

Article

State of Charge Estimation of Lithium Battery Based on Integrated Kalman Filter Framework and Machine Learning Algorithm

Hongyuan Yuan ^{1,2,*} , Jingnan Liu ³, Yu Zhou ² and Hailong Pei ^{1,*} ¹ School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China² China Energy Construction Group Guangdong Electric Power Design and Research Institute Co., Ltd., Guangzhou 510700, China³ School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China

* Correspondence: g8159@gedi.com.cn (H.Y.); auhlpei@scut.edu.cn (H.P.)

Abstract: Research on batteries' State of Charge (SOC) estimation for equivalent circuit models based on the Kalman Filter (KF) framework and machine learning algorithms remains relatively limited. Most studies are focused on a few machine learning algorithms and do not present comprehensive analysis and comparison. Furthermore, most of them focus on obtaining the state space parameters of the Kalman filter frame algorithm models using machine learning algorithms and then substituting the state space parameters into the Kalman filter frame algorithm to estimate the SOC. Such algorithms are highly coupled, and present high complexity and low practicability. This study aims to integrate machine learning with the Kalman filter frame algorithm, and to estimate the final SOC by using different combinations of the input, output, and intermediate variable values of five Kalman filter frame algorithms as the input of the machine learning algorithms of six main streams. These are: linear regression, support vector Regression, XGBoost, AdaBoost, random forest, and LSTM; the algorithm coupling is lower for two-way parameter adjustment and is not applied between the machine learning and Kalman filtering framework algorithms. The results demonstrate that the integrated learning algorithm significantly improves the estimation accuracy when compared to the pure Kalman filter framework or the machine learning algorithms. Among the various integrated algorithms, the random forest and Kalman filter framework presents the highest estimation accuracy along with good real-time performance. Therefore, it can be implemented in various engineering applications.

Keywords: Kalman Filter; random forest (RF); XGBoost; AdaBoost; support vector regression (SVR); long short-term memory (LSTM)



Citation: Yuan, H.; Liu, J.; Zhou, Y.; Pei, H. State of Charge Estimation of Lithium Battery Based on Integrated Kalman Filter Framework and Machine Learning Algorithm. *Energies* **2023**, *16*, 2155. <https://doi.org/10.3390/en16052155>

Received: 7 February 2023

Revised: 21 February 2023

Accepted: 21 February 2023

Published: 23 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Owing to the rapid development of the modern energy industry, energy storage batteries have been widely implemented in various fields, such as electric vehicles, energy storage power stations, data centers, communication base stations, and others. Lithium batteries have become the primary choice for energy storage batteries due to their high energy density and long cycle life. The efficient and safe management of lithium batteries has become crucial for engineering applications. Adequate battery management can help in achieving balanced charge and discharge for Li-ion batteries and can reduce the risk of excessive battery capacity consumption, which can lead to fires or explosions. Real-time and accurate estimation of a battery's state of charge (SOC) is essential for battery management. However, the SOC of a lithium battery corresponds to its complex internal physical and chemical reactions, along with the external working environment. Therefore, it has highly complex nonlinear time-varying characteristics and is difficult to measure directly, which considerably hinders engineering design and research.

Most of the previous studies conducted on lithium batteries' SOC estimation have been focused on the Kalman filter frame algorithms [1–3] and machine learning algorithms [4–9]. The Kalman filter frame algorithm dynamically estimates the SOC by using the circuit model of the battery to establish a state space model. It presents a moderate computational complexity with acceptable accuracy, but the complex relationship between the long time series data is underutilized. Machine learning algorithms can be used to determine the complex relationship between long time series data through the regression fitting of labeled time series data. However, the machine learning model is difficult to explain, the utilization efficiency of the data is low, and the accuracy is often insufficient for a small amount of data. Few studies have been conducted on SOC estimation that combine the machine learning and Kalman filter frame algorithms; they were primarily focused on a few machine learning algorithms and did not present a comprehensive analysis and comparison of different machine learning algorithms. Furthermore, most of them were focused on obtaining the state space parameters of the Kalman filter frame algorithm model using machine learning algorithms and then substituting the state space parameters into the Kalman filter frame algorithm to estimate the SOC [10–12]. Such algorithms are highly coupled and present high complexity and low practicability.

This study aims to simply and comprehensively integrate the machine learning and Kalman filter frame algorithms, and to estimate the final SOC by using different combinations of the input, output, and intermediate variable values of five Kalman filter frame algorithms, i.e., EKF, UKF, MIUKF, UKF-VFFRLS, MIUKF-VFFRLS [3], as the input of six mainstream machine learning algorithms, i.e., linear regression, support vector Regression [13], XGBoost [14,15], AdaBoost [16], Random Forest [17,18], and LSTM [19,20]; the algorithm coupling is lower for two-way parameter adjustment and is not applied between the machine learning and Kalman filtering framework algorithms. The results demonstrate that the integrated algorithm significantly improved the estimation accuracy when compared to the pure Kalman filter framework or machine learning algorithms. Among the integration algorithms, the XGBoost, AdaBoost, and Random Forest integration algorithms present high accuracy. The Random Forest integration algorithm presents the highest accuracy along with good real-time performance, and can be implemented in various engineering applications. In the cases which have strict real-time requirements, pure machine learning algorithms such as the XGBoost and Random Forest algorithms can be implemented due to their high speed and good accuracy.

2. General Design

To comprehensively analyze the various integrated algorithms and compare their performance with pure machine learning algorithms or pure Kalman filter frame algorithms, three aspects of grouping were considered in the design process. In the first group, the measured current and voltage are considered as input variables and the pure machine learning algorithms are considered as the control group with the target algorithm. In the second group, the feature variables obtained by the Kalman filter algorithm are considered as the input variables. The third group performs feature filtering on the feature variables obtained using the Kalman filter framework algorithms, from which, the three most important feature variables were selected as the input variables. The SOC estimation problem is essentially a prediction problem of time series; therefore, the time sliding window of variables is used for each grouping and the statistics of the time sliding window are used to transform and reorganize the input variables.

A large number of feature variables are obtained from the Kalman filter framework algorithms, and many of them exhibit linear correlations with each other. The conventional machine learning algorithms (not deep learning algorithms such as LSTM) present limited processing performance or are sensitive to linear redundancy. Consequently, the transformation and reorganization of the input variables in the third group are only performed in the LSTM algorithm.

The overall block diagram of the algorithm is depicted in Figure 1.

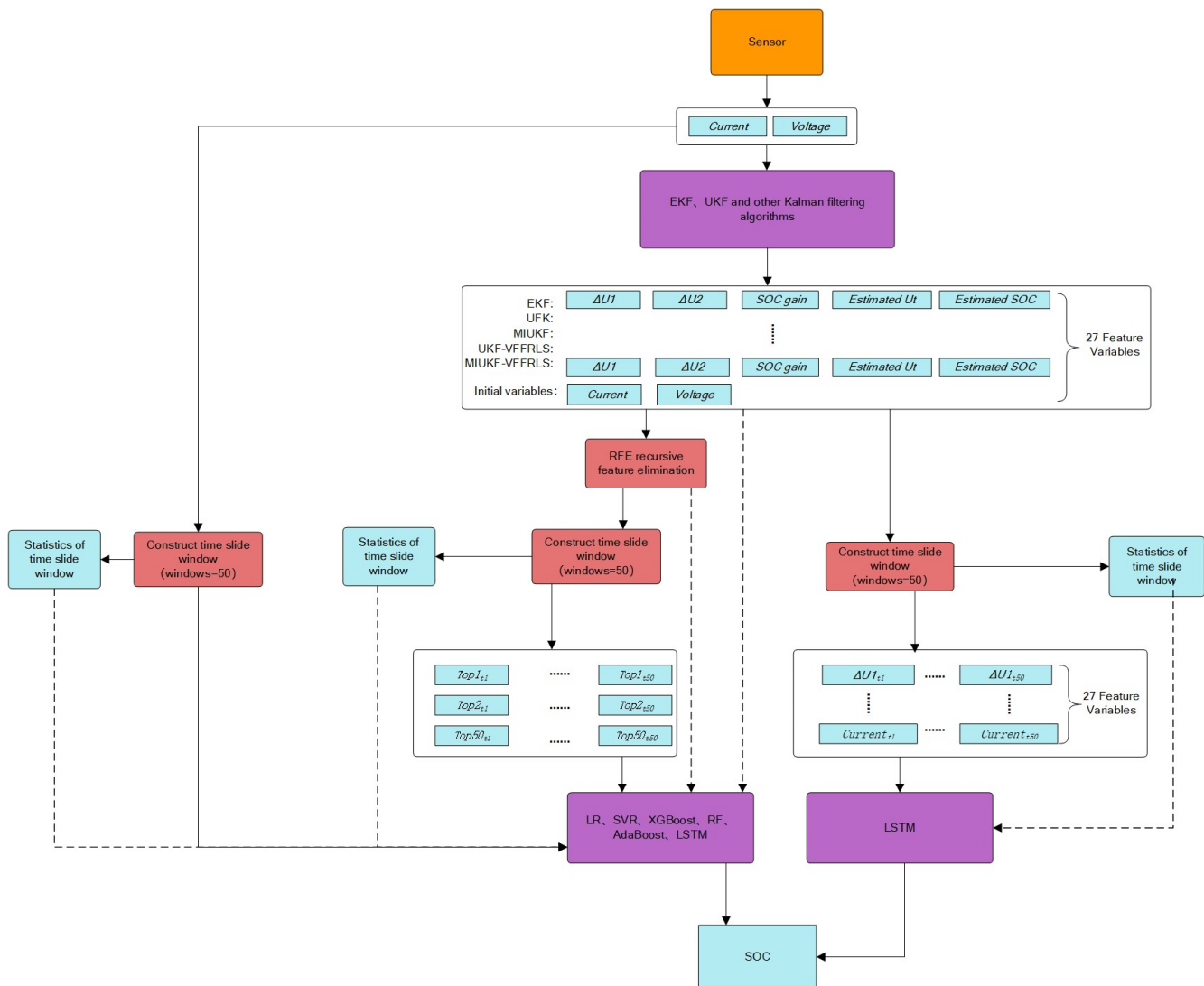


Figure 1. The overall block diagram of the algorithm.

3. Experimental Setting

3.1. Experimental Data Acquisition

The battery cell used in this paper is a 37 Ah/3.7 V ternary material Li-ion high power density cell manufactured by CATL (Contemporary Amperex Technology Co., Limited from Ningde City, Fujian Province, China), the mode name is S5E891 and the cut-off voltage range is 2.8~4.2 V. Considering that the voltage will drop sharply at the initial stage of every UDDS discharge cycle, in order to ensure battery safety, the protection voltage is set to 3.2 V. According to this voltage window, the measured battery capacity is 32 Ah, which is used as the reference capacity of SOC.

The experimental platform comprises the battery cells, power battery test system, and high/low temperature humidity and heat alternating test box. In this study, we used the test data from the U.S. Urban Road (UDDS) [21] under cycle conditions to verify the effectiveness of the algorithm; all the conditions were tested below a temperature of 25 °C. The battery was first fully charged for half an hour and then discharged for 13 UDDS cycles.

During this cycle, the SOC dropped from 100% to 1.2%, the voltage dropped from 4.18 V to 3.21 V (dropped from the cut-off voltage of 4.2 V in fact, the voltage at the initial stage of discharge dropped rapidly due to the existence of ohmic internal resistance, so the first sample data shows 4.18 V due to the sampling interval.), the interval for experimental data acquisition was set to 1 s, and a total of 20,000 data samples were collected.

3.2. Running Environment

The algorithms were run on a PC (Lenovo Savior R9000P, CPU by AMD Ryzen 5800H (16 cores)) with a Windows 10 operating system. Matlab R2019b was used as the compilation language for the Kalman filtering framework algorithms, and Python 3.8 was used as the compilation language for the machine learning algorithms. The libraries used by the machine learning algorithm are Pandas 1.4.2, Scikit-learn 1.0.2, Numpy 1.22.3, Pytorch 1.8.1, and XGBoost 1.4.2.

3.3. Design of Input Variables

The proposed design considers the different situations in which the measured variables, the variables obtained through the Kalman filter frame algorithms and the variables obtained after the filtering of the feature variables, are used as inputs for comparison. Furthermore, it considers the time series of the input variable and its statistics as the transformation input feature variables for comparative analysis.

3.3.1. Combination of Input Variables Based on Measured Variables

The measured current and voltage variables, the transformation of the sliding window with 50 time points, and the transformation of statistics of the sliding window with 50 time points are considered as the inputs of the six machine learning algorithms for experimental comparison.

(a) Variable Ontology

We used the measured voltage, U_t , and current, I_t , to estimate the SOC_t at time, t , as follows:

$$SOC_t = F(U_t, I_t) \quad (1)$$

where, F represents the function fitted by the algorithm.

(b) Transformation variables of sliding window with 50 time points

The SOC_t at time, t , is estimated by the voltage, $U_{t-49}, U_{t-48}, \dots, U_t$, and current, $I_{t-49}, I_{t-48}, \dots, I_t$, at time, t , and before 49 time points, as shown below:

$$SOC_t = F(U_{t-49}, U_{t-48}, \dots, U_t; I_{t-49}, I_{t-48}, \dots, I_t) \quad (2)$$

where, F represents the function fitted by the algorithm, each segment separated by a semicolon refers to a time series combination of a variable, and the input variables of 49 time points before time 0 are filled with the value of time, 0.

(c) Transformation variables of statistics of sliding window with 50 time points

The SOC_t at time t is estimated by using the statistics of the mean, standard deviation, and median of the voltage and current measured at time, t , and at the previous 49 time points, as shown below:

$$\begin{aligned} SOC_t = F(&Mean(U_{t-49}, U_{t-48}, \dots, U_t), Median(U_{t-49}, U_{t-48}, \dots, U_t), \\ &Std(U_{t-49}, U_{t-48}, \dots, U_t), Mean(I_{t-49}, I_{t-48}, \dots, I_t), \\ &Median(I_{t-49}, I_{t-48}, \dots, I_t), Std(I_{t-49}, I_{t-48}, \dots, I_t)) \end{aligned} \quad (3)$$

where, F represents the function of the algorithm fitting, $Mean$ represents the function of determining the mean value of the variable, $Median$ represents the function of determining the median of the variable, Std represents the function of determining the standard deviation of the variable, and the input variables of 49 time points before time, 0, are filled with the value of time, 0.

3.3.2. Combination of Input Variables Based on Kalman Framework Algorithm

The measured current, voltage, and statistics of 27 variables based on Kalman filter frame algorithm, the transformation of the sliding window with 50 time points, and the transformation of statistics of the sliding window with 50 time points, are considered as the inputs of the six machine learning algorithms for experimental comparison.

(a) Variable Ontology

The Kalman filter framework algorithm establishes the state equation based on the equivalent circuit model of the battery to estimate the battery terminal voltage and SOC. It dynamically corrects the state variable using the error of the measured terminal voltage. In this paper, we employ the common second-order equivalent circuit DP model, whose circuit structure is depicted in Figure 2.

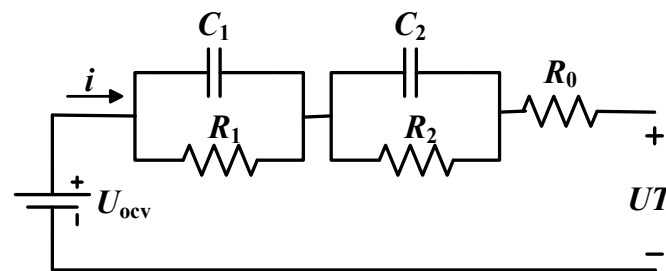


Figure 2. DP Equivalent circuit model.

It can be observed that the DP model comprises a voltage source, ohmic internal resistance, and RC network, where U_{ocv} represents the battery open circuit voltage, R_0 represents the battery ohmic internal resistance, and U_t represents the battery terminal voltage. The parallel network constructed by R_1 and C_1 is used to describe the long-term concentration polarization effect, the parallel network constructed by R_2 and C_2 is used to describe the short-term electrochemical polarization effect, and U_1 and U_2 represent the low order and high order polarization voltages of the second-order circuit model, respectively.

According to Kirchhoff's law, the output voltage of the DP model is given as follows:

$$\begin{cases} U_t = U_{ocv} - IR_0 - U_1 - U_2 \\ \dot{U}_1 = -\frac{1}{C_1 R_1} U_1 + \frac{1}{C_1} I \\ \dot{U}_2 = -\frac{1}{C_2 R_2} U_2 + \frac{1}{C_2} I \end{cases} \quad (4)$$

where, $\dot{U}_1$ and $\dot{U}_2$ represent the derivatives of U_1 and U_2 corresponding to time, respectively.

The general form of the state variables of the state equation of the above circuit model is $X = (U_1, U_2, SOC)^T$ that the superscript T refers to the transposition of the matrix.

The five algorithms based on the second-order model of Li-ion batteries, i.e., *EKF*, *UKF*, *MIUKF*, *UKF-VFFRLS* and *MIUKF-VFFRLS*, follow the framework of Kalman filtering. Their state variables follow the general form presented above, which includes five output variables, i.e., U_1 gain, ΔU_1 , U_2 gain, ΔU_2 , SOC gain, ΔSOC , estimated terminal voltage, U_t , and estimated SOC; these five variables are closely related to the final real SOC value. Therefore, a total of 25 variables, obtained from five variables in five algorithms, are selected as the input variables of the machine learning algorithm. The original measured current and voltage variables are the basic variables used to estimate the SOC, and are also included in the input variables. Therefore, a total of 27 variables are used to estimate the SOC, as shown below:

$$\begin{aligned}
SOC_t = F(&U_t, I_t, \Delta U1_t^{EKF}, \Delta U2_t^{EKF}, \Delta SOC_t^{EKF}, UT_t^{EKF}, SOC_t^{EKF}, \\
&\Delta U1_t^{UKF}, \Delta U2_t^{UKF}, \Delta SOC_t^{UKF}, UT_t^{UKF}, SOC_t^{UKF}, \\
&\Delta U1_t^{MIUKF}, \Delta U2_t^{MIUKF}, \Delta SOC_t^{MIUKF}, UT_t^{MIUKF}, SOC_t^{MIUKF}, \\
&\Delta U1_t^{UKF-VFFRLS}, \Delta U2_t^{UKF-VFFRLS}, \Delta SOC_t^{UKF-VFFRLS}, UT_t^{UKF-VFFRLS}, SOC_t^{UKF-VFFRLS} \\
&\Delta U1_t^{MIUKF-VFFRLS}, \Delta U2_t^{MIUKF-VFFRLS}, \Delta SOC_t^{MIUKF-VFFRLS}, UT_t^{MIUKF-VFFRLS}, SOC_t^{MIUKF-VFFRLS})
\end{aligned} \quad (5)$$

where, F represents the function fitted by the algorithm, and the superscript refers to the corresponding Kalman filter framework algorithm.

(b) Transformation variables of sliding window with 50 time points

The SOC_t is estimated by the 27 input variables at time, t , and before 49 time points. The input variables of the 49 time points before time, 0, are filled with the value of time 0. This formula is omitted due to its complexity.

(c) Transformation variables of statistics of sliding window with 50 time points

The SOC_t at time, t , is estimated by using the statistics of the mean, standard deviation, and median of 27 input variables at time, t , and the previous 49 time points. The input variables of 49 time points before time, 0, are filled with the value of time, 0. This formula is also omitted due to its complexity.

3.3.3. Combination of Input Variables after Elimination of Redundant Variables Based on Kalman Frame Algorithm Variables

The transformation of the time sliding window increases the computational complexity of the tree model algorithms (Random Forest, XGBoost, and AdaBoost) due to the large number of algorithm variables based on the Kalman filtering framework, which considerably increases the computational time of the experiment. Furthermore, Linear Regression and SVR are sensitive to the redundant feature variables; therefore the redundant variables must be eliminated. Additionally, the LSTM algorithm performs a performance comparison between all the feature variables and the selected feature variables. Therefore, in this paper, we employed the recursive feature elimination method to eliminate the redundant variables based on the Kalman filter framework, which ultimately selects the three most important feature variables corresponding to the different algorithms, as the input.

(a) Variable ontology after PEF recursive feature elimination

We employed a top-down method, in which all the features were included at the beginning, and some were gradually discarded to analyze the results. The evaluation method is described below.

We divided the data set into five parts, each of which was disjoint, of the same size. We then successively selected one of these five parts as the validation set, and the remaining four parts were used as the training set. Therefore, five separate model training and validation actions were performed. The average of the five validation results was considered as the validation error of this model.

Lastly, the three most important features for each algorithm were selected, as shown in Table 1.

(b) Transformation variables of sliding window with 50 time points after PEF recursive feature elimination

The SOC_t is estimated by the three most important input variables at time, t , and before the 49 time points. The input variables of the 49 time points before time, 0, are filled with the value of time, 0. This formula is omitted due to its complexity.

(c) Transformation variables of statistics of sliding window with 50 time points after PEF recursive feature elimination

The SOC_t at time, t , is estimated by using the statistics of the mean, standard deviation, and median of the three most important input variables at time, t , and the previous 49 time points. The input variables of the 49 time points before time, 0, are filled with the value of time, 0. This formula is omitted due to its complexity.

Table 1. The three most important features of each algorithm.

Feature Sorting Algorithm	First Feature	Second Feature	Third Feature
XGBoost	SOC^{EKF}	$\Delta U1^{EKF}$	$SOC^{UKF-VFFRLS}$
AdaBoost	SOC^{EKF}	$\Delta U1^{UKF-VFFRLS}$	$SOC^{UKF-VFFRLS}$
RF	SOC^{EKF}	$\Delta U1^{UKF-VFFRLS}$	$SOC^{UKF-VFFRLS}$
LR	$SOC^{UKF-VFFRLS}$	SOC^{MIUKF}	$SOC^{MIUKF-VFFRLS}$
SVR	$SOC^{UKF-VFFRLS}$	SOC^{MIUKF}	$SOC^{MIUKF-VFFRLS}$
LSTM	SOC^{EKF}	$\Delta U1^{UKF-VFFRLS}$	$SOC^{UKF-VFFRLS}$

3.4. Parameter Design of the Algorithm

We analyzed a total of 20,000 data elements, of which 80% were randomly selected as the training set and 20% were selected as the test set, and random seed was set to 2022.

The settings for the hyperparameters of the algorithm were determined based on the basic principles of machine learning algorithm training to ensure the convergence of the algorithm and to avoid falling into the local optimum point as much as possible.

The Kalman filtering framework algorithms and parameters were the same as the existing paper [9].

4. Results and Discussion

Improving the estimation accuracy and satisfying the real-time production requirements are crucial factors in engineering applications and research. Therefore, the accuracy and time consumption of the machine learning algorithms must be comprehensively analyzed. In this section, the accuracy and time consumption of each algorithm were compared and analyzed based on the aforementioned experimental settings.

4.1. Analysis of Accuracy of the Algorithm

The accuracy of the algorithm is determined by the final output. Therefore, the accuracy of the integrated algorithm can be analyzed by analyzing the accuracy of the export algorithm, i.e., the six machine learning algorithms. The accuracy is generally determined by the RMSE and MAE values. RMSE is significantly affected by outliers when compared to MAE.

The RMSE and MAE of all the algorithms when selecting the different input variables are listed in Tables 2 and 3.

Table 2. Comparison of algorithms' RMSE when selecting different input variables.

Algorithms Inputs	XGBoost	AdaBoost	RF	LR	SVR	LSTM
Current, Voltage	1.3100%	1.3200%	1.3400%	2.2200%	2.2700%	1.5200%
Sliding windows with 50 time points for current and voltage	0.4300%	0.5120%	0.7850%	1.6400%	2.6900%	0.1790%
Statistics of sliding windows with 50 time points for current and voltage	0.3600%	0.2710%	0.4220%	1.6800%	2.3500%	1.7895%
TOP 3 important feature variables	0.0161%	0.0174%	0.0108%	0.2090%	0.2760%	0.0964%

Table 2. Cont.

Inputs \ Algorithms	XGBoost	AdaBoost	RF	LR	SVR	LSTM
Sliding windows with 50 time points for TOP 3 important feature variables	0.0180%	0.0182%	0.0127%	4.5100%	0.2680%	0.2310%
Statistics of Sliding windows with 50 time points for TOP 3 important feature variables	0.0145%	0.0156%	0.0094%	0.1850%	0.2330%	0.1475%
27 features	0.0135%	0.0166%	0.0101%	0.0251%	0.0967%	0.0898%
Sliding windows with 50 time points for 27 features	\	\	\	\	\	0.0861%
Statistics of sliding windows with 50 time points for 27 features	\	\	\	\	\	0.9160%

Table 3. Comparison of algorithms' MAE when selecting different input variables.

Inputs \ Algorithms	XGBoost	AdaBoost	RF	LR	SVR	LSTM
Current, Voltage	0.7730%	0.8040%	0.7570%	1.7800%	1.7400%	1.0400%
Sliding windows with 50 time points for current and voltage	0.2970%	0.2610%	0.4760%	1.3200%	2.1600%	0.1250%
Statistics of sliding windows with 50 time points for current and voltage	0.2100%	0.0901%	0.1610%	1.3800%	0.1820%	1.4667%
TOP 3 important feature variables	0.0091%	0.0095%	0.0051%	0.1490%	0.1570%	0.0682%
Sliding windows with 50 time points for TOP 3 important feature variables	0.0113%	0.0109%	0.0070%	0.3250%	0.1410%	0.1630%
Statistics of Sliding windows with 50 time points for TOP 3 important feature variables	0.0082%	0.0085%	0.0044%	0.1350%	0.1710%	0.0997%
27 features	0.0075%	0.0087%	0.0051%	0.0197%	0.0547%	0.0647%
Sliding windows with 50 time points for 27 features	\	\	\	\	\	0.0644%
Statistics of sliding windows with 50 time points for 27 features	\	\	\	\	\	0.0197%

The RMSE of Kalman filtering framework algorithms are shown in Table 4.

Table 4. The RMSE of Kalman filtering framework algorithms.

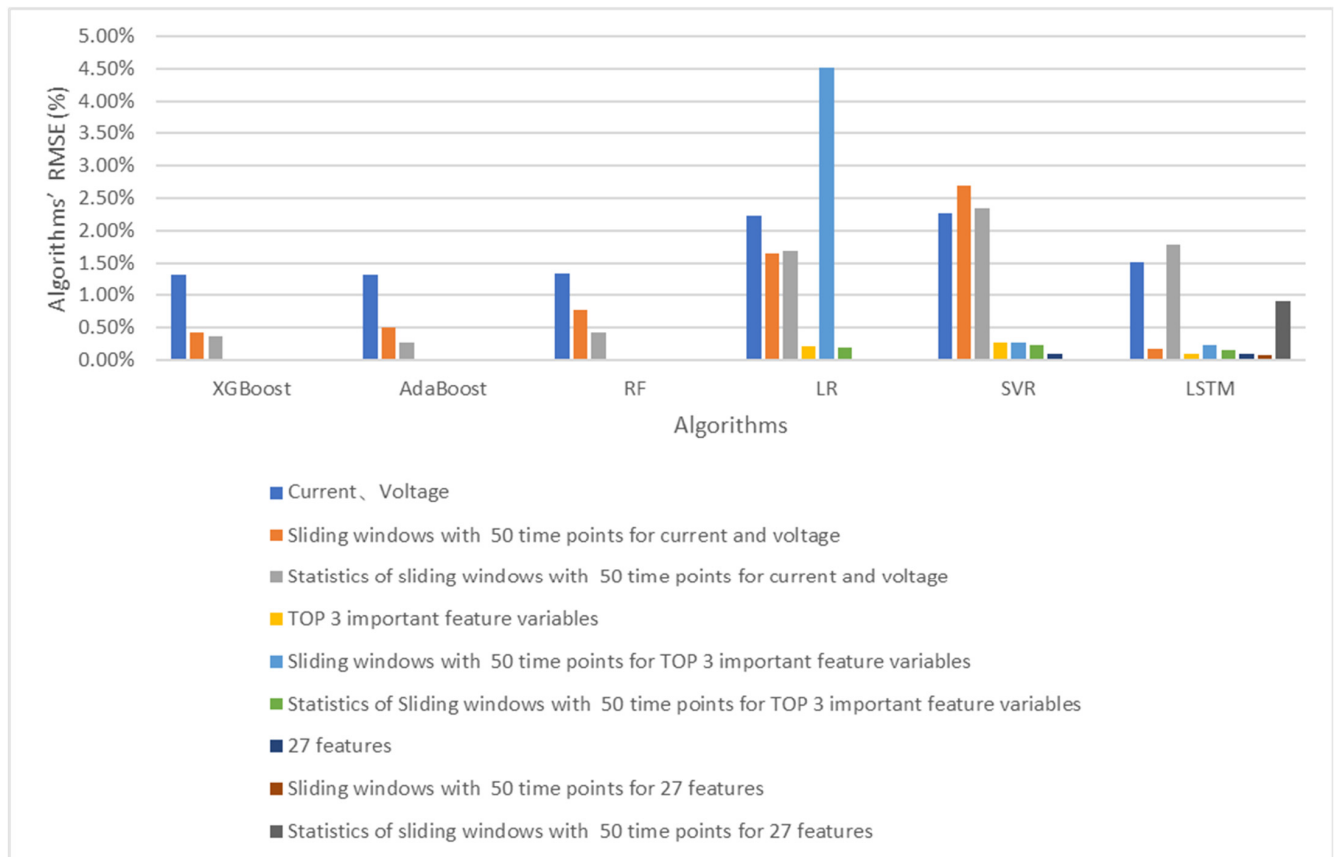
Inputs \ Algorithms	EKF	UKF	MIUKF	UKF-VFFRLS	MIUKF-VFFRLS
Current, Voltage	1.2891%	1.2434%	1.3307%	0.7754%	0.7020%

The MAE of Kalman filtering framework algorithms are shown in Table 5.

Table 5. The MAE of Kalman filtering framework algorithms.

Algorithms	EKF	UKF	MIUKF	UKF-VFFRLS	MIUKF-VFFRLS
Inputs					
Current, Voltage	1.1964%	1.0248%	0.7796%	0.2461%	0.2261%

The RMSE cluster histograms of each algorithm when selecting the different input variables are illustrated in Figures 3 and 4.

**Figure 3.** Comparison of the RMSE of algorithms when selecting different input variables (all).

The MAE cluster histograms of each algorithm when selecting different input variables are shown in Figures 5 and 6.

We can observe that the integrated algorithm exhibits a significant improvement in the estimation accuracy when compared to the pure Kalman filter framework or machine learning algorithm. When we used the pure Kalman filter framework algorithm, MIUKF-VFFRLS exhibits the highest accuracy, and the RMSE and MAE of the algorithm are 0.7020% and 0.2261%, respectively. When we use the pure machine learning algorithm, AdaBoost exhibits the highest accuracy for the “input current and voltage with 50 time sliding window statistics,” and the RMSE and MAE are 0.2710% and 0.0901%, respectively. The RMSE and MAE of the integrated algorithms are significantly lower than that of the pure Kalman filter framework and machine learning algorithm, except for the linear regression when “50 time sliding windows of the three most important features” is considered as the input and the LSTM algorithm when “statistics of 27 features with 50 time sliding windows” is considered as the input.

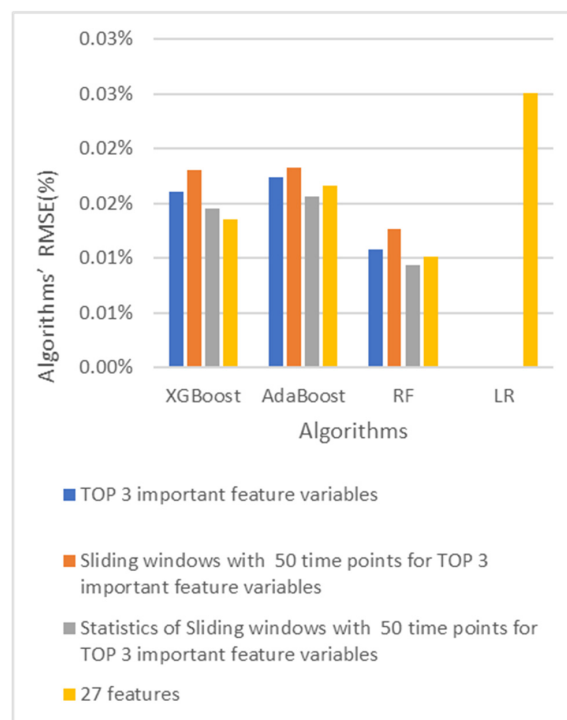


Figure 4. Comparison of the RMSE of algorithms when selecting different input variables (accuracy within 0.03%).

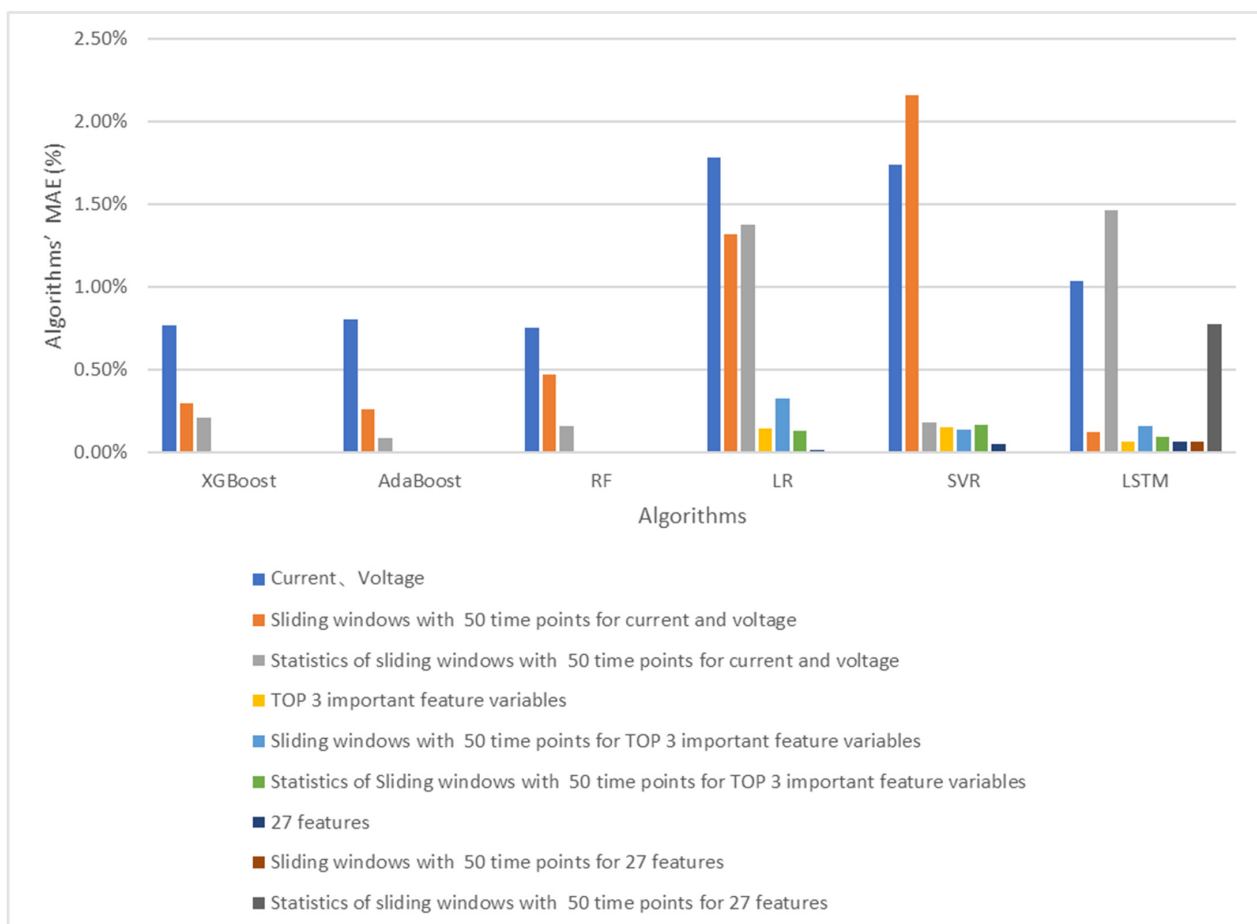


Figure 5. Comparison of algorithms' MAE when selecting different input variables (all).

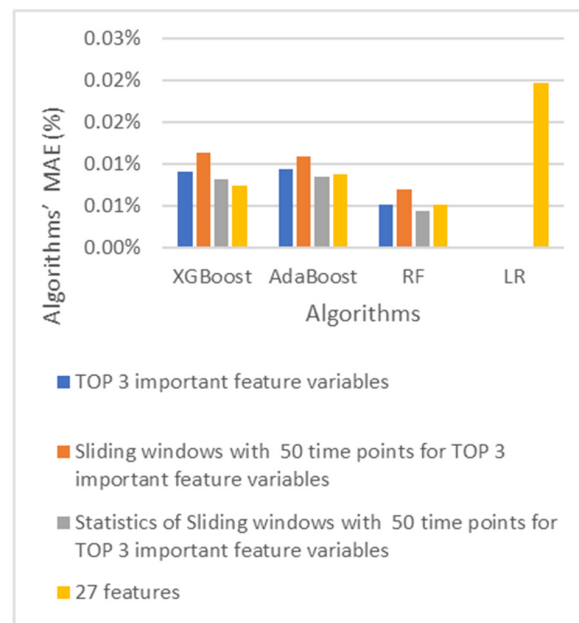


Figure 6. Comparison of algorithms' MAE when selecting different input variables (accuracy within 0.02%).

Furthermore, it can be observed from Figures 3–6 that among all the integrated algorithms, XGBoost, AdaBoost, and Random Forest exhibit higher accuracy. Among them, the Random Forest integrated algorithm exhibits the highest accuracy with the number of 0.0094% for RMSE and 0.0044% for MAE when “50 time sliding windows statistics of the three most important features” is used. Additionally, the Linear Regression algorithm with “27 features” considered as the input also exhibits high accuracy.

The deep learning algorithm, LSTM, exhibits a lower accuracy when compared to the tree model algorithm, except for the “current and voltage for 50 time sliding windows” input. The accuracy is not as good as that of the Linear Regression algorithm even when “27 features” is considered as the input. From the perspective of the size of the model parameter scale, the Linear Regression algorithm is less than the tree model algorithm, and the tree model algorithm is less than the deep learning algorithm. The accuracies of the models with low- and medium-scale parameters are more optimized, indicating that empirical evidence is crucial in selecting an appropriate algorithm for actual problems. Thus, it does not mean that when the algorithm model is larger, the accuracy is better. The tree model has good regression performance, which indicates that the prediction ability corresponding to its parameter scale is better adapted to the SOC estimation of battery in this working condition and similar cycle scale.

4.2. Analysis of Time Consumption of the Algorithm

The time consumption of the integrated algorithm can be obtained by summing up the total time consumption of the Kalman filter framework algorithm and the machine learning algorithm. Table 6 presents a comparison of the time consumption of each machine learning algorithm for different inputs.

Table 6. The time consumption comparison of each machine learning algorithm for different inputs (s).

Inputs \ Algorithm	XGBoost	AdaBoost	RF	LR	SVR	LSTM
Current, Voltage	0.0171	4.19	0.0991	0.0001	0.00015	1.16
Sliding windows with 50 time points for current and voltage	0.218	11.5	0.117	0.00156	0.00312	1.71

Table 6. *Cont.*

Inputs \ Algorithms	XGBoost	AdaBoost	RF	LR	SVR	LSTM
Statistics of sliding windows with 50 time points for current and voltage	0.189	4.234	0.28	0.001	0.001	0.5
TOP 3 important feature variables	0.0468	3.95	0.28	0.0000609	0.00016	1.08
Sliding windows with 50 time points for TOP 3 important feature variables	0.051	14.2	0.293	0.003	0.003	1.66
Statistics of Sliding windows with 50 time points for TOP 3 important feature variables	0.036	4.18	0.264	0.001	0.001	0.5
27 features	0.0355	5.37	0.271	0.000406	0.0005	1.12
Sliding windows with 50 time points for 27 features	\	\	\	\	\	1.65
Statistics of sliding windows with 50 time points for 27 features	\	\	\	\	\	0.5

The time consumption statistics of the Kalman filter framework algorithm are as show in Table 7.

Table 7. Time consumption of Kalman filter frame algorithm (s).

Algorithms \ Data	All Data	Test Set
EKF	0.4375	0.0875
UKF	1.5781	0.3156
MIUKF	1.9375	0.3875
UKF + VFFRLS	2.0469	0.4094
MIUKF + VFFRLS	3.8125	0.7625

The Kalman filter frame algorithms can be implemented simultaneously; therefore, when multiple Kalman filter frame algorithms are used, the longest time consumption is added to the time consumption of the machine learning algorithm to estimate the total time consumption. Table 8, Figures 7–9 present the total time consumption of the integrated algorithms for the test set data.

Table 8. The total time consumption of the integrated algorithms for test set data (s).

Inputs \ Algorithms	XGBoost	AdaBoost	RF	LR	SVR	LSTM
Current, Voltage	0.0171	4.19	0.0991	0.0001	0.00015	1.16
Sliding windows with 50 time points for current and voltage	0.218	11.5	0.117	0.00156	0.00312	1.71
Statistics of sliding windows with 50 time points for current and voltage	0.189	4.234	0.28	0.001	0.001	0.5
TOP 3 important feature variables	0.45618	4.35938	0.68938	0.7625609	0.76266	1.48938
Sliding windows with 50 time points for TOP 3 important feature variables	0.46038	14.60938	0.70238	0.7655	0.7655	2.06938
Statistics of Sliding windows with 50 time points for TOP 3 important feature variables	0.44538	4.58938	0.67338	0.7635	0.7635	0.90938

Table 8. Cont.

Inputs \ Algorithms	XGBoost	AdaBoost	RF	LR	SVR	LSTM
27 features	0.798	6.1325	1.0335	0.762906	0.763	1.8825
Sliding windows with 50 time points for 27 features	\	\	\	\	\	2.4125
Statistics of sliding windows with 50 time points for 27 features	\	\	\	\	\	1.2625

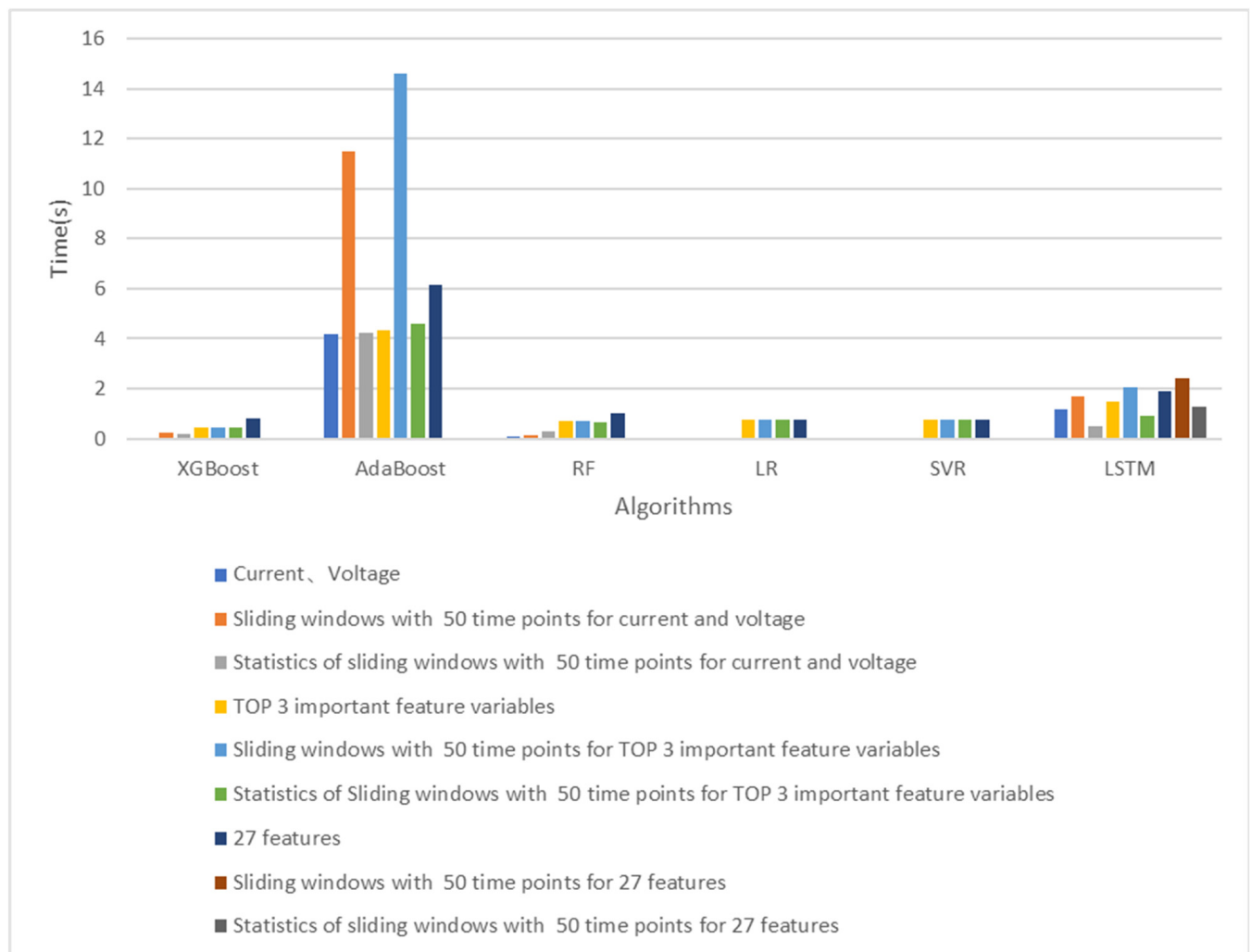


Figure 7. Total time consumption comparison of integrated algorithms for test set data (all).

It can be observed that XGBoost, Random Forest, Linear Regression, and SVR present low time consumption. The time consumption of the Kalman filter framework algorithm contributes the most to the total time consumption of the integrated algorithm. This time consumption can be considerably reduced by using a pure machine learning algorithm with a voltage and current input. Therefore, pure machine learning algorithms can be considered in situations with strict real-time requirements; the XGBoost and Random Forest algorithms are preferred in these situations due to their high speed and good accuracy.

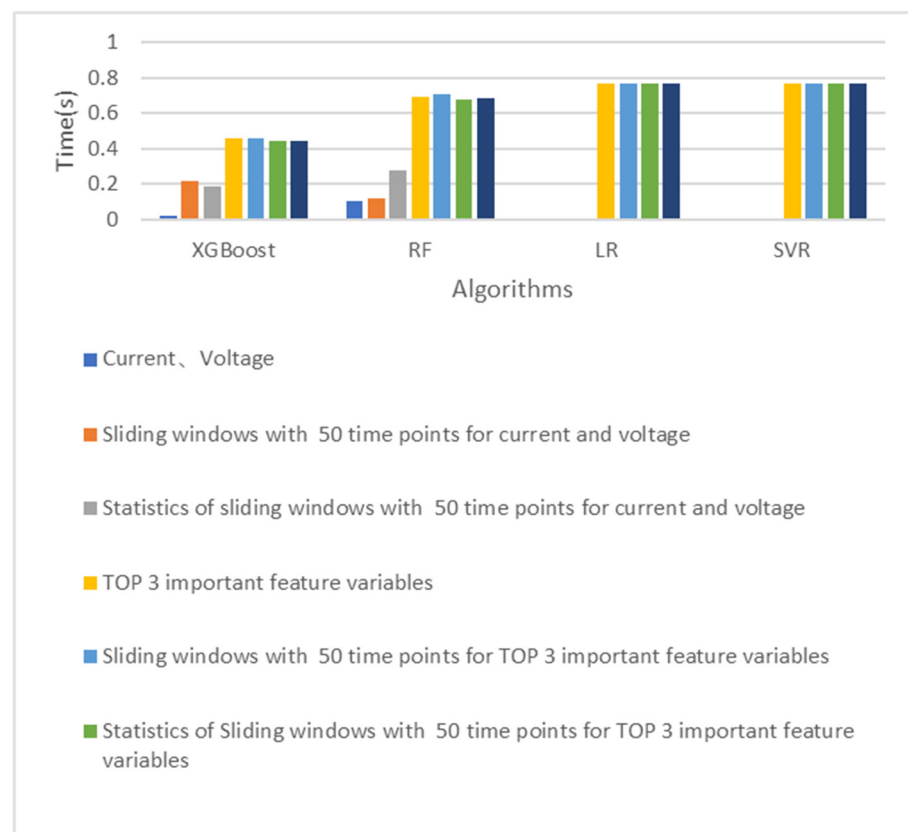


Figure 8. Total time consumption comparison of integrated algorithms for test set data (time within 0.8 s).

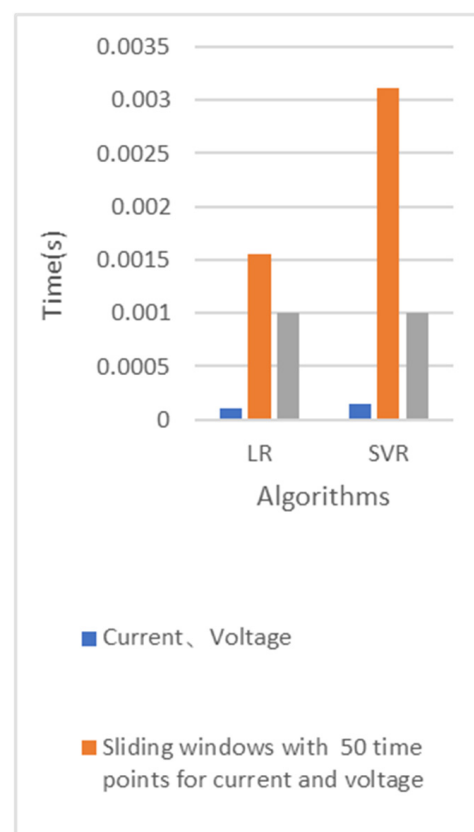


Figure 9. Total time consumption comparison of integrated algorithms (time within 0.003 s).

5. Conclusions

Accurate and fast SOC estimation is the basic requirement for the efficient operation and safe management of lithium batteries. The estimation of lithium batteries' SOC exhibits strong nonlinear characteristics due to the complex physical and chemical changes in lithium batteries. Improving the estimation accuracy and satisfying the requirements of real-time production is crucial for engineering applications and research. The existing studies are focused on the Kalman filter frame algorithms and machine learning algorithms, and few studies have been conducted on the integrated algorithms. Furthermore, most of these studies employ machine learning algorithms to estimate the parameters in the Kalman filter frame algorithm, and the estimated parameters are then substituted into the Kalman filter frame algorithm to estimate the final SOC. This results in highly coupled algorithms with complex design and low practicality. In this study, we try to integrate the machine learning and Kalman filter frame algorithms in an easier and more comprehensive way; the output variable of the Kalman filter algorithms are used as the inputs of the machine learning algorithms to estimate the final SOC. We integrate five mainstream Kalman filter frame algorithms and six machine learning algorithms, conduct time series and statistics transformation on the input variables, and perform a comparison of the various input combinations; the algorithm coupling is lower when a two-way parameter adjustment is not applied between the machine learning and Kalman filtering framework algorithms. The results demonstrate that the integrated algorithm significantly improved the estimation accuracy when compared to the pure Kalman filter framework or machine learning algorithms. Among the various types of integration algorithms, the XGBoost, AdaBoost, and Random Forest algorithms exhibit high accuracy. The Random Forest integration algorithm presents the highest estimation accuracy along with good real-time performance, and can thus be implemented in various engineering applications. Pure machine learning algorithms can be considered in situations with strict real-time requirements; the XGBoost and Random Forest algorithms are preferred in these situations due to their high speed and good accuracy.

This paper verifies the integrated algorithm of the collected data of a discharge cycle of lithium battery under the UDDS working condition and 25 °C experimental condition. Data with other working conditions, different temperatures and longer periods are not verified due to the limited time and energy available. If the experimental data is further expanded, algorithms with larger parameters, such as LSTM, may show better performance. Further research can be expanded in this direction.

Author Contributions: Conceptualization, H.Y.; methodology, H.Y.; software, J.L.; validation, H.Y. and Y.Z.; formal analysis, H.Y.; investigation, H.Y. and H.P.; resources, Y.Z.; data curation, H.Y.; writing—original draft preparation, H.Y.; writing—review and editing, H.Y.; visualization, J.L.; supervision, H.P.; project administration, H.P. All authors have read and agreed to the published version of the manuscript. H.Y. and J.L. contributed equally to this work.

Funding: This research received no external funding.

Data Availability Statement: Can be provided by email if asked.

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

References

1. Zhang, J.; Wang, Q.; Meng, F.; Shi, H.; Xi, Y. New Energy Vehicle Battery SOC Evaluation Method based on Robust Extended Kalman Filter. *J. Phys. Conf. Ser.* **2022**, *2196*, 012037. [\[CrossRef\]](#)
2. Hossain, M.; Haque, M.; Arif, M. Kalman filtering techniques for the online model parameters and state of charge estimation of the Li-ion batteries: A comparative analysis. *J. Energy Storage* **2022**, *51*, 104174. [\[CrossRef\]](#)
3. Yuan, H.; Han, Y.; Zhou, Y.; Chen, Z.; Du, J.; Pei, H. State of Charge Dual Estimation of a Li-ion Battery Based on Variable Forgetting Factor Recursive Least Square and Multi-Innovation Unscented Kalman Filter Algorithm. *Energies* **2022**, *15*, 1529. [\[CrossRef\]](#)
4. Cui, Z.; Wang, L.; Li, Q.; Wang, K. A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network. *Int. J. Energy Res.* **2021**, *46*, 5423–5440. [\[CrossRef\]](#)
5. Liu, X.; Dai, Y. Energy storage battery SOC estimate based on improved BP neural network. *J. Phys. Conf. Ser.* **2022**, *2187*, 012042. [\[CrossRef\]](#)

6. Hu, C.; Cheng, F.; Ma, L.; Li, B. State of Charge Estimation for Lithium-Ion Batteries Based on TCN-LSTM Neural Networks. *J. Electrochem. Soc.* **2022**, *169*, 030544. [\[CrossRef\]](#)
7. Meng, J.; Luo, G.; Ricco, M.; Swierczynski, M.; Stroe, D.-I.; Teodorescu, R. Overview of Lithium-Ion Battery Modeling Methods for State-of-Charge Estimation in Electrical Vehicles. *Appl. Sci.* **2018**, *8*, 659. [\[CrossRef\]](#)
8. Chandran, V.; Patil, C.; Karthick, A.; Ganeshaperumal, D.; Rahim, R.; Ghosh, A. State of Charge Estimation of Lithium-Ion Battery for Electric Vehicles Using Machine Learning Algorithms. *World Electr. Veh. J.* **2021**, *12*, 38. [\[CrossRef\]](#)
9. Wang, Q. Battery state of charge estimation based on multi-model fusion. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; pp. 2036–2041. [\[CrossRef\]](#)
10. de Bézenac, E.; Rangapuram, S.S.; Benidis, K.; Bohlke-Schneider, M.; Kurlle, R.; Stella, L.; Hasson, H.; Gallinari, P.; Januschowski, T. Normalizing Kalman filters for multivariate time series analysis. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 2995–3007.
11. Rangapuram, S.S.; Seeger, M.W.; Gasthaus, J.; Stella, L.; Wang, Y.; Januschowski, T. Deep state space models for time series forecasting. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7785–7794.
12. Liu, X.; Li, K.; Wu, J.; He, Y.; Liu, X. An extended Kalman filter based data-driven method for state of charge estimation of Li-ion batteries. *J. Energy Storage* **2021**, *40*, 102655. [\[CrossRef\]](#)
13. Xuan, L.; Qian, L.; Chen, J.; Bai, X.; Wu, B. State-of-Charge Prediction of Battery Management System Based on Principal Component Analysis and Improved Support Vector Machine for Regression. *IEEE Access* **2020**, *8*, 164693–164704. [\[CrossRef\]](#)
14. Ipek, E.; Eren, M.K.; Yilmaz, M. State-of-Charge Estimation of Li-ion Battery Cell using Support Vector Regression and Gradient Boosting Techniques. In Proceedings of the 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Istanbul, Turkey, 27–29 August 2019; pp. 604–609. [\[CrossRef\]](#)
15. Lucas, A.; Barranco, R.; Refa, N. EV Idle Time Estimation on Charging Infrastructure, Comparing Supervised Machine Learning Regressions. *Energies* **2019**, *12*, 269. [\[CrossRef\]](#)
16. Li, H.; Wang, S.; Islam, M.; Bobabee, E.D.; Zou, C.; Fernandez, C. A novel state of charge estimation method of lithium-ion batteries based on the IWOA-AdaBoost-Elman algorithm. *Int. J. Energy Res.* **2022**, *46*, 5134–5151. [\[CrossRef\]](#)
17. Sidhu, M.S.; Ronanki, D.; Williamson, S. State of Charge Estimation of Lithium-Ion Batteries Using Hybrid Machine Learning Technique. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019.
18. Li, C.; Chen, Z.; Cui, J.; Wang, Y.; Zou, F. The lithium-ion battery state-of-charge estimation using random forest regression. In Proceedings of the 2014 Prognostics and System Health Management Conference (PHM-2014 Hunan), Zhangjiajie, China, 24–27 August 2014.
19. Bian, C.; He, H.; Yang, S.; Huang, T. State-of-charge sequence estimation of lithium-ion battery based on bidirectional long short-term memory encoder-decoder architecture. *J. Power Sources* **2019**, *449*, 227558. [\[CrossRef\]](#)
20. Ma, L.; Hu, C.; Cheng, F. State of Charge and State of Energy Estimation for Lithium-Ion Batteries Based on a Long Short-Term Memory Neural Network. *J. Energy Storage* **2021**, *37*, 102440. [\[CrossRef\]](#)
21. US EPA. Urban Dynamometer Driving Schedule, Emission Standards Reference Guide [EB/OL]. USA, 2017. Available online: <https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules> (accessed on 11 January 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.