Remaining useful life prediction hybrid model of lithium-ion battery based on improved GWO-LightGBM.

ZHOU, Y., WANG, S., SHEN, X., SOJIB, A.S.M. and FERNANDEZ, C.

2024

Remaining Useful Life Prediction Hybrid Model of Lithium-ion Battery Based on Improved GWO-LightGBM

Yifei Zhou
Southwest University of Science and Technology
School of Information Engineering
Mian Yang, China
976850947@qq.com

Shunli Wang*
Southwest University of Science and Technology
School of Information Engineering
Mian Yang, China
wangshunli1985@qq.com

Xianfeng Shen
Southwest University of Science and Technology
School of Information Engineering
Mian Yang, China
shenxianfeng0929@qq.com

Ahamed S M Sojib
Southwest University of Science and Technology
School of Information Engineering
Mian Yang, China
smsojibahmed@gmail.com

Carlos Fernandez
Robert Gordon University
School of Pharmacy and Life Sciences
Aberdeen, UK
c.fernandez@rgu.ac.uk

***Abstracts*—Lithium-ion batteries, as the core of new energy vehicles, determine the safety of new energy vehicles. Remaining useful life of the battery is the most important parameter, and it is particularly important to estimate the remaining life accurately. This paper proposes a hybrid algorithm of GWO algorithm and LightGBM algorithm based on improved convergence factor and proportional weights, which is used to predict the remaining life of lithium-ion batteries. It is verified by using NASA data set, which proves that the optimization of GWO algorithm can significantly improve LightGBM algorithm. RMSE, MAE and MAPE increased by 71.46%, 80.59% and 75.79% respectively.**

***Keywords*—remaining useful life, improved convergence factor, proportional weights, grey wolf optimization algorithm, LightGBM algorithm, hybrid algorithm**

I. INTRODUCTION

Since lithium-ion (Li-ion) batteries are the key component of new energy vehicles, which have gradually gained popularity due to the depletion of fossil fuels and the rapid development of new energy, accurately estimating the remaining usable life (RUL) of Li-ion batteries has become more difficult. Recent studies have shown that hybrid models using deep learning have gained popularity as a method of RUL prediction [1, 2]. This publication [3] proposes a generalized hybrid particle filter-Short Term memory network (PF-LSTM) prediction approach that precisely describes the degraded state of equipment when paired with subtraction fuzzy cluster analysis. This paper [4] uses adaptive noise completely integrated empirical mode decomposition to break down the original capacity signal into two components: a global degradation trend and a local fluctuation. The Adaptive Neural Fuzzy Inference System (ANFIS) is then fed these two components in order to make predictions. Seizing the chance, it put out a hybrid model for LightGBM method parameter optimization for RUL prediction, based on enhanced convergence factor and proportional weights.

II. METHODOLOGY

A. Improve grey wolf optimizer algorithm

A metaheuristic algorithm called the Grey Wolf Optimizer (GWO) is based on the social structure and hunting techniques of grey wolves [5]. The hierarchy of the pack is rigid: *W* is the lowest ranking, *a* (alpha) is in charge, *b* is subordinate, and *c* obeys *a* and *b*. Tracking, pursuing, encircling, and ultimately assaulting the target are all part of the hunt.

The best solution in the population is designated as *a*, the second-best solution as *b*, the third-best solution as *c*, and the remaining individuals are designated as *w*. This assumes that the population size of the grey wolves is *N*. The location of the *i*-th grey wolf is shown as. The following is a description of the mathematical model used to explain how grey wolves hunt:

$$D = \left| C \cdot X_p(t) - X(t) \right| \tag{1}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{2}$$

Among them, *t* denotes the current iteration number, *A* and *C* are coefficient vectors, *X* represents the position vector of the prey, and X_i is the position vector of an single grey wolf.

$$A = 2a \cdot r_1 - a \tag{3}$$

$$C = 2 \cdot r_2 \tag{4}$$

where, r_1 and r_2 are both random vectors within [0,1], and *a* is a convergence factor that linearly decreases from 2 to 0 with the number of iterations. The positions of the other grey wolves in the population are jointly determined by the positions of *a*, *b*, and *c*.

$$\begin{cases} D_a = C_1 \cdot X_a - X,\ X_1 = X_a - A_1 \cdot D_a \\ D_b = C_1 \cdot X_b - X,\ X_1 = X_b - A_1 \cdot D_b \\ D_c = C_1 \cdot X_c - X,\ X_1 = X_c - A_1 \cdot D_c \end{cases} \tag{5}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{6}$$

GWO algorithm's proficiency in executing both localized and global search paradigms is intricately influenced by the fine-tuned value of A, as meticulously outlined in publication [6]. The subsequent Fig. 1 offers a comprehensive visualization of the exact progression followed by CGWO methodology, underscoring the intricate interplay between *A* and the algorithm's search capabilities.

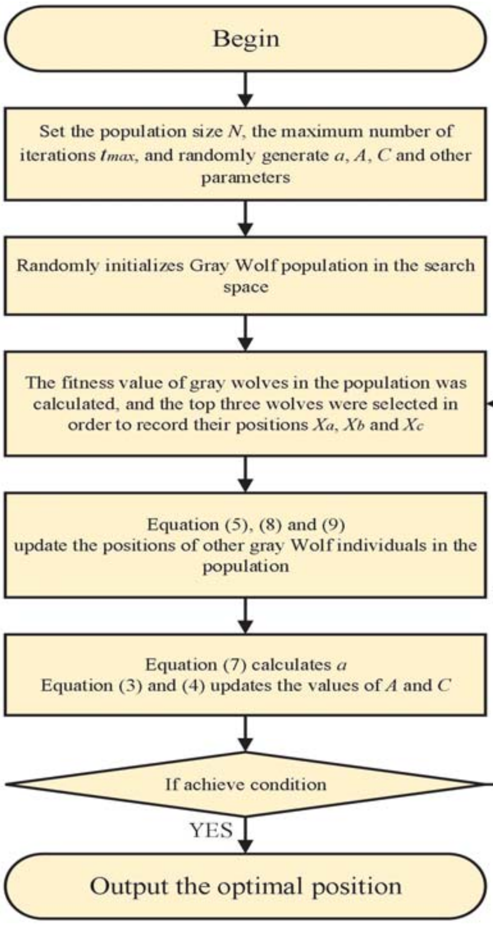

Fig. 1. Flow chart of convergence factor and proportional weight grey wolf optimizer algorithm

In the intricate dynamics of wolf pack hunting behavior, the parameter A plays a pivotal role in determining the pack's strategic adaptation. When the value of A transcends unity, it signifies a shift towards a more expansive hunting strategy. In this scenario, the wolves broaden their search perimeter, engaging in a global exploration that enhances their ability to swiftly converge upon potential prey across vast territories. This approach ensures a wider net is cast, maximizing the chances of encountering and successfully capturing prey. Conversely, when A dips below 1, the pack's hunting tactics undergo a transformation, narrowing their focus to a localized search. This concentrated effort enables the wolves to meticulously scrutinize a smaller area, intensifying their attacks on identified prey and leading to a more gradual but targeted convergence. Thus, the regulation of A serves as a flexible mechanism, allowing the wolf pack to dynamically adjust its hunting strategy based on the prevailing conditions and optimize its chances of success. Although (3) depicts A as varying linearly with a convergence factor that decreases from 2 to 0, the real-world search pattern exhibits nonlinearity. To more accurately capture this dynamic behavior, this paper introduces a cosine-shaped modification to the convergence factor, as demonstrated in (7).

$$\begin{cases} a = a_{final} + (a_{initial} - a_{final})\dfrac{1+[\cos(\frac{(t-1)\pi}{t_{\max}-1})]^n}{2}, t \le \dfrac{1}{2}t_{\max} \\ a = a_{final} + (a_{initial} - a_{final})\dfrac{1+\left|\cos(\frac{(t-1)\pi}{t_{\max}-1})\right|^n}{2}, \dfrac{1}{2}t_{\max} \le t \le t_{\max} \end{cases} \tag{7}$$

In the equation, $a_{initial}$ and a_{final} represent the initial and final values of the convergence factor a respectively. In this paper, it set $a_{initial} = 2$ and $a_{final} = 0$. t is the current iteration number, t_{max} is the maximum number of iterations, and n is the decremental exponent with $0 < n \le 1$. The variation of a is depicted in Fig. 2.

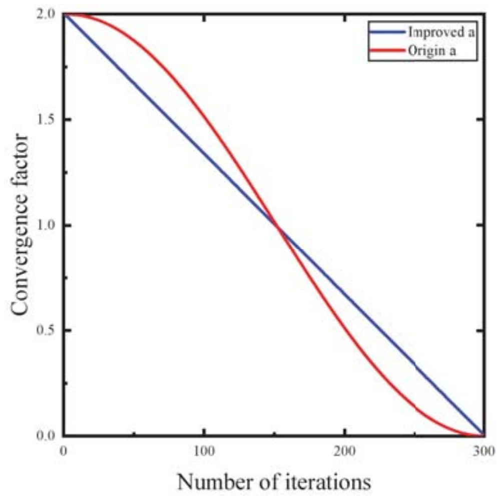

Fig. 2. Comparison chart of optimization parameters a

Fig. 2 highlights a distinct difference between the original and modified convergence factors, denoted as a. While the original a undergoes a steady linear decrease, the newly introduced cosine-shaped modification initially slows down the decline, preserving a higher value for an extended period. This extended phase of a relatively large ' fosters improved search efficiency by allowing for a broader exploration. Subsequently, the rate of decline accelerates, maintaining a smaller a for a longer duration, which refines the search and enhances precision. This adaptive approach successfully harmonizes the phases of global search exploration and local search refinement.Based on the optimization of the inclusion factor a, a fitness-based proportional weight is introduced, which is expressed as follows:

$$\begin{aligned} W_a &= \frac{f_a + f_b + f_c}{f_a} \\ W_b &= \frac{f_a + f_b + f_c}{f_b} \\ W_c &= \frac{f_a + f_b + f_c}{f_c} \end{aligned} \tag{8}$$

$$X(t+1) = \frac{X_1 \cdot W_{\mathrm{a}} + X_2 \cdot W_{\mathrm{b}} + X_3 \cdot W_{\mathrm{c}}}{W_{\mathrm{a}} + W_{\mathrm{b}} + W_{\mathrm{c}}} \tag{9}$$

where W_a, W_b, and W_c represent the weights assigned to the alpha, beta, and delta wolves, respectively. f_a, f_b, and f_c denote the fitness values of the alpha, beta, and delta wolves. The allocation of weights within the wolf pack is determined by their respective fitness levels, where the alpha wolf, leading the hunt, receives the highest weight. This hierarchy descends, with the beta wolf occupying the second-highest weight, and the omega wolf, theoretically having less insight into the prey's whereabouts, being assigned the lowest weight.

B. LightGBM algorithm

Gradient boosting decision tree (GBDT) is a popular classifier algorithm. Given a training set $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where x are the data samples and y are the class labels. That use $F(x)$ to represent the estimated function and the optimization goal of GBDT is to minimize the loss function $L(y, F(x))$:

$$\hat{F} = \arg\min_{F} E_{x,y}[L(y, f(x))] \tag{10}$$

Then, it can obtain the iterative criterion of the GBDT using *a* line search to minimize the loss function.

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{11}$$

where $\gamma_m = \arg\min \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$, m is the iteration number, $h_m(x)$ represents the base decision tree.

Despite its strengths, Gradient Boosting Decision Tree (GBDT) can encounter challenges in terms of both efficiency and accuracy when dealing with massive datasets or high-dimensional features. To mitigate these limitations, Ke [7] introduced LightGBM, an advanced gradient boosting algorithm tailored for enhanced performance. LightGBM incorporates two key techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). These innovative strategies help to streamline the learning process and improve the model's ability to handle large-scale and complex data, thereby enhancing both its efficiency and accuracy.

Traditionally, in the framework of Gradient Boosting Decision Trees (GBDT), the information gain metric serves as the cornerstone for selecting split points within each tree node. However, LightGBM introduces a paradigm shift by leveraging Gradient-based One-Side Sampling (GOSS) as its means of determining split points. Instead of relying on information gain, LightGBM calculates the variance gain, a metric tailored to the GOSS methodology, to identify optimal split points. This approach allows LightGBM to prioritize data instances in a more targeted manner, leading to improved performance and efficiency. Then the subset B whose size is $b \times |A^c|$ is randomly selected from the retained samples A^c. Finally, the instances are split through the estimated variance $\widetilde{V}_j(d)$ on $A \cup B$.

$$\widetilde{V}j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^i(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)} \right) \tag{12}$$

where $A_l = \{x_i \in A: x_{ij} \leq d\}$, $A_r = \{x_i \in A: x_{ij} > d\}$, $B_l = \{x_i \in B: x_{ij} \leq d\}$, $B_r = \{x_i \in B: x_{ij} > d\}$, g_i represents the negative gradient of the loss function, $(1\text{-}a)/b$ is employed to normalize the sum of gradients.

High-dimensional datasets frequently display sparsity, featuring numerous mutually exclusive sparse features. This inherent property enables the utilization of EFB to accelerate the training process of GBDT, thereby enhancing efficiency. The computation complexity of LightGBM is reduced to $O\,(data * bundle)$ from $O\,(data * feature)$, where $bundle \ll feature$.

Essentially, LightGBM presents an optimized iteration of GBDT, integrating GOSS and EFB strategies to elevate computational efficiency without sacrificing prediction accuracy. GOSS introduces a variance gain-based mechanism to identify optimal split points, enhancing the precision of node partitioning. Simultaneously, EFB streamlines the training process by intelligently bundling exclusive features into a compressed set of dense representations. The LightGBM model, denoted as $F_M(x)$, is formulated through a sophisticated weighted combination method, as detailed in Equation (13), further refining the predictive capabilities of the underlying GBDT framework.

$$F_M(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{13}$$

where M is the maximum number of iteration and $h_m(x)$ is the base decision tree.

Optimizing LightGBM's intricate parameters can pose significant challenges, necessitating a tailored approach. To unleash the full potential of LightGBM's performance, we harness the power of the CGWO algorithm. This algorithm is specifically designed to refine LightGBM's parameters, ensuring that each is meticulously tuned to deliver optimal results. By leveraging CGWO, we aim to maximize the effectiveness of LightGBM, as outlined in [8], thereby enhancing its predictive capabilities and overall performance.

III. EXPERIMENTS AND ANALYSIS

In this study, the NASA battery open data set was selected to evaluate the proposed model. NASA used batteries with a rated capacity of 2.2Ah to conduct cyclic charge-discharge experiments. Based on the total number of battery experiment cycles, 40% was used as the training set, and the last 60% was used as the test set.

A. Feature extraction

Three health characteristics of constant current charging time (CCCT), constant voltage charging time (CVCT) and constant voltage drop time (CVDT) are obtained by feature extraction of current and voltage data. As shown in Fig. 3.

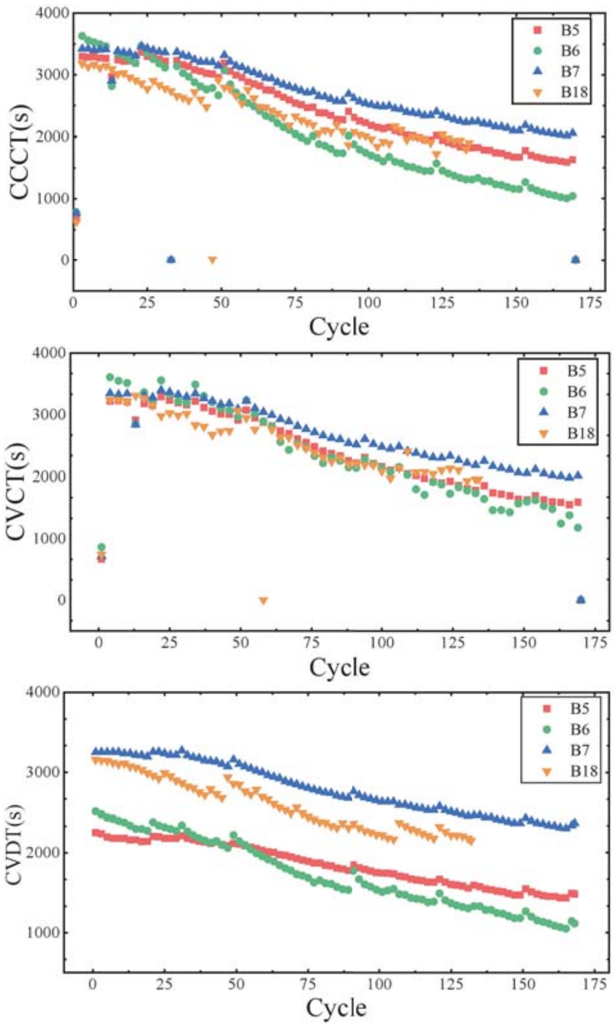

Fig. 3. Feature extraction result

B. Result analysis

In essence, the hybrid model incorporated CCCT, CVCT, and CVDT as input variables, iteratively refining its predictions to derive the estimated residual capacity. Figure 3 showcases the comparative experimental outcomes for four distinct battery types, illustrating the model's performance in assessing their capacities.

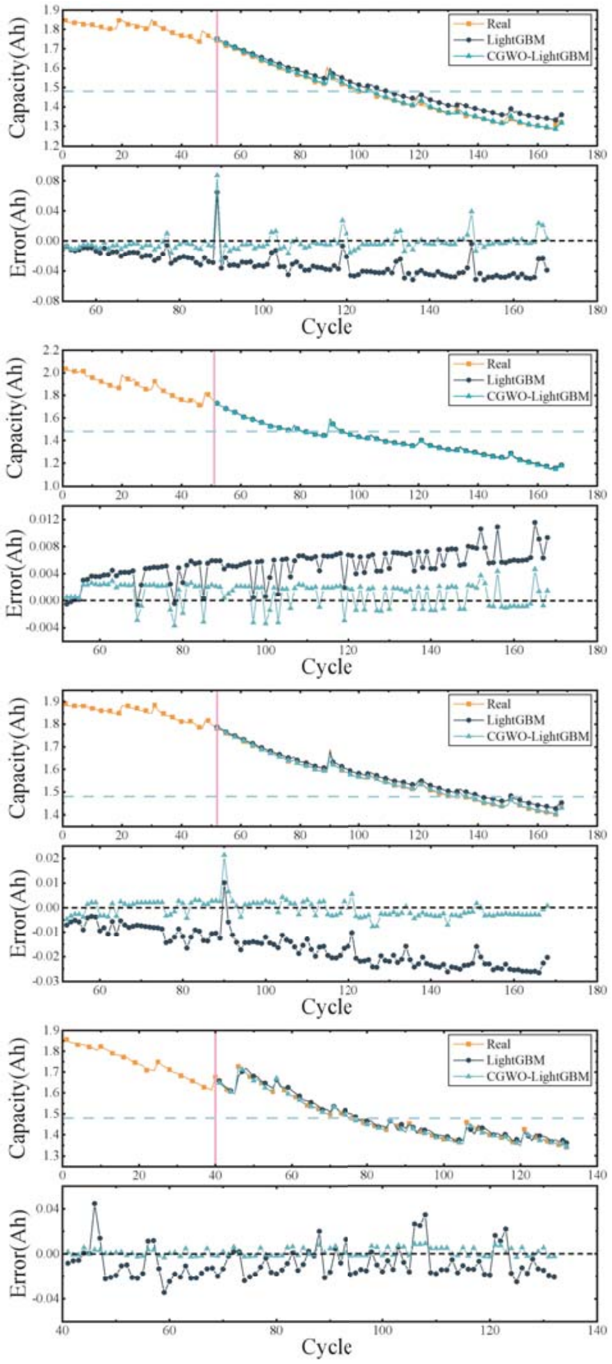

Fig. 4. Experimental results of four batteries

As prominently illustrated in Fig. 4, a notable improvement in predictive accuracy is observed for the LightGBM algorithm, which has been optimized utilizing the hybrid CGWO algorithm. This optimization leads to a substantially more precise estimation of residual capacity when compared against the standard baseline algorithm. Experimental results of four batteries have proved the accuracy and robustness of CGWO-LightGBM algorithm. The specific evaluation indicators are shown in Table 1 below.

TABLE I. THREE EVALUATION INDICATORS

Cell	**Algorithm**	**Evaluation index**		
		RMSE	MAE	MAPE
B0005	LightGBM	1.728	1.596	2.196%
	CGWO-LightGBM	0.612	0.391	0.531%
B0006	LightGBM	0.284	0.623	0.390%
	CGWO-LightGBM	0.101	0.093	0.133%
B0007	LightGBM	0.879	0.814	1.064%
	CGWO-LightGBM	0.178	0.143	0.184%
B0018	LightGBM	0.825	0.726	0.978%
	CGWO-LightGBM	0.189	0.150	0.208%

As shown in Table 1 above, it can be seen that after mixing the CGWO algorithm, various indicators of the prediction results of the four batteries have been significantly improved, with an average reduction of RMSE by 71.46%, MAE by 80.59% and MAPE by 75.79%. It is proved that CGWO algorithm improves the accuracy of LightGBM algorithm, and also proves that the hybrid model has high accuracy and robustness.

IV. CONCLUSION

In this paper, it proposed a hybrid model using CGWO algorithm to optimize LightGBM algorithm parameters. Firstly, convergence factors and proportional weights are introduced to optimize the basic gwo algorithm, and then the optimized CGWO is used to optimize LightGBM important parameters. Finally, NASA battery open data set is used. After feature extraction of current and voltage, three parameters CCCT, CVCT and CVDT are obtained, which are substituted into CGWC-LightGBM hybrid model for remaining capacity prediction as health features. The final results show that CGWO improves the accuracy and robustness of LightGBM algorithm significantly, which proves the effectiveness of CGWO-LightGBM algorithm.

REFERENCES

[1] X. Li, D. Yu, V. S. Byg, and S. D. Ioan, "The development of machine learning-based remaining useful life prediction for lithium-ion batteries," *Journal of Energy Chemistry,* vol. 82, pp. 103-121, 2023.

[2] M. H. Lipu *et al.*, "A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations," *Journal of cleaner production,* vol. 205, pp. 115-133, 2018.

[3] K. Xue, J. Yang, M. Yang, and D. Wang, "An improved generic hybrid prognostic method for RUL prediction based on PF-LSTM learning," *IEEE Transactions on Instrumentation and Measurement,* vol. 72, pp. 1-21, 2023.

[4] S. E. Samada, V. Puig, and F. Nejjari, "Robust TS-ANFIS MPC of an autonomous racing electrical vehicle considering the battery state of charge," *IEEE/ASME transactions on mechatronics,* 2023.

[5] S. N. Makhadmeh *et al.*, "Recent advances in Grey Wolf Optimizer, its versions and applications," *IEEE Access,* 2023.

[6] H. Kasahara and S. Narita, "Practical multiprocessor scheduling algorithms for efficient parallel processing," *IEEE Transactions on computers,* vol. 33, no. 11, pp. 1023-1029, 1984.

[7] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems,* vol. 30, 2017.

[8] H. Yang, Z. Chen, H. Yang, and M. Tian, "Predicting coronary heart disease using an improved LightGBM model: Performance analysis and comparison," *IEEE Access,* vol. 11, pp. 23366-23380, 2023.