

Article

Improving the Reliability of Photovoltaic and Wind Power Storage Systems Using Least Squares Support Vector Machine Optimized by Improved Chicken Swarm Algorithm

Zhi-Feng Liu ^{1,2}, Ling-Ling Li ^{1,2}, Ming-Lang Tseng ^{3,4,*} , Raymond R. Tan ⁵ and Kathleen B. Aviso ⁵

¹ State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China

² Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability of Hebei Province, Hebei University of Technology, Tianjin 300130, China

³ Institute of Innovation and Circular Economy, Asia University, Taichung 41354, Taiwan

⁴ Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40447, Taiwan

⁵ Chemical Engineering Department, Gokongwei College of Engineering, De La Salle University, Manila 1004, Philippines

* Correspondence: tsengminglang@gmail.com or tsengminglang@asia.edu.tw

Received: 6 July 2019; Accepted: 3 September 2019; Published: 10 September 2019



Abstract: In photovoltaic and wind power storage systems, the reliability of the battery directly affects the overall reliability of the energy storage system. Failed batteries can seriously affect the stable operation of energy storage systems. This paper aims to improve the reliability of the storage systems by accurately predicting battery life and identifying failing batteries in time. The current prediction models mainly use artificial neural networks, Gaussian process regression and hybrid models. Although these models can achieve high prediction accuracy, the computational cost is high due to model complexity. Least squares support vector machine (LSSVM) is a computationally efficient alternative. Hence, this study combines the improved chicken swarm optimization algorithm (ICSO) and LSSVM into a hybrid ICSO-LSSVM model for the reliability of photovoltaic and wind power storage systems. The following are the contributions of this work. First, the optimal penalty parameter and kernel width are determined. Second, the chicken swarm optimization algorithm (CSO) is improved by introducing chaotic search behavior in the hen and an adaptive learning factor in the chicks. The performance of the ICSO algorithm is shown to be better than CSO using standard test problems. Third, the prediction accuracy of the three models is compared. For NMC1 battery, the predicted relative error of ICSO-LSSVM is 0.94%; for NMC2 battery, the relative error of ICSO-LSSVM is 1%. These findings show that the proposed model is suitable for predicting the failure of batteries in energy storage systems, which can improve preventive and predictive maintenance of such systems.

Keywords: least squares support vector machine; chicken swarm optimization algorithm; storage system; sustainable lithium-ion battery; chaotic search

1. Introduction

Fossil energy will be exhausted eventually, thus requiring people to develop new energy sources in the near future [1]. Also, the development of new energy has become mainstream in the world's energy strategy [2]. Nowadays, there has been a great effort placed in new combustion technology [3–7]

and end-of-pipe treatment technology [8–10] to reduce environmental pollution from fossil energy. Renewables have also become increasingly attractive. Due to the advantages of wind energy and solar energy, such as extremely low levels of emissions, more and more attention has been paid to such technologies for a power generation system in the world [11,12].

On the other hand, solar power and wind power generation are intermittent, unstable and uncontrollable. Energy storage systems can reduce the impact of wind power on the power grid, thus ensuring the stability of power system output [13,14]. Lithium-ion batteries have broad application prospects in wind and photovoltaic power generation systems because of its long service life, no environmental pollution, and fast charging speed [15–17]. As the batteries start to age, operational problems begin to occur which may affect the performance of integrated renewable energy systems [18]. For example, dead batteries may cause a security accident if not found in time. Predicting the end of life of a battery from early symptoms is thus essential.

Battery capacity declines over an increasing number of charging cycles [19]. The physical model and data-driven model are used to predict the life of lithium-ion batteries [20]. The main methods of physical model prediction methods include the degradation mechanism model and equivalent circuit model [21,22]. The data-driven methods mainly include support vector machine (SVM), artificial neural networks (ANNs) and Gaussian process regression [23–25].

Andre et al. [26] proposed an ANN model for forecasting the life of a lithium-ion battery. An ANN model has a strong nonlinear mapping ability to capture the behavior of lithium-ion batteries, but the training time of ANN is long and can lose predictive capability by overfitting training data. Gao et al. [27] proposed a new model for aging state analysis of lithium-ion batteries. In this model, symptoms such as capacity loss, solid electrolyte interphase (SEI) growth and deposited layer growth are used. The thickness of SEI was quantified by scanning electron microscopy (SEM) and transmission electron microscopy (TEM). Zhang et al. [28] proposed a hybrid method, which combines relevance vector machine and particle filter method. Experiments show that the proposed method can reduce the dependence on training samples.

In comparison to ANN, SVM has strong generalization ability and better robustness [29]. Least squares support vector machine (LSSVM) is developed on the basis of SVM [30]. LSSVM simplifies the solution of Lagrange multiplier by transforming inequality constraints into equality constraints in SVM. This results in computational efficiency. Liu and Jia [31] used LSSVM to forecast telecommunication traffic. To enhance the prediction performance, an improved artificial bee colony algorithm was used to optimize the LSSVM parameters. Zheng et al. [32] propose a novel method that combines wavelet technique with LSSVM to forecast dissolved gases in oil-immersed power transformers; the parameters are optimized with particle swarm optimization (PSO) algorithm. Gao and Huang [24] proposed a novel multi-kernel SVM to forecast the useful life of lithium-ion batteries with better prediction accuracy. Khelif et al. [33] used support vector regression (SVR) to predict the remaining useful life of equipment. This method does not need to estimate degradation states or the failure threshold.

The Chicken Swarm Optimization algorithm (CSO) is a novel swarm intelligence optimization algorithm based on PSO that simulates a hierarchical system with different ways of foraging. The CSO algorithm includes chicks, hens and roosters. It is a multi-population algorithm. Compared with classical PSO, CSO algorithm has stronger convergence ability by taking advantage of different foraging strategies used by three tiers of agents. This technique has proven to be useful for many engineering applications. Shayokh and Shin [34] proposed an improved CSO algorithm to solve a wireless sensor network node localization problem. Yu et al. [35] used CSO algorithm for deep well monitoring. The complexity of CSO algorithm can be reduced by the clustering method of wheel graph theory. Chen et al. [36] used a novel model based on improved boundary CSO to estimate the parameter of nonlinear systems. Hu et al. [37] applied the CSO algorithm to optimize the economic operation of a micro-grid.

This paper proposes the ICSO-LSSVM model to forecast the end of life of lithium-ion batteries as part of energy storage systems integrated with renewable energy systems. In order to enhance the

prediction accuracy, the ICSO algorithm optimizes the parameters of the LSSVM model. In particular, hen particles are sorted. The first 10% of the hen particles with superior performance replace the rooster particles with poor fitness, which speeds up the search speed of the chickens and avoids the rooster particles falling into local minima.

Furthermore, an adaptive learning factor is introduced into the chick location and the mutation operator is proposed. The adaptive learning factor makes the chicks move quickly to the best particles that speed up the convergence rate of the algorithm; the mutation operator broadens the shrinking population search space and makes the chicken swarm jump out of previous locally optimal locations. This feature enlarges the search space and reduces the risk of entrapment in local optima. By introducing chaotic search, mutation operator and adaptive learning factor in ICSO algorithm, the whole convergence performance is improved. The life of NMC1 and NMC2 batteries is forecasted by using this ICSO-LSSVM model, and the prediction error of the model is analyzed. Two groups of test data are used to test the ICSO-LSSVM model. The test results showed that the proposed model can predict the life of lithium-ion batteries efficiently and reliably.

This paper is organized as follows. Section 2 presents the ICSO-LSSVM method. Section 3 introduces the charge and discharge test data of lithium-ion battery. The results are discussed in detail in Section 4. Section 5 then gives the findings and conclusions.

2. Lifetime Prediction Model for Lithium-Ion Batteries

2.1. Least Squares Support Vector Machine(LSSVM) Principle

Because of the good generalization ability of support vector machine (SVM), SVM has been widely used for forecasting [38]. However, the training process of SVM is a constrained quadratic programming problem. The sample size equals the number of constraints. When the sample is too large, the training process takes a long time. A linear system of equations needs to be solved in the training process of LSSVM, which improves the training efficiency of SVM to some extent. The principle of LSSVM model is as follows:

$Q = \{(u_j, v_j) | j = 1, 2, \dots, n\}$ is the training sample, where $u_j (u_j \in R^m)$ is the input value, and $v_j (v_j \in R)$ is the output value. The kernel function maps samples. And regression function β , as indicated in Equation (1), is constructed in multidimensional space [39].

$$\beta = \frac{1}{2} \mu^T \mu + \frac{1}{2} \delta \sum_{i=1}^n e_i^2 \tag{1}$$

Note: μ is the weight vector; δ is the penalty coefficient; and e_i is the prediction error.

According to the principle of structural risk minimization, the constrained optimization problem is shown in Equation (4) [40]:

$$\begin{cases} \min \beta(\mu, e) = \frac{1}{2} \mu^T \mu + \frac{1}{2} \delta \sum_{i=1}^n e_i^2 \\ \text{s.t. } v_i = \mu^T \Phi(u_i) + o + e_i, i = 1, 2, 3 \dots n \end{cases} \tag{2}$$

where o is the offset and $\Phi(\bullet)$ represents the mapping method.

Based on the theory of constrained optimization, the corresponding Lagrange form can be obtained by introducing Lagrange multiplier $g_i (g_i \geq 0)$:

$$\psi = \frac{1}{2} \|\mu\|^2 + \frac{1}{2} \delta \sum_{i=1}^n e_i^2 - \sum_{i=1}^n g_i \{ \mu^T \Phi(u_i) + o + e_i - v_i \} \tag{3}$$

Equation (4) is obtained by the Karush–Kuhn–Tucher condition:

$$\begin{cases} \frac{\partial \psi}{\partial \mu} = 0 \rightarrow \mu = \sum_{i=1}^n g_i \Phi(u_i) \\ \frac{\partial \psi}{\partial o} = 0 \rightarrow \sum_{i=1}^n g_i = 0 \\ \frac{\partial \psi}{\partial e_i} = 0 \rightarrow g_i = \delta e_i \\ \frac{\partial \psi}{\partial g_i} = 0 \rightarrow \mu \Phi(u_i) + o + e_i - v_i = 0 \end{cases} \quad (4)$$

Equation (5) is shown as follows:

$$\begin{bmatrix} 0 & U \\ U^T & H + \delta^{-1}E \end{bmatrix} \begin{bmatrix} o \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ v \end{bmatrix} \quad (5)$$

where $v = [v_1, v_2, \dots, v_l]$; $U = [1; 1; \dots; 1]$; g is the Lagrange multiplier; E is the unit matrix; and H is the kernel function.

According to the Mercer condition, the relationship between the mapping function $\Phi(\bullet)$ and the kernel function H can be obtained using Equation (6):

$$H(u_i, u_j) = \exp\left(-\frac{\|u_i - u_j\|}{2\theta^2}\right) \quad (6)$$

where θ is the Gauss kernel parameter.

Therefore, the regression function of LSSVM is represented by Equation (7).

$$F(u) = \sum_{j=1}^n \delta_j H(u, u_j) + o \quad (7)$$

In LSSVM, parameters g and θ have a great influence on the generalization ability of LSSVM. The right g and θ can improve the prediction performance.

2.2. The Particle Swarm Optimization (PSO) Algorithm

Particle swarm optimization (PSO) is a classical swarm intelligence optimization algorithm. The optimization process of PSO algorithm is to imitate the foraging process of particles [41]. In the optimization process of the PSO algorithm, the current particle position can be updated by referring to its historical best position and global best position. The optimal position of the particle is obtained by multiple iterations [42,43].

Set in the dm dimension space, there are a total of R particles. In the l_{th} iteration, the position of the i_{th} particle in the j_{th} dimensional space is $pos_{i,j}^l (i = 1, 2, \dots, R; j = 1, 2, \dots, dm)$, and the velocity is $vel_{i,j}^l$. At the $(l + 1)_{th}$ iteration, the position and velocity of the particles are updated as follows [44,45].

$$vel_{i,j}^{l+1} = w * vel_{i,j}^l + q_1 * p_1 * (pbest_{i,j}^l - pos_{i,j}^l) + q_2 * p_2 * (pgest - pos_{i,j}^l) \quad (8)$$

$$pos_{i,j}^{l+1} = pos_{i,j}^l + vel_{i,j}^l \quad (9)$$

where w is the weight coefficient; q_1 and q_2 are the learning coefficients; p_1 and p_2 are random numbers between 0 and 1; $pbest$ is the optimal position of the individual; and $pgest$ is the global optimal position.

2.3. The Chicken Swarm Optimization Algorithm (CSO)

The CSO algorithm is a biologically inspired metaheuristic algorithm, which imitates swarm foraging [34,35]. The CSO algorithm includes roosters, hens, and chicks. The rooster is the leader of

the flock and leads the population to find food; the hen follows the rooster for food; and the chick follows the hen for foraging. The algorithm follows the following four rules:

Rule 1: There are many groups of chicken. There is a rooster, a small number of chicks, and several hens in each population.

Rule 2: Particles are grouped based on the fitness value. Particles with good fitness are chosen as roosters in each subgroup. Every rooster has the widest range of foraging. The particles with poor fitness are chosen as chicks which have the smallest foraging range. Other individuals are chosen as hens. The foraging range of hens is less than that of roosters but larger than that of chicks. Hens randomly select the group of roosters.

Rule 3: The hierarchical order is only updated during the iterative process.

Rule 4: The hens in each group forage around the roosters, and the chicks feed around the hens.

Because of the difference of the foraging ability of roosters, hens and chicks, different strategies are used to update their position.

It is assumed that D is search space and M is the population size. Each individual is depicted by position $x_{i,j}^t (i = 1, 2, \dots, M; j = 1, 2, \dots, D)$ at t moment [46,47]. The position update equation for the i_{th} rooster is shown in Equation (10):

$$x_{ij}^{t+1} = x_{ij}^t(1 + randn(0, \sigma^2)) \tag{10}$$

$$\sigma^2 = \begin{cases} e^{\left(\frac{f_k - f_i}{|f_i| + \epsilon}\right)} & f_i > f_k, i \neq k \\ 1 & else \end{cases} \tag{11}$$

where $randn(0, \sigma^2)$ is a normal distribution with a mean value of 0 and standard deviation of σ^2 ; ϵ is an infinite decimal; f_k is the fitness value of any rooster after removing the i_{th} rooster.

The position update equation of the i_{th} hen is shown in Equation (12):

$$x_{ij}^{t+1} = x_{ij}^t + e^{\left(\frac{f_i - f_r}{|f_i| + \epsilon}\right)} rand(x_{rj}^t - x_{ij}^t) + e^{(f_s - f_r)} rand(x_{sj}^t - x_{ij}^t) \tag{12}$$

where $r \neq s$; $rand$ is a random number between [0,1], which obeys uniform distribution.

The position of the i_{th} chick is as follows:

$$x_{ij}^{t+1} = x_{ij}^t + F(x_{mj}^t - x_{ij}^t) \tag{13}$$

where $F (F \in [0, 2])$ is a learning parameter; x_{mj}^t means the position of the hen followed by the chick.

2.4. The Improved Chicken Swarm Optimization Algorithm (ICSO)

As the iteration goes deeper, the search ability of rooster particles becomes worse. In addition, the number of hens is the largest, and the search results of the hen particles have a great influence on the algorithm. Therefore, the chaotic search is introduced in the ICSO algorithm. The hens are sorted based on their fitness values and the first 10% of the hen particles with good fitness are searched. After the search, the first 10% of the hen particles with superior performance replace roosters with poor fitness which speeds up the search speed of the swarm. The neighborhood points of many local optimal solutions are generated in the iterative process due to the characteristics of ergodicity and randomness of chaotic search [48].

In the ICSO algorithm, the chaotic search is constructed by using the Logistic equation [49]:

$$C_{i+1} = \mu C_i(1 - C_i) \quad i = 1, 2, \dots, n \tag{14}$$

where $C_i (C_i \in (0, 1))$ is a chaotic factor, and μ is the control parameter. When $\mu = 4$, the system is completely in the chaotic state.

The foraging ability of the rooster is the strongest, the hen is second, and the chick is the worst in the CSO algorithm. The adaptive learning factor is introduced in the chick's location to enhance the foraging ability of the chick. The relative change rate of the fitness value of chicks is defined as:

$$k = \frac{|f_i(t) - f_{best}|}{f_{best}} \tag{15}$$

where $f_i(t)$ is the fitness value of the i_{th} chick, and f_{best} is the fitness value of the best particle in the flock.

In the t_{th} generation, the adaptive learning factor of the i_{th} chick is updated:

$$s_i^t = \frac{1}{1 + \exp(-k)} \tag{16}$$

where k is the relative change rate.

After the improvement is complete, the position of the i_{th} chick is shown in Equation (17):

$$x_{ij}^{t+1} = s_i^t x_{ij}^t + x_{ij}^t (1 + randn(0, \sigma^2)) \tag{17}$$

where $randn(0, \sigma^2)$ is a normal distribution, 0 is the mean value, σ^2 is the standard deviation; and s_i^t is the adaptive learning factor of the i_{th} chick in the t_{th} generation.

The relative change rate k of the chicks reflects the difference between the i_{th} chick and the optimum particle. The adaptive learning factor s_i^t reflects the deviation degree of the position of the i_{th} chick and the optimum particle position in the flock. When the relative change rate k increases, the adaptive learning factor s_i^t increases. When the relative change rate k decreases, the adaptive learning factor s_i^t decreases. In this way, the movement of chick particles to the best particles is accelerated. The mutation operator is introduced in position of the chick to further avoid the chicks falling into extreme values. The fitness variance of chicks is as shown in Equation (18):

$$\sigma^2 = \frac{1}{n_k} \sum_{i=1}^{n_k} \left(\frac{f_i - f_{avg}}{f} \right)^2 \tag{18}$$

where n_k is the number of chicks; f_i is the fitness value of the i_{th} chick; f_{avg} is the average fitness value of the chicks; and $f = \max\{1, |f_i - f_{avg}|\}$ ($i \in [1, n_k]$) is the normalized factor.

The fitness variance σ^2 of the chicks reflects the aggregation degree of the fitness values of the chicks. The smaller the σ^2 is, the higher the aggregation degree of the fitness value is. When the chick particles fall into the local extreme value, the fitness variance σ^2 tends to 0. If the fitness variance σ^2 reaches the threshold, the chicks are mutated. The narrowing search space of the chick can be broadened by the mutation operation.

There are several steps in the optimization process of ICSO algorithm.

(1) The parameters of the algorithm are defined, such as the number of roosters n_r , hens n_h , chicks n_k , etc.

(2) The chicken flock X is initialized and the chickens according to their fitness values are grouped.

(3) To start the iteration, we first decide whether the status of the chicken flocks needs to be updated. If necessary, the chicken group status is updated; if not, the rooster's position is updated by Equation (10). Then the position of the hen is updated by Equation (12). The fitness values of the hens are sorted, and the first 10% of the hen particles are searched. After the search, the first 10% of the hen particles with superior performance replace the rooster particles with poor fitness value. The chick's position is updated by Equation (17), the fitness values of the chicks are calculated, and the mutation operator determines whether to reset the position of the chicks or not.

(4) Update individual fitness values and find global optimal locations.

(5) Determine whether to terminate the algorithm. If the termination condition is not met, return to step three to continue the iteration.

The ICSO algorithm flow chart is shown in Figure 1.

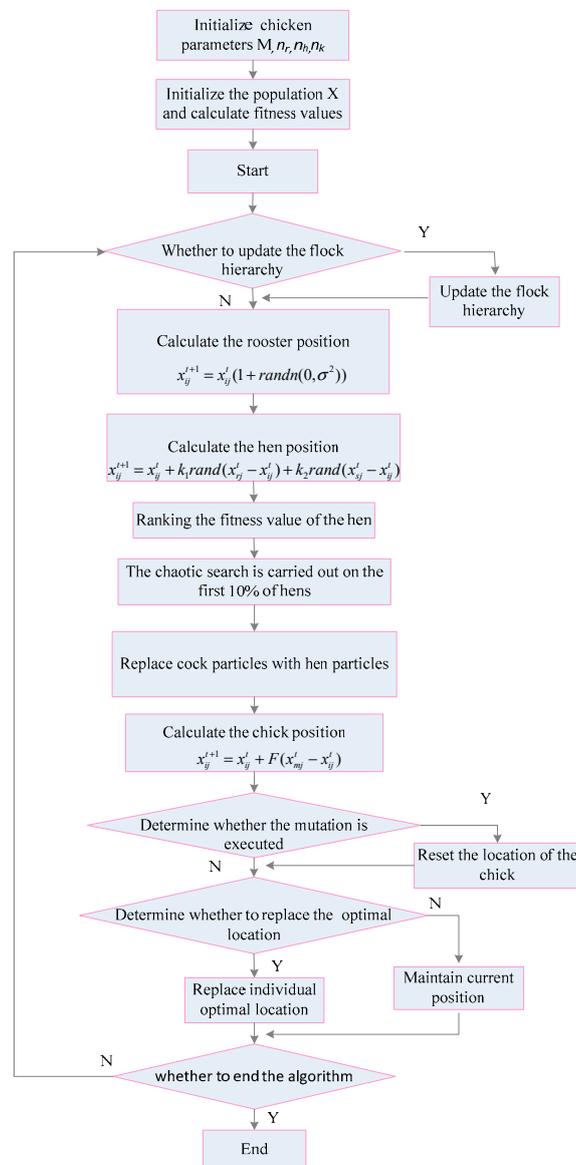


Figure 1. ICSO algorithm.

2.5. ICSO-LSSVM Model

Penalty parameter and kernel width parameter are two important parameters of the prediction model. The penalty parameter affects the generalization ability of the prediction model, and kernel width affects the training and prediction speed of the model. Intelligent algorithm ICSO is used to find the parameters of the LSSVM model to improve the prediction effect of the LSSVM model, and the ICSO-LSSVM model is thus constructed.

During training, Equation (19) is used as the fitness function of the ICSO algorithm.

$$f = \left(\sum_{j=1}^d (\hat{p}(j) - p(j))^2 \right) / d \tag{19}$$

where \hat{p} and p are the predicted and true values.

The residual life prediction process of lithium-ion battery is as follows.

- (1) The NMC1 and NMC2 battery capacity data are imported.
- (2) The input and output sample sets are normalized.
- (3) The parameters of chicken flocks are initialized, such as the number of roosters n_r , hens n_h and chicks n_k .
- (4) ICSO algorithm is used to optimize the Super-parameters of the LSSVM model.
- (5) The optimized parameters are inputted into the LSSVM model.
- (6) ICSO-LSSVM model is used to forecast the remaining life of batteries.
- (7) Evaluate the prediction error of the model.

The prediction flow of ICSO-LSSVM model is shown in Figure 2.

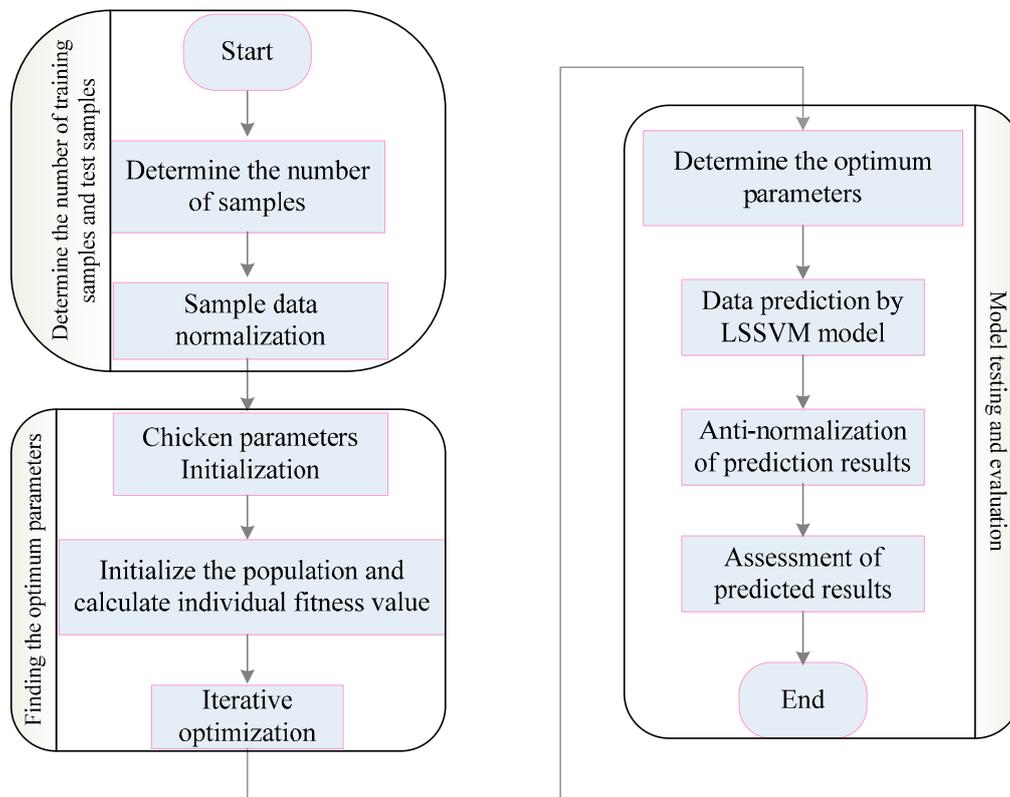


Figure 2. Life prediction process of ICSO-LSSVM.

3. Charge and Discharge Test

Ternary material $\text{Li}[\text{Ni}_x\text{Co}_y\text{Mn}_{1-x-y}]\text{O}_2$ is abbreviated as NMC. A lithium battery with a positive electrode of NMC ternary material is called an NMC lithium battery. The cell NMC1 and NMC2 with a capacity of 2000/mAh are used as the experimental object, and the charge and discharge experiments are carried out on NMC1 and NMC2 cells at a room temperature of 25 °C. Specific implementation measures are presented:

The constant current charging with the constant voltage is performed on NMC1 and NMC2 cells. The charging rate is 1.0 C, the charge cut-off voltage is 4.20 V, and the charge cut-off current is 0.02 A. Finally, the two batteries are put aside for about 30 min after charging.

The constant current discharge is performed on NMC1 and NMC2 cells. The discharge rate is 1.0 C, and the discharge cut-off voltage is 2.50 V. Finally, the two batteries are put aside for about 30 min after discharging.

The lithium-ion battery capacity decay curve is shown in Figure 3. As the number of cycles increases, the battery discharge capacity gradually decreases. The dotted line in Figure 3 indicates the failure threshold. When the battery capacity is reduced to the failure threshold, the battery fails. The failure threshold is 80% of the rated capacity in this paper. The failure thresholds of NMC1 and NMC2 battery are 1600/mAh.

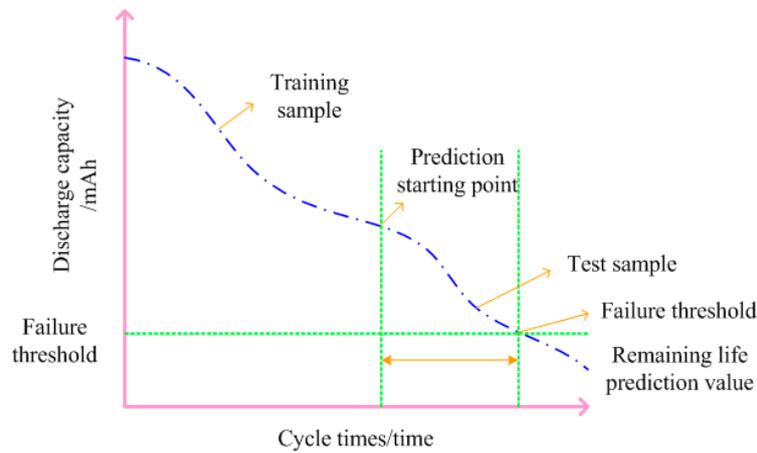


Figure 3. Capacity decay curve of lithium-ion battery.

NMC1 batteries include 760 lithium-ion battery capacity decay data, NMC1 batteries fail after 712 cycles, NMC2 batteries include 1060 lithium-ion battery capacity decay data, and NMC2 batteries fail after 999 cycles. In order to make the model fit the data better during the training process, 60–70% of the sample data were selected as the training samples in the study, and the remaining 30–40% of the data were selected as the test samples.

The NMC1 battery capacity decay curve is shown in Figure 4, and the NMC2 battery capacity decay curve is shown in Figure 5.

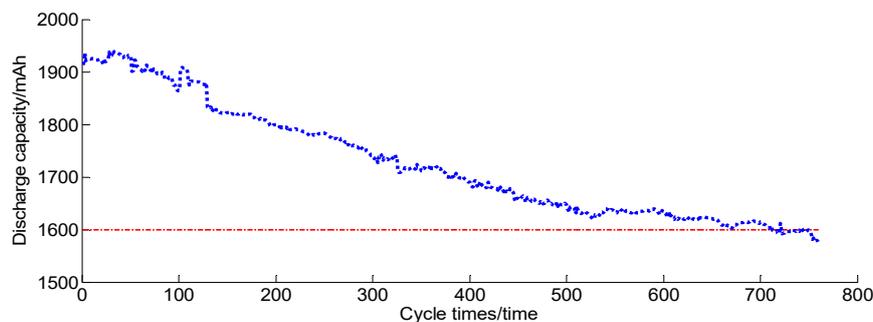


Figure 4. NMC1 battery capacity decay curve.

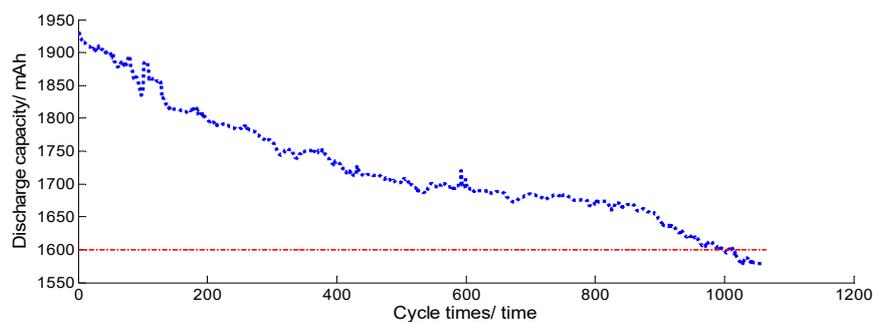


Figure 5. NMC2 battery capacity decay curve.

4. Test Simulation and Result Analysis

4.1. Analysis of Algorithm Convergence Performance

The parameters of the CSO algorithm and ICSO algorithm are set. The population size is 20; the number of roosters, chicks and hens are 4, 2 and 14; the number of hens with mother–child relationship is 6; the maximum number of cycles is 300. The parameters of the PSO algorithm are set. The population size is 20; the maximum particle speed is 50; the minimum particle speed is −50; the maximum particle position is 500; the minimum particle position is −500; and the iteration number is 300. The algorithms are tested on the same platform. The main parameters of the computer used in the simulation experiment are as follows: i5 processor, CPU main frequency 2.5 GHz, memory capacity 2 GB, hard disk capacity 500 GB.

Four test functions are used to test the algorithms. The range, optimal value and expression of the functions are as follows. The first test function is D1 ($D1 = \sum_{j=1}^S t_j^2$), and the optimal value of the function is 0, which ranges from [−100, 100]; the second test function is D2 ($D2 = \sum_{j=1}^S |t_j| + \prod_{j=1}^S |t_j|$), and the optimal value of the function is 0, which ranges from [−10, 10]; the third test function is D3 ($D3 = \sum_{j=1}^S (t_j + 0.5)^2$), and the optimal value of the function is 0, which ranges from [−100, 100]; the fourth test function is D4 ($D4 = \sum_{j=1}^S j * t_j^4 + random[0, 1)$), and the optimal value of the function is 0, which ranges from [−100, 100]. The dimension of the function is 20. Each function is tested 10 times for the three algorithms. The test results are as follows.

Table 1 shows the convergence accuracy and calculation cost of the three optimization algorithms. By analyzing the convergence results of the three algorithms, it is found that neither PSO nor CSO converge to 0 for the D1, D2 and D4 functions. Compared with CSO and ICSO, the convergence result of PSO is poor, which is caused by the fact that the ability of PSO to jump out of the minimum is poor. For D1 and D3, the ICSO converges to 0; for D2 and D4, the ICSO does not converge to 0, but the convergence value of the ICSO is significantly better than that of PSO and CSO.

Table 1. Optimization Results.

D	Algorithm	Optimum Value	Worst Optimization Value	Average Optimization Value	Time Consumed by 10 Optimization/s	Average Consumption Time for Optimizing/s
D1	CSO	6.79×10^{-21}	5.03×10^{-16}	5.73×10^{-17}	7.05	0.71
	PSO	0.28	1.05	0.52	5.11	0.51
	ICSO	0	0	0	8.94	0.89
D2	CSO	8.87×10^{-17}	3.39×10^{-14}	5.23×10^{-15}	8.73	0.87
	PSO	1.34	3.17	2.40	5.19	0.52
	ICSO	2.16×10^{-197}	8.56×10^{-188}	1.10×10^{-188}	9.76	0.98
D3	CSO	0	0	0	7.75	0.78
	PSO	0	4	1.30	4.52	0.45
	ICSO	0	0	0	10.81	1.08
D4	CSO	1.27×10^{-2}	0.27	0.04	8.33	0.83
	PSO	0.57	2.27	1.22	6.11	0.61
	ICSO	1.08×10^{-4}	2.3×10^{-3}	8.44×10^{-4}	10.87	1.09

For computing cost, the running time spent by iteration of the PSO is the smallest. For the four test functions, the total time consumed by the PSO program iteration is significantly shorter than that of CSO and ICSO. This is because the structure of the PSO algorithm is simple and the cost of calculation is small. For the four functions, the total time consumed by the PSO for 40 optimizations is 20.93 s; the total time consumed by the CSO program for 40 optimizations is 31.86 s; the total time consumed by the ICSO for 40 optimizations is 40.38 s. The time consumed by running ICSO programs is nearly twice as long as the PSO; it is 26.74% more than the CSO. Compared with PSO and CSO, ICSO runs longer. Because chaotic search is introduced into ICSO algorithm, the computation cost of ICSO is increased.

In order to analyze the convergence speed of the three algorithms, Figure 6 shows the convergence curves of the algorithms.

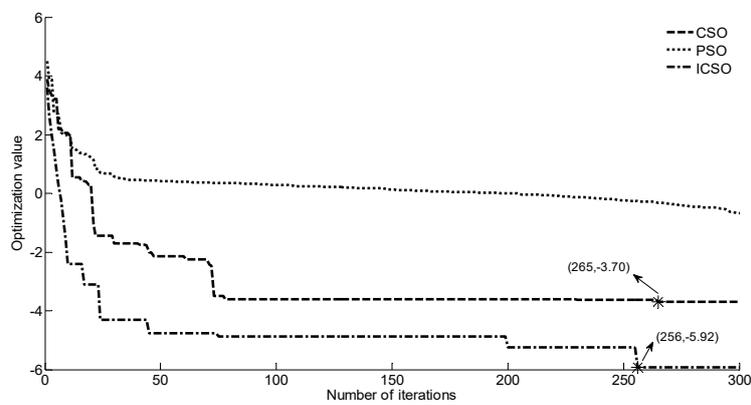
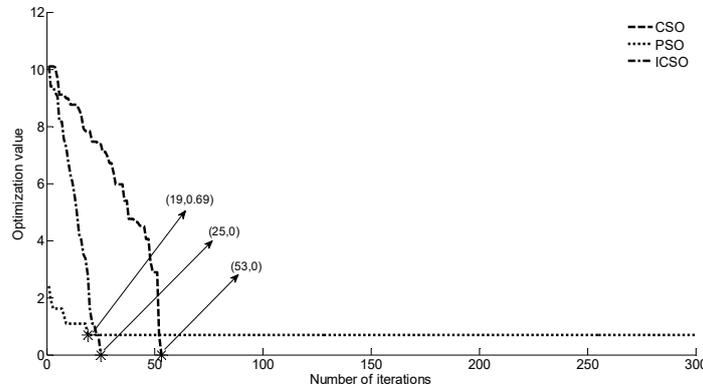
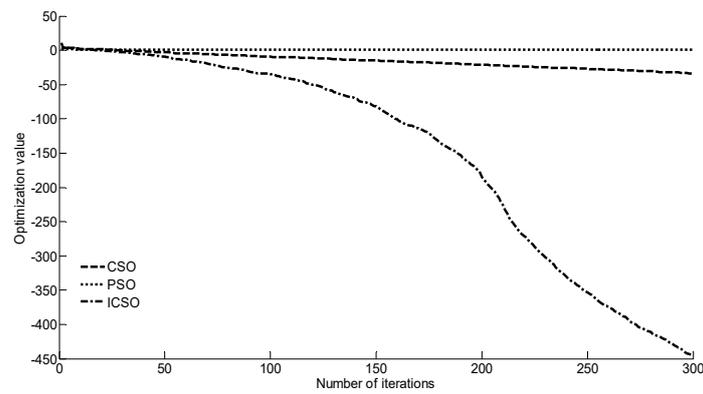
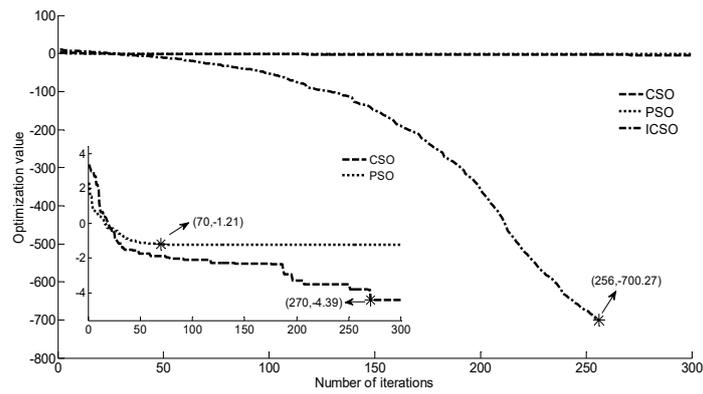


Figure 6. Convergence curves of the algorithms.

The convergence effect of the three algorithms can be more clearly observed from Figure 6. Compared with CSO, the convergence performance of ICSO is significantly improved. For the D1, D3, and D4 functions, ICSO converges to the optimal value 4, 28, and 9 generations earlier than CSO, respectively, and ICSO maintains a high convergence accuracy. Comprehensive comparison of the convergence data in Table 1 and the convergence curves in Figure 6 reveals that ICSO has better convergence ability.

4.2. Simulation Experiment on Life Prediction of Lithium-Ion Battery

Firstly, the ICSO-LSSVM model is trained. Then, the model is validated with the test samples. The ICSO-LSSVM model is tested by NMC1 and NMC2 battery capacity data. For NMC1 battery, the number of training samples and testing samples utilizes the first 500 and the last 260 samples respectively; for NMC2 battery, the number of training samples and testing samples utilizes the first 700 and the last 360 samples.

The goal of the optimization process is to find the minimum value of f . The percentage of relative error (RE) is used as the evaluation index, and the percentage of the relative error of capacity prediction is as follows:

$$RE = \frac{\hat{p} - p}{p} * 100\% \quad (20)$$

The predictive effects of three models were tested by using the data of NMC1 and NMC2 batteries, respectively. The first 500 sets of NMC1 battery capacity data are used as training samples; three models are trained by training samples. Then the 260 groups of data are predicted by PSO-LSSVM model, CSO-LSSVM model and ICSO-LSSVM model. The prediction curves are as follows.

From Figure 7, it can be seen that the degradation trend of the three prediction curves is similar to that of the actual value curves. The prediction curves of the three models reach the failure threshold at 220, 215 and 212 times, respectively. The higher degree of fitting of the red prediction curve indicates that the ICSO-LSSVM model has a stronger ability to interpret NMC1 battery data.

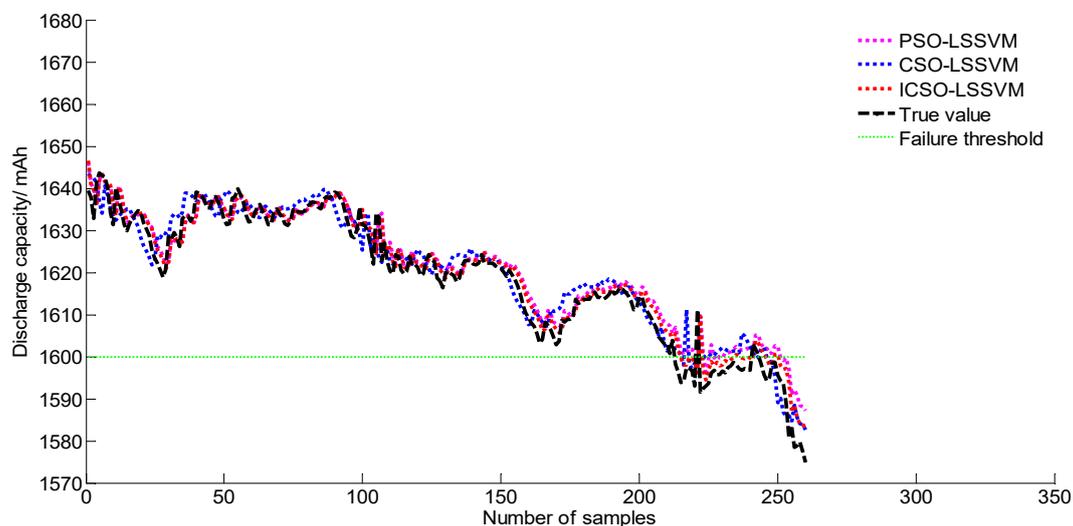


Figure 7. Discharge capacity prediction curve of NMC1 battery.

The prediction errors of the three models are calculated according to the prediction results of Figure 7, and the prediction results are evaluated according to the relative error RE.

As shown in Figure 8, The maximum relative error of the PSO-LSSVM model exceeds 1%. The prediction error of the CSO-LSSVM and ICSO-LSSVM models is controlled within 1%. With the increase in the number of predicted samples, the prediction errors of the three models did not show significant fluctuations, indicating that the three models have high stability for NMC1 battery prediction.

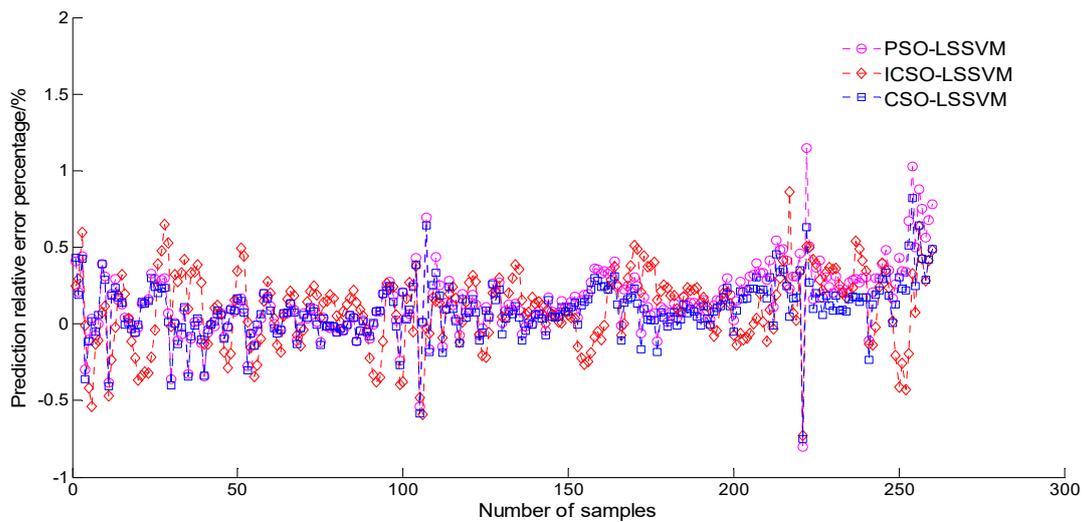


Figure 8. Model error curves.

Table 2 presents the RE of the PSO-LSSVM model, which fluctuates between $\pm 1.14\%$, and the average RE is 0.14%. The RE of the CSO-LSSVM model fluctuates between $\pm 0.86\%$, and the average RE of CSO-LSSVM model is 0.1559%. The RE of the ICSO-LSSVM model fluctuates between $\pm 0.82\%$, and the average RE of ICSO-LSSVM model is 0.11%. By comparison, The ICSO-LSSVM model controls the error to a small interval $[-0.75\% \ 0.82\%]$.

Table 2. Model error analysis.

Model	Maximum Reverse RE/%	Maximum Forward RE/%	Average RE/%
PSO-LSSVM	-0.81	1.13	0.14
CSO-LSSVM	-0.73	0.86	0.15
ICSO-LSSM	-0.75	0.82	0.11

The first 700 sets of NMC2 battery capacity data are used as training samples. Then the 360 groups of data are predicted by PSO-LSSVM, CSO-LSSVM and ICSO-LSSVM models. The NMC2 battery life prediction curves for each model are shown in Figure 9.

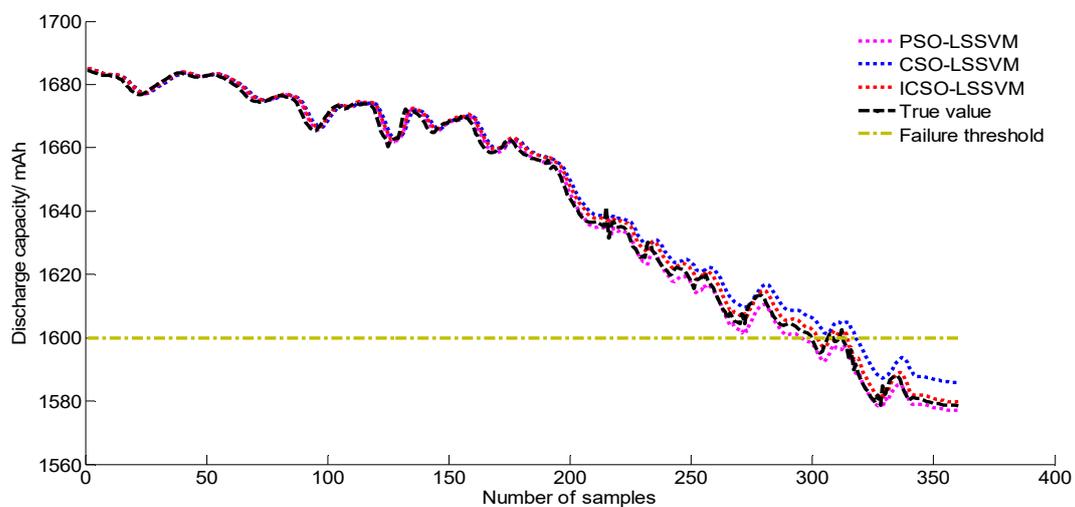


Figure 9. Discharge capacity prediction curve of NMC2 battery.

As shown in Figure 9, the black line reaches the failure threshold on the 299th time. The pink line, blue line and red line present the prediction curves of PSO-LSSVM, CSO-LSSVM and model and ICSO-LSSVM models. The prediction curves reach the failure threshold on the 295th, 317th and 302nd time.

For the remaining life of the NMC2 battery, the prediction results of each model are as follows.

The prediction error of the model before 200 samples is small, and the RE interval is controlled at $[-0.4\% \ 0.4\%]$. After 200 test samples, the prediction error of CSO-LSSVM and PSO-LSSVM increases, and the maximum RE of PSO-LSSVM is close to 0.8%. Compared to the blue and pink line, the red line fluctuations are more stable as shown in Figure 10.

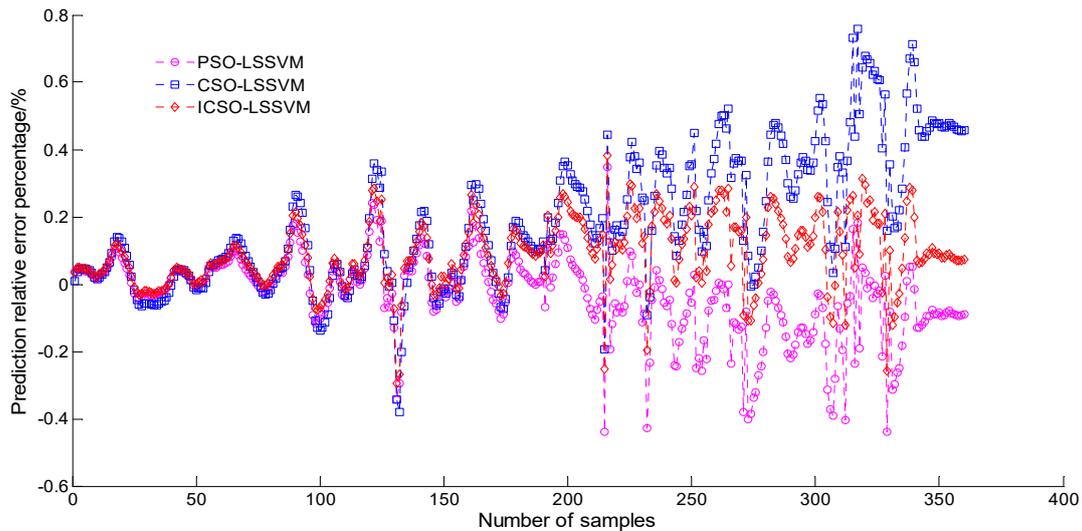


Figure 10. The RE of the three models.

Table 3 shows the RE intervals of the PSO-LSSVM, CSO-LSSVM and ICSO-LSSVM models, which are $\pm 0.5\%$, $\pm 0.8\%$ and $\pm 0.4\%$, respectively. The average RE values of the ICSO-LSSVM model are 0.02% and 0.09% smaller than the PSO-LSSVM model and CSO-LSSVM model, respectively. By comparison, it is found that the RE of ICSO-LSSVM is smaller than that of PSO-LSSVM and CSO-LSSVM.

Table 3. RE of NMC2 battery.

Model	Maximum Reverse RE/%	Maximum Forward RE/%	Average RE/%
PSO-LSSVM	-0.44	0.35	0.09
CSO-LSSVM	-0.38	0.76	0.20
ICSO-LSSVM	-0.29	0.38	0.11

Table 4 presents the remaining life prediction of the NMC1 battery, the $|RE|$ of ICSO-LSSVM model is 0.94%, which is lower than 3.77% of the PSO-LSSVM model and 1.41% of the CSO-LSSVM model. For the remaining life prediction of NMC2 battery, the $|RE|$ of ICSO-LSSVM model is 1.00%, which is lower than -1.33% of the PSO-LSSVM model and 6.02% of the CSO-LSSVM model. The test result showed that the ICSO-LSSVM model is more accurate than the PSO-LSSVM and CSO-LSSVM models.

Table 4. Comparison of three models for NMC1 and NMC2 battery capacity prediction.

Battery Type	Prediction Model	True Value of Remaining Life	Predicted Value of Residual Life	RE
NMC1	PSO-LSSVM	212	220	3.77%
	CSO-LSSVM		215	1.41%
	ICSO-LSSVM		214	0.94%
NMC2	PSO-LSSVM	299	295	1.33%
	CSO-LSSVM		317	6.02%
	ICSO-LSSVM		302	1.00%

5. Conclusions

Wind power and PV power play an important role in decarbonizing electricity generation. Lithium-ion battery energy storage systems are an important part of such an energy generation system as they stabilize the inherent volatility of wind and sunlight. Deteriorating batteries can be replaced in time by predicting the battery life to enhance the reliability of energy-storage systems. This work developed an ICSO-LSSVM model for this purpose.

The novel features of this work are as follows. First, the chaotic search, mutation operator and adaptive learning factor are introduced into the CSO algorithm to improve its performance. Secondly, results show that the precision and convergence speed of the ICSO algorithm are better than CSO PSO; thirdly, three models are tested using the 260 groups of data of NMC1 battery capacity data and the 360 groups of data of NMC2 battery capacity data. These results show that the improved ICSO-LSSVM model is suitable for predicting the life of lithium-ion batteries.

The main engineering contribution of this new model is that it can improve the safety and reliability of integrated PV and wind power systems by enhancing predictive maintenance capability for the storage units. This capability has important implications for grid decarbonization. Nonetheless, this study has limitations. This paper studies the residual life of a single battery, but in practical applications, the energy storage equipment is a battery pack with multiple components. The remaining life of battery packs should be predicted as a whole in future research.

Author Contributions: Z.-F.L. designed the experiment and got the experimental data; R.R.T, K.B.A. and L.-L.L. analyzed the experimental data; Z.-F.L. and M.-L.T. wrote the paper. All authors contributed to discussing and revising the manuscript.

Funding: This work was supported by the Natural Science Foundation of Hebei Province, China [grant numbers E2018202282] and the key project of Tianjin Natural Science Foundation of China [grant numbers 19JCZDJC32100].

Acknowledgments: The authors would like to thank the editors and reviewers for their constructive comments on this study.

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

References

1. Wang, Y.P.; Li, J. Spatial Spillover Effect of Non-Fossil Fuel Power Generation on Carbon Dioxide Emissions across China's Provinces. *Renew. Energy* **2019**, *136*, 317–330. [[CrossRef](#)]
2. Mamat, R.; Sani, M.S.M.; Sudhakar, K. Renewable Energy in Southeast Asia: Policies and Recommendations. *Sci. Total Environ.* **2019**, *670*, 1095–1102.
3. Jiaqiang, E.; Zhang, Z.; Tu, Z.; Zuo, W.; Hu, W.; Han, D.; Jin, Y. Effect analysis on flow and boiling heat transfer performance of cooling water-jacket of bearing in the gasoline engine turbocharger. *Appl. Therm. Eng.* **2018**, *130*, 754–766.
4. Deng, Y.; Liu, H.; Zhao, X.; Jiaqiang, E.; Chen, J. Effects of cold start control strategy on cold start performance of the diesel engine based on a comprehensive preheat diesel engine model. *Appl. Energy* **2018**, *210*, 279–287. [[CrossRef](#)]

5. Zhang, Z.; Jiaqiang, E.; Deng, Y.; Zuo, W.; Peng, Q.; Hieu, P.; Yin, Z. Effects of fatty acid methyl esters proportion on combustion and emission characteristics of a biodiesel fueled marine diesel engine. *Energy Convers. Manag.* **2018**, *159*, 244–253. [[CrossRef](#)]
6. Zhang, B.; Jiaqiang, E.; Gong, J.; Yuan, W.; Zuo, W.; Li, Y.; Fu, J. Multidisciplinary design optimization of the diesel particulate filter in the composite regeneration process. *Appl. Energy* **2016**, *181*, 14–28. [[CrossRef](#)]
7. Zhang, B.; Jiaqiang, E.; Gong, J.; Yuan, W.; Zhao, X.; Hu, W. Influence of structural and operating factors on performance degradation of the diesel particulate filter based on composite regeneration. *Appl. Therm. Eng.* **2017**, *121*, 838–852. [[CrossRef](#)]
8. Ashuri, T.; Zaaijer, M.B.; Martins, J.; van Bussel, G.J.W.; van Kuik, G.A.M. Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy. *Renew. Energy* **2014**, *68*, 893–905. [[CrossRef](#)]
9. Jiaqiang, E.; Zuo, W.; Gao, J.; Peng, Q.; Zhang, Z.; Hieu, P. Effect analysis on pressure drop of the continuous regeneration-diesel particulate filter based on NO₂ assisted regeneration. *Appl. Therm. Eng.* **2016**, *100*, 356–366.
10. Chen, M.R.; Zeng, G.Q.; Lu, K.D. Constrained multi-objective population extremal optimization based economic-emission dispatch incorporating renewable energy resources. *Renew. Energy* **2019**, *143*, 277–294. [[CrossRef](#)]
11. Ye, B.; Yang, P.; Jiang, J.J.; Miao, L.X.; Shen, B.; Li, J. Feasibility and economic analysis of a renewable energy powered special town in China. *Resour. Conserv. Recycl.* **2017**, *121*, 40–50. [[CrossRef](#)]
12. Solomon, A.A.; Kammen, D.M.; Callaway, D. Investigating the impact of wind-solar complementarities on energy storage requirement and the corresponding supply reliability criteria. *Appl. Energy* **2016**, *168*, 130–145. [[CrossRef](#)]
13. Du, W.; Bi, J.; Lv, C.; Littler, T. Damping torque analysis of power systems with DFIGs for wind power generation. *IET Renew. Power Gener.* **2017**, *11*, 10–19. [[CrossRef](#)]
14. Hill, C.A.; Such, M.C.; Chen, D.; Gonzalez, J.; Grady, W.M. Battery Energy Storage for Enabling Integration of Distributed Solar Power Generation. *IEEE Trans. Smart Grid* **2012**, *3*, 850–857. [[CrossRef](#)]
15. Ma, X.; Qiu, D.F.; Tao, Q.; Zhu, D.Y. State of Charge Estimation of a Lithium Ion Battery Based on Adaptive Kalman Filter Method for an Equivalent Circuit Model. *Appl. Sci. Basel* **2019**, *9*, 2765. [[CrossRef](#)]
16. Liu, Y.; Wang, H.C.; Yang, K.K.; Yang, Y.N.; Ma, J.Q.; Pan, K.M.; Pang, H. Enhanced Electrochemical Performance of Sb₂O₃ as an Anode for Lithium-Ion Batteries by a Stable Cross-Linked Binder. *Appl. Sci. Basel* **2019**, *9*, 2677. [[CrossRef](#)]
17. Tian, L.L.; Wang, M.K.; Xiong, L.; Guo, H.J.; Huang, C.; Zhang, H.R.; Chen, X.D. The Effect of Different Mixed Organic Solvents on the Properties of p(OPal-MMA) Gel Electrolyte Membrane for Lithium Ion Batteries. *Appl. Sci. Basel* **2018**, *8*, 2587. [[CrossRef](#)]
18. Jaguemont, J.; Boulon, L.; Venet, P.; Dubé, Y.; Sari, A. Lithium-Ion Battery Aging Experiments at Subzero Temperatures and Model Development for Capacity Fade Estimation. *IEEE Trans. Veh. Technol.* **2016**, *65*, 4328–4343. [[CrossRef](#)]
19. Richa, K.; Babbitt, C.W.; Gaustad, G.; Wang, X. A future perspective on lithium-ion battery waste flows from electric vehicles. *Resour. Conserv. Recycl.* **2014**, *83*, 63–76. [[CrossRef](#)]
20. Wang, D.; Yang, F.; Tsui, K.-L.; Zhou, Q.; Bae, S.J. Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Spherical Cubature Particle Filter. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1282–1291. [[CrossRef](#)]
21. Waag, W.; Kabitz, S.; Sauer, D.U. Experimental investigation of the lithium-ion battery impedance characteristic at various conditions and aging states and its influence on the application. *Appl. Energy* **2013**, *102*, 885–897. [[CrossRef](#)]
22. Sanchez, L.; Couso, I.; Gonzalez, M. A design methodology for semi-physical fuzzy models applied to the dynamic characterization of LiFePO₄ batteries. *Appl. Soft Comput.* **2014**, *14*, 269–288. [[CrossRef](#)]
23. Sbarufatti, C.; Corbetta, M.; Giglio, M.; Cadini, F. Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks. *J. Power Sources* **2017**, *344*, 128–140. [[CrossRef](#)]
24. Gao, D.; Huang, M.H. Prediction of Remaining Useful Life of Lithium-ion Battery based on Multi-kernel Support Vector Machine with Particle Swarm Optimization. *J. Power Electron.* **2017**, *17*, 1288–1297.
25. Wang, Z.P.; Ma, J.; Zhang, L. State-of-Health Estimation for Lithium-Ion Batteries Based on the Multi-Island Genetic Algorithm and the Gaussian Process Regression. *IEEE Access* **2017**, *5*, 21286–21295. [[CrossRef](#)]

26. Andre, D.; Nuhic, A.; Soczka-Guth, T.; Sauer, D.U. Comparative study of a structured neural network and an extended Kalman filter for state of health determination of lithium-ion batteries in hybrid electric vehicles. *Eng. Appl. Artif. Intell.* **2013**, *26*, 951–961. [[CrossRef](#)]
27. Gao, Y.Z.; Zhang, X.; Yang, J.; Guo, B.J.; Zhou, X. A Novel Model for Lithium-Ion Battery Aging Quantitative Analysis Based on Pseudo Two-Dimension Expressions. *Int. J. Electrochem. Sci.* **2019**, *14*, 3180–3203. [[CrossRef](#)]
28. Zhang, Y.Z.; Xiong, R.; He, H.W.; Pecht, M. Validation and Verification of a Hybrid Method for Remaining Useful Life Prediction of Lithium-Ion Batteries. *J. Clean. Prod.* **2019**, *212*, 240–249. [[CrossRef](#)]
29. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
30. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [[CrossRef](#)]
31. Liu, J.X.; Jia, Z.H. Telecommunication Traffic Prediction Based on Improved LSSVM. *Int. J. Pattern Recognit. Artif. Intell.* **2018**, *32*, 16. [[CrossRef](#)]
32. Zheng, H.B.; Zhang, Y.Y.; Liu, J.F.; Wei, H.; Zhao, J.H.; Liao, R.J. A novel model based on wavelet LS-SVM integrated improved PSO algorithm for forecasting of dissolved gas contents in power transformers. *Electr. Power Syst. Res.* **2018**, *155*, 196–205. [[CrossRef](#)]
33. Khelif, R.; Chebel-Morello, B.; Malinowski, S.; Laajili, E.; Fnaiech, F.; Zerhouni, N. Direct Remaining Useful Life Estimation Based on Support Vector Regression. *IEEE Trans. Ind. Electron.* **2017**, *64*, 2276–2285. [[CrossRef](#)]
34. Al Shayokh, M.; Shin, S.Y. Bio Inspired Distributed WSN Localization Based on Chicken Swarm Optimization. *Wirel. Pers. Commun.* **2017**, *97*, 5691–5706. [[CrossRef](#)]
35. Yu, X.W.; Zhou, L.X.; Li, X.Y. A Novel Hybrid Localization Scheme for Deep Mine Based on Wheel Graph and Chicken Swarm Optimization. *Comput. Netw.* **2019**, *154*, 73–78. [[CrossRef](#)]
36. Biswal, P.C. Ancestor Approach to Number of Spanning Trees of Wheel Graph W-N in Terms of Number of Spanning Trees of Fan Graph F-N. *Adv. Sci. Lett.* **2016**, *22*, 567–569. [[CrossRef](#)]
37. Zahid, A.; Saleem, M.U.; Kashif, A.; Khan, M.; Meraj, M.A.; Irfan, R. Spanning Simplicial Complex of Wheel Graph W-N. *Algebra Colloq.* **2019**, *26*, 309–320. [[CrossRef](#)]
38. Datta, A.; Augustin, M.J.; Gupta, N.; Viswamurthy, S.R.; Gaddikeri, K.M.; Sundaram, R. Impact Localization and Severity Estimation on Composite Structure Using Fiber Bragg Grating Sensors by Least Square Support Vector Regression. *IEEE Sens. J.* **2019**, *19*, 4463–4470. [[CrossRef](#)]
39. Wang, G.J.; Zhang, G.Q.; Choi, K.S.; Lu, J. Deep Additive Least Squares Support Vector Machines for Classification with Model Transfer. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1527–1540. [[CrossRef](#)]
40. Shahsavari, A.; Bagherzadeh, S.A.; Mahmoudi, B.; Hajizadeh, A.; Afrand, M.; Nguyen, T.K. Robust Weighted Least Squares Support Vector Regression Algorithm to Estimate the Nanofluid Thermal Properties of Water/Graphene Oxide-Silicon Carbide Mixture. *Phys. Stat. Mech. Appl.* **2019**, *525*, 1418–1428. [[CrossRef](#)]
41. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
42. Le, L.T.; Nguyen, H.; Zhou, J.; Dou, J.; Moayedi, H. Estimating the Heating Load of Buildings for Smart City Planning Using a Novel Artificial Intelligence Technique PSO-XGBoost. *Appl. Sci. Basel* **2019**, *9*, 2714. [[CrossRef](#)]
43. Cheng, S.F.; Lu, F.; Peng, P.; Wu, S. Multi-Task and Multi-View Learning Based on Particle Swarm Optimization for Short-Term Traffic Forecasting. *Knowl. Based Syst.* **2019**, *180*, 116–132. [[CrossRef](#)]
44. Almahdi, S.; Yang, S.Y. A Constrained Portfolio Trading System Using Particle Swarm Algorithm and Recurrent Reinforcement Learning. *Expert Syst. Appl.* **2019**, *130*, 145–156. [[CrossRef](#)]
45. Gharebaghi, S.A.; Zangoeei, E. Chaotic particle swarm optimization in optimal active control of shear buildings. *Struct. Eng. Mech.* **2017**, *61*, 347–357. [[CrossRef](#)]
46. Chen, S.L.; Yang, R.Y.; Yang, R.H.; Yang, L.; Yang, X.Z.; Xu, C.B.; Liu, W.P. A Parameter Estimation Method for Nonlinear Systems Based on Improved Boundary Chicken Swarm Optimization. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 11. [[CrossRef](#)]
47. Hu, H.; Li, J.; Huang, J. Economic Operation Optimization of Micro-grid Based on Chicken Swarm Optimization Algorithm. *High Volt. Appar.* **2017**, *53*, 119–125.

48. Gupta, S.; Deep, K. An Efficient Grey Wolf Optimizer with Opposition-Based Learning and Chaotic Local Search for Integer and Mixed-Integer Optimization Problems. *Arab. J. Sci. Eng.* **2019**, *44*, 7277–7296. [[CrossRef](#)]
49. Zhang, Z.X.; Yang, R.N.; Li, H.Y.; Fang, Y.H.; Huang, Z.Y.; Zhang, Y. Antlion Optimizer Algorithm Based on Chaos Search and Its Application. *J. Syst. Eng. Electron.* **2019**, *30*, 352–365.



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).