved coulomb-counting algorithm and uncertainty evaluation. Journal of Energy Storage **48**, 104061 (2022)

[23] Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for r. Journal of statistical software **27**, 1–22 (2008)

[24] Adam, S.P., Alexandropoulos, S.-A.N., Pardalos, P.M., Vrahatis, M.N.: No free lunch theorem: A review. Approximation and optimization, 57–82 (2019)

[25] Battery Archive. https://www.batteryarchive.org

[26] Juarez-Robles, D., Jeevarajan, J.A., Mukherjee, P.P.: Degradation-safety analytics in lithium-ion cells: Part i. aging under charge/discharge cycling. Journal of The Electrochemical Society **167**(16), 160510 (2020)

[27] Devie, A., Baure, G., Dubarry, M.: Intrinsic variability in the degradation of a batch of commercial 18650 lithium-ion cells. Energies **11**(5), 1031 (2018)

[28] Preger, Y., Barkholtz, H.M., Fresquez, A., Campbell, D.L., Juba, B.W., Romàn-Kustas, J., Ferreira, S.R., Chalamala, B.: Degradation of commercial lithium-ion cells as a function of chemistry and cycling conditions. Journal of The Electrochemical Society **167**(12), 120532 (2020)

[29] Hyndman, R.J., Akram, M.: Time series data library. Available from Internet: http://robjhyndman. com/TSDL (2010)

[30] Iea: Tracking Energy Storage 2020 – analysis (2020). https://www.iea.org/reports/tracking-energy-storage-2020

[31] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research **7**, 1–30 (2006)