

Predictive Prognostic Model for Lithium Battery Based on A Genetic Algorithm (GA-ELM) Extreme Learning Machine

JAMSHER ALI¹, SHI YONG SHENG¹, ABDUL REHMAN², IMRAN AFZAL³,
SHOAIB UL HASSAN¹

¹Department of Electrical and Computer Engineering
Shaanxi University of Science and Technology
Xian, Shaanxi, China.

²Department of Electrical and Control Engineering
Harbin University of Science and Technology
Harbin, China.

³Department of Electrical and Electronics Engineering
Hunan University. Hunan, China.

DOI: 10.29322/IJSRP.10.12.2020.p10818

<http://dx.doi.org/10.29322/IJSRP.10.12.2020.p10818>

Abstract: Prognostic and health management (PHM) verify the working safely and reliably of lithium batteries, PHM system determines the State of Health (SOH) and the Remaining Useful Life (RUL). To avoid severe negative consequences of the system, this paper presents the preliminary development of genetic algorithm, using an extreme learning machine (ELM) method to predict the State of Health (SOH) of the lithium-ion battery. The low prediction accuracy of the State of Health (SOH) of lithium battery, the extreme learning machine (ELM) algorithm improved by genetic algorithm can improve the prediction accuracy of SOH of lithium battery. The weight from the ELM input layer to the hidden layer and the threshold of the hidden layer unit are randomly generated. The ELM algorithm only needs to set the number of hidden layer units and the type of hidden layer activation function. The proposed GA-ELM approach was compared against other traditional algorithms, namely, Back Propagation (BP), Support Vector Machine (SVM), Radial Based function (RBF). Extreme learning machine (ELM) introduces genetic algorithm (GA) to optimize the weight of the input layer to the hidden layer and the threshold of the hidden layer unit. This method can enhance the stability of the ELM algorithm. The experiments compared the GA-ELM algorithm with the ELM algorithm, BP algorithm, RBF algorithm and SVR algorithm for the prediction of lithium battery SOH. The results show that the GA-ELM algorithm has improved prediction accuracy and algorithm stability compared with other algorithms.

Keywords: Prognostic and health management (PHM); State of Healthy (SOH); Extreme learning machine; Genetic algorithm; Model simulation

I. INTRODUCTION

As a high-performance energy storage device, lithium batteries are used in various occasions, namely electric vehicles (EVs), Laptops, Mobile phones, Calculators etc. Therefore, research and development are being actively performed to improve the lithium-ion battery performance.

State of Health (SOH) is an important parameter for evaluating the performance of lithium batteries. Predicting the health status of lithium batteries accurately has important guiding significance for the rational use of batteries.

The prediction of the health status of lithium batteries includes empirical and performance-based methods. The methods based on empirical prediction include the cycle and amperage life methods. The methods based on performance prediction include model-based methods and data-driven methods. And a fusion-based approach.

Due to the highly non-linear changes in the parameters of the lithium battery when its performance is degraded, the linear prediction of the health status of the lithium battery is less accurate. Non-linear prediction methods of lithium battery health include PF algorithm, SVR algorithm, and BP algorithm. Non-linear prediction models are more accurate than linear prediction models.

In this experiment, a batch of 18650 lithium iron phosphate batteries from a certain brand was used as the research object, and the relationship between the health status of lithium batteries and the changes in battery external parameters was explored.

The experiment uses ELM algorithm, BP algorithm, SVR algorithm and RBF algorithm to construct the battery.

The prediction model of SOH is compared with GA-ELM algorithm model, and the results show that GA-ELM algorithm prediction ability is better than other algorithm models.

II. EXTREME LEARNING MACHINE (ELM)

Extreme learning machine (ELM) proposed is a single-hidden layer feed forward networks (SLFNs) which randomly selected the input weights and analytically determines the output weights of SLFNs. ELM consists of 3 layers: input layer, hidden layer and output layer. The

input layer has n neurons, representing n input variables and the hidden layer has l neurons; the output layer has m neurons corresponding to m Output variable.

The connection weight w of the input layer and the hidden layer is:

$$\omega = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{l1} & \omega_{l2} & \dots & \omega_{ln} \end{bmatrix}_{l \times n} \quad (1)$$

In the formula, w_{ij} represents the connection weight between the i_{th} neuron in the input layer and the j_{th} neuron in the hidden layer. The connection weight β between the hidden layer and the output layer is:

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{l1} & \beta_{l2} & \dots & \beta_{lm} \end{bmatrix}_{l \times m} \quad (2)$$

In the formula, β_{jk} represents the connection weight of the j_{th} neuron in the hidden layer and the k_{th} neuron in the output layer.

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{bmatrix}_{l \times 1} \quad (3)$$

In the formula, b_i represents the threshold value of the i_{th} neuron in the hidden layer. The activation function $g(x)$ of the hidden layer can choose a continuous nonlinear function as the output mapping function.

The output T of the ELM network is:

$$T = [t_1 \quad t_2 \quad \dots \quad t_Q]_{m \times Q} \quad (4)$$

$$t_j = \begin{bmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{mj} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^l \beta_{i1} g(\omega_i x_j + b_i) \\ \sum_{i=1}^l \beta_{i2} g(\omega_i x_j + b_i) \\ \vdots \\ \sum_{i=1}^l \beta_{im} g(\omega_i x_j + b_i) \end{bmatrix}_{m \times 1} \quad (5)$$

$j = 1, 2, \dots, Q;$

Where, $\omega_i = [\omega_{i1} \quad \omega_{i2} \quad \dots \quad \omega_{in}]$

$$x_j = [x_{1j} \quad x_{2j} \quad \dots \quad x_{nj}]$$

The above formula can be expressed as $H\beta = T'$

Where T' is the transposed matrix of the output matrix T , and H is the output matrix of the hidden layer.

$$H = \begin{bmatrix} g(\omega_1 x_1 + b_1) & g(\omega_2 x_1 + b_2) & \dots & g(\omega_l x_1 + b_l) \\ g(\omega_1 x_2 + b_1) & g(\omega_2 x_2 + b_2) & \dots & g(\omega_l x_2 + b_l) \\ \vdots & \vdots & \ddots & \vdots \\ g(\omega_1 x_Q + b_1) & g(\omega_2 x_Q + b_2) & \dots & g(\omega_l x_Q + b_l) \end{bmatrix}_{Q \times l}$$

If the number of hidden layer units is equal to the number of training samples, for any given w and b , the network can approximate the training samples with zero errors. When the hidden layer is smaller than the number of training samples, the network training error can approach arbitrary accuracy. When the activation function $g(x)$ of the hidden layer unit is infinitely differentiable, the weight w and the hidden layer threshold b from the input layer to the hidden layer can be randomly selected, and the values of ω and b remain unchanged during the algorithm training process. The weight matrix β from the hidden layer to the output layer can be obtained by.

$$\min_{\beta} \|H\beta - T'\| \quad \beta = H^+ T' \quad (6)$$

Where H^+ is the Moore-Penrose generalized inverse of H .

The LM algorithm steps include:

- Determine the number of hidden layer neurons, and randomly generate the weight w from the input layer to the hidden layer and the threshold b of the hidden layer;
- Select the hidden layer activation function $g(x)$ as an infinitely differentiable function;
- Calculate the output layer H of the hidden layer and solve the output layer weights

$$\beta = H^+ T' \quad (7)$$

III. GA-ELM FRAMEWORK

Genetic Algorithm (GA) is an evolutionary algorithm. The algorithm encodes the parameters of the problem as the individuals to be trained, Train individuals to select, intersect, and mutate operations and calculate fitness values. The highest fitness value is selected as the optimal individual. The optimal individual obtains the optimal value of the parameter to be solved through the decoding operation. Genetic algorithm is suitable for solving nonlinear optimization problems that are difficult to be solved by traditional algorithms. Genetic algorithms provide a general framework for solving nonlinear problems, the algorithm optimizes specific problem parameters by encoding into chromosomes,

It reduces the risk that traditional optimization algorithms (such as gradient descent method) easily fall into local minima.

The steps for ELM parameter optimization using GA include:

- Determine the network topology, select the appropriate number of hidden layer units

according to the data size, and select the appropriate hidden layer activation function $g(x)$;

- The initial population of parameters is obtained by encoding the hidden layer weights and thresholds of the ELM, bringing the decoded parameters into the ELM network, training and verifying the model's prediction accuracy, and the error value $Mse = E(T_{sim} - T_{test}^2)$ T_{sim} is the actual value of the test set;
- Cross, mutate, recombine the elite individuals of the chromosome to obtain the fitness value of the offspring individuals, and select the individual with the best fitness through the ranking function. When the loop reaches the loop maximum or the test error is lower than the set threshold, the GA-ELM algorithm stops,

The algorithm flowchart is shown in Fig. 1.

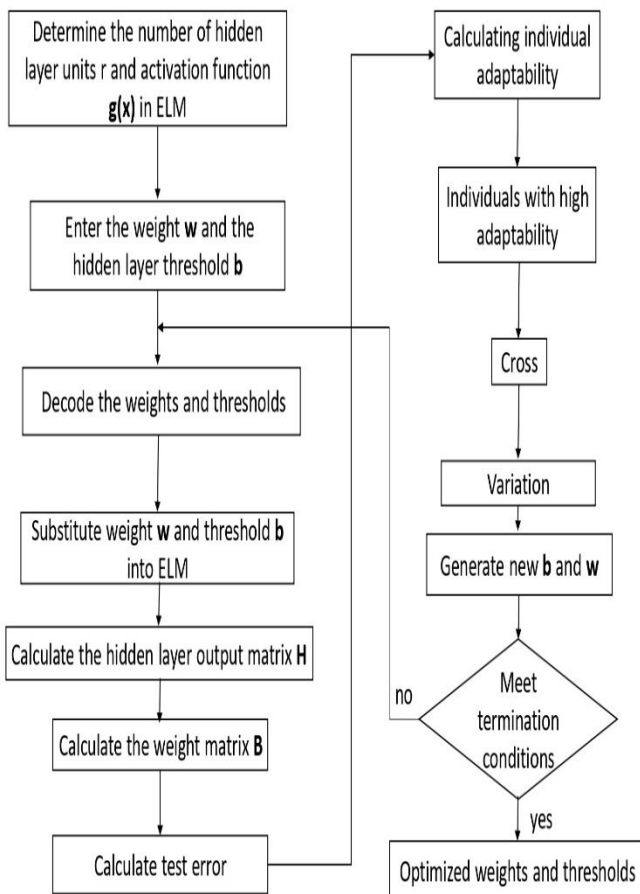


Fig. 1. GA-ELM Framework

IV. DESCRIPTION LITHIUM BATTERY HEALTH

A. Definition of health status

Battery health is equal to the ratio of available battery capacity to rated capacity,

$$SOH = \frac{C_{norm}}{C_{new} \times 100\%} \quad (8)$$

Where C_{new} is the amount of battery Constant capacity, C_{norm} is the available capacity of the battery, this formula

is based on the remaining power Method to define the health status of lithium batteries

B. Performance parameters of 2 lithium battery

- Discharge DC internal resistance

The increase in the number of discharges of the lithium battery, due to the decrease in the electrolyte activity of the battery and the aging of the electrode material, leads to a decrease in the battery's conductivity. The results are reflected in the battery parameters, that is, the internal resistance of the battery.

- Average discharge voltage

The discharge average voltage refers to the terminal voltage of the battery when the amount of charge drops to half of the initial value during the discharge of the battery. As the number of discharge cycles of the battery increases, the battery's ability to carry lithium ions decreases, and the average voltage of the battery decreases.

- Discharge rate decay rate

The discharge capacity decay rate refers to the ratio of the amount of charge discharged in this discharge cycle to the amount of charge discharged in the previous discharge cycle. As the number of discharge cycles increases, the overall battery capacity shows a downward trend.

The discharge capacity decay rate, that is, the speed of the capacity decay, reflects the health of the battery.

V. TEST RESULTS

Measure the median voltage, discharge DC internal resistance, and discharge decay rate of the lithium battery during the discharge cycle, and calculate and record the SOH value of the lithium battery at this time. The number of experimental cycles, the median voltage, DC internal resistance and discharge decay rate are used as inputs to the prediction model.

The lithium battery SOH value is used as the output of the prediction model. The experimental data are shown in Table -1.

Table -I Battery experiment data

cycles	Average voltage/V	Internal resistance	Discharge decay rate /%	Lithium battery SOH
1	3.481	0.065	100	1.00
2	3.467	0.067	98	0.972
3	3.461	0.067	96	0.964
4	3.463	0.071	96	0.960
5	3.460	0.072	95	0.956
6	3.478	0.062	98	0.981
7	3.474	0.063	98	0.986
8	3.473	0.069	97	0.979
9	3.462	0.068	95	0.956
10	3.458	0.073	95	0.950
⋮	⋮	⋮	⋮	⋮

95	3.449	0.069	97	0.834
96	3.440	0.068	95	0.816
97	3.424	0.075	93	0.799
98	3.419	0.081	92	0.791
99	3.436	0.073	94	0.805
100	3.442	0.070	95	0.812

VI. EXPERIMENTAL VERIFICATION

A. ELM and GA-ELM algorithm model simulation

In order to verify the effectiveness of the improved ELM algorithm, the number of ELM hidden layer units is set to 3, the hidden layer activation function $g(x)$ selects the non-linear Sigmoid function. The number of genetic algorithm individuals is set to 20, The maximum genetic generation is 100, Chromosome coded as a 10-bit binary number, The input weight and the hidden layer threshold range between -0.5 to +0.5, The generation gap is 0.05, Crossover probability is 0.5, Chromosome mutation probability is 0.01, Observe and record the simulation results of the algorithm as shown in Fig. 2,

From the test set mean square error MSE and the determination coefficient R2, we can see that the GA-ELM algorithm has better prediction set accuracy than the ELM algorithm. The prediction error of the GA-ELM model has stabilized after 40 generations. Experiments on the same data set for multiple simulations. To verify the reliability of the experimental results, Fig. 3 is the result of repeated experiments.

The instability of the ELM algorithm can be seen by comparing Figure 2 and Figure 3.

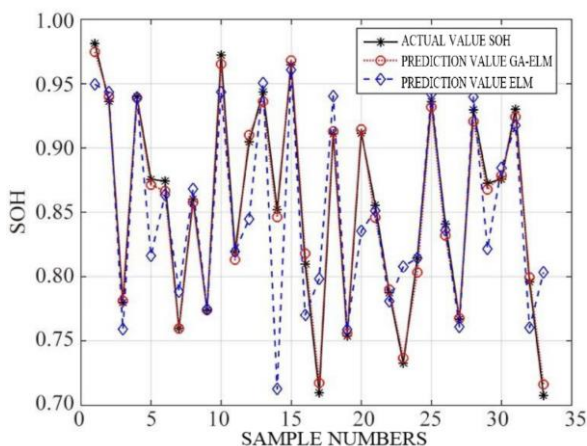


Fig. 2. Comparison of GA-ELM and ELM algorithms

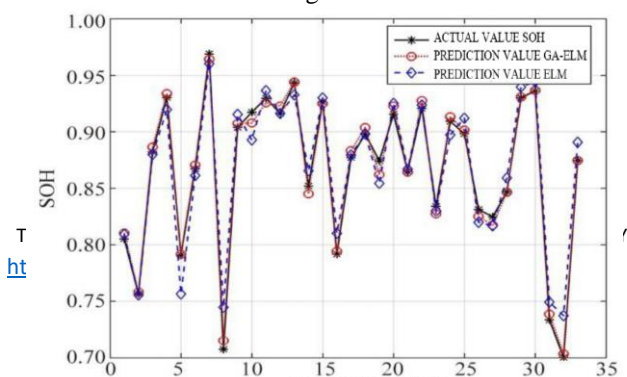


Fig. 3. GA-ELM and ELM algorithm simulation

B. BP algorithm model simulation

In order to compare the prediction performance gap between BP algorithm and ELM algorithm, BP algorithm uses the same network structure as ELM algorithm. The hidden layer transfer function of the BP network also selects the Sigmoid function, the output layer transfer function selects the linear Purelin function, the BP algorithm is optimized by gradient descent method, and the number of reverse iterations is selected 100 times. The simulation result of BP algorithm are shown in Fig. 4.

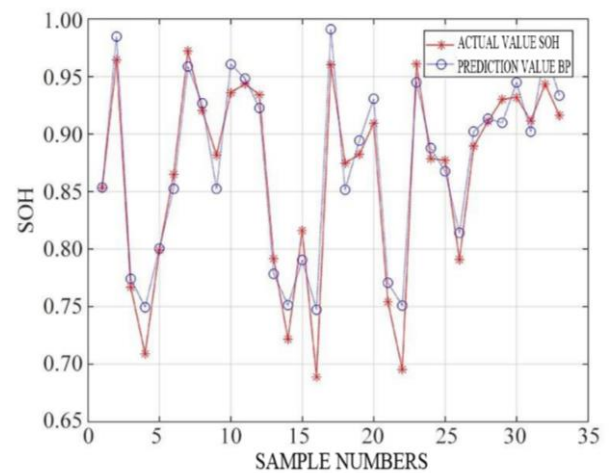


Fig. 4. BP simulation results

Experimental results show that the mean square error of the BP algorithm is equal to 0.015318, the determination coefficient is 0.935, and the algorithm operation cycle is 4.668 s. Under the same network topology, the prediction accuracy and algorithm period of the BP algorithm test set are inferior to the ELM algorithm.

C. RBF and SVR algorithm model simulation

The structure of the RBF algorithm and the SVR algorithm model is different from that of ELM. The accuracy of the RBF algorithm is greatly affected by the spread value of the radial basis function spread velocity. The prediction accuracy is ideal when the model spread = 2. The SVR algorithm needs to set relevant parameters according to the data scale and the type of problem to be solved. The kernel function of the SVR is set to RBF, and the gamma value of the kernel function is obtained by cross-validation method (CV). Gamma = 2.8,

The simulation results of RBF algorithm and SVR algorithm are shown in Fig. 5 and Fig. 6.

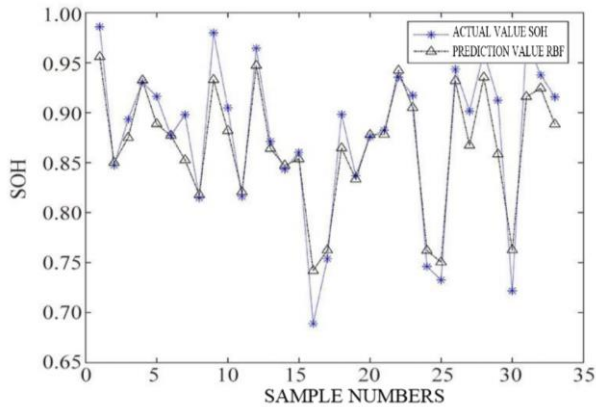


Fig. 5. RBF simulation result

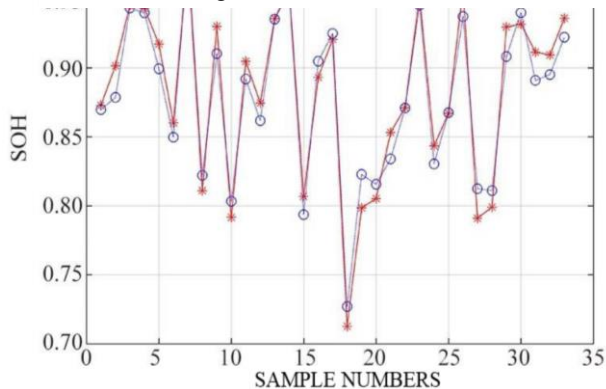


Fig. 6. SVR simulation result

Table-II summarizes the mean square error values Mse , algorithm determination coefficients R^2 , and algorithm operation period of the above algorithms.

It can be seen from Table -II that compared with ELM algorithm, BP algorithm, RBF algorithm, and SVR algorithm, the GA-ELM algorithm has the smallest mean square error value and the highest degree of prediction fit, but the GA-ELM algorithm has the longest running time.

Table-II: Algorithm prediction results data

Algorithm no.	MSE	R^2	Operational time/s
GA-ELM	3.153×10^{-5}	0.993	25.421
ELM	1.040×10^{-3}	0.817	1.149
BP	1.533×10^{-2}	0.937	4.667
RBF	1.145×10^{-3}	0.902	2.453
SVR	1.313×10^{-2}	0.957	6.850

VII. RESULT AND DISCUSSION

In summary, the BP algorithm has a long iteration period. The RBF network weights can be solved by linear equations, but the model prediction accuracy is greatly affected by the spread value, and the algorithm parameter selection is more complicated.

The prediction accuracy of the SVR algorithm is greatly affected by the number of hidden layer units of the model and the activation function selected.

The selection of ELM algorithm parameters is relatively simple. Experiments result show that the GA-ELM algorithm has a better ability to solve regression problems than the ELM and BP algorithms. GA-ELM model is less complex than RBF algorithm and SVR algorithm.

VIII. CONCLUSION

This paper analyzes the advantages of the GA-ELM algorithm over the traditional algorithms. The prediction accuracy and stability of the GA-ELM algorithm are improved compared to the ELM algorithm. The experimental results also confirm the feasibility of the integrated algorithm in practical applications. Genetic algorithms are widely used in the fields of machine learning and signal processing. The characteristics of genetic algorithms are suitable for optimization of large-scale, nonlinear problems. Compared with the particle filter algorithm and GA-SVR algorithm, the GA-ELM algorithm has higher prediction accuracy for the state of health (SOH) of lithium batteries.

The shortcoming of GA-ELM algorithm lies in the long running period of the algorithm, which is caused by the low search efficiency of genetic algorithm. How to reduce the running period of GA-ELM algorithm needs further research.

REFERENCES

- [1]. M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renew. Sustain. Energy Rev.*, vol. 78, pp. 834854, Oct. 2017.
- [2]. H. Farzin, M. Fotuhi-Firuzabad, and M. Moeini-Aghaie, "A practical scheme to involve degradation cost of lithium-ion batteries in vehicle-to-grid applications," *IEEE Trans. Sustain. Energy*, vol. 7, no. 4, pp. 17301738, Oct. 2016.
- [3]. Huang Guangbin, Zhu Qinyu, Siew C H. *Extreme Learning Machine: Theory And Applications*[J]. *Neurocomputing*, 2006(70):489-501.
- [4]. Huang Guangbin, Chen lei. *Enhanced Random Search Based Incremental Extreme Learning Machine*[J]. *Neurocomputing*, 2008, 71(16-18):3460-3468.
- [5]. Chen X, Shen W, Dai M, Cao Z, Jin J, and Kapoor A April 2016 Robust adaptive sliding mode observer using RBF neural network for Lithium-ion battery state of charge estimation in electric vehicles *IEEE Transactions on Vehicular Technology* 65 1936 – 47.
- [6]. J. C. A. Antón, P. J. G. Nieto, C. B. Viejo, and J. A. V. Vilán, "Support vector machines used to estimate the battery state of charge," *IEEE Trans. Power Electron.*, vol. 28, no. 12, pp. 59195926, Dec. 2013.
- [7]. L. Kang, X. Zhao, and J. Ma, "A new neural network model for the state-of-charge estimation in the battery degradation process," *Appl. Energy*, vol. 121, pp. 2027, May 2014.
- [8]. Y. Shen, "Adaptive online state-of-charge determination based on neuro controller and neural network," *Energy Convers. Manage.*, vol. 51, no. 5, pp. 10931098, 2010.
- [9]. H. Chaoui and C. C. Ibe-Ekeocha, "State of charge and state of health estimation for lithium batteries using recurrent neural networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 87738783, Oct. 2017.
- [10]. K. Thirugnanam, T. P. Ezhil Reena Joy, M. Singh, and P. Kumar, "Mathematical modeling of Li-ion battery using genetic algorithm approach for V2G applications,"
- [11]. Susilo, D.D.; Widodo, A.; Prahasto, T.; Nizam, M. State of Health

- Estimation of Lithium-Ion Batteries Based on Combination of Gaussian Distribution Data and Least Squares Support Vector Machines Regression. In *Materials Science Forum*; Trans Tech Publications: Princeton, NJ, USA, 2018; Volume 929, pp. 93–102.
- [12]. Bai, G.; Wang, P.; Hu, C.; Pecht, M. A generic model-free approach for lithium-ion battery health management. *Appl. Energy* 2014, 135, 247–260
- [13]. Saha, B.; Goebel, K. Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, San Diego, CA, USA, 27 September 27–1 October 2009; pp. 2909–2924.
- [14]. G.-B. Huang and C.-K. Siew, "Extreme learning machine: RBF network case," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, vol. 2, pp. 1029–1036, Kunming, China, December 2004.
- [15]. G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1352–1357, 2009.
- [16]. Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.
- [17]. Kim K, Han I (2000) Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl* 19:125–132
- [18]. Hossain D, Capi G (2017) Genetic algorithm based deep learning parameters tuning for robot object recognition and grasping. *Int Sch Sci Res Innov* 11(3):629–633
- [19]. Butcher JB, Verstraeten D, Schrauwen B, Day CR, Haycock PW (2013) Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Netw* 38:76–89
- [20]. Huang G, Siew C (2004) Extreme learning machine: RBF network case. In: *IEEE control, automation, robotics and vision conference*, pp 1–8
- [21]. M. Charkhgard and M. H. Zarif, "Design of adaptive H_{∞} filter for implementing on state-of-charge estimation based on battery state-of-charge-varying modelling," *IET Power Electron.*, vol. 8, no. 10, pp. 1825–1833, 2015.
- [22]. M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad, and A. Ayob, "Neural Network Approach for Estimating State of Charge of Lithium Ion Battery Using Backtracking Search Algorithm," *IEEE Access*, vol. 6, pp. 10069–10079, 2018.
- [23]. E. Chemali, P. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long Short-Term Memory-Networks for Accurate State of Charge Estimation of Li-ion Batteries," *IEEE Trans. Ind. Electron.*, pp. 1–1, 2017.
- [24]. M. A. Awadallah and B. Venkatesh, "Accuracy improvement of SOC estimation in lithium-ion batteries," *J. Energy Storage*, vol. 6, pp. 95–104, 2016.
- [25]. J. Du, Z. Liu, and Y. Wang, "State of charge estimation for Li-ion battery based on model from extreme learning machine," *Control Eng. Pract.*, vol. 26, no. 1, pp. 11–19, 2014.
- [26]. G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006
- [27]. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl Math Comput* 2016; 274: 292–305.
- [28]. Huang, GB, Zhu, QY, Siew, CK. Extreme learning machine: theory and applications. *Neurocomputing* 2006; 70: 489–501.

AUTHORS

First Author – Jamsheer ali graduated from "The University of Faisalabad", Pakistan. Currently I am doing master in electrical engineering from Shaanxi University of science and technology.

Research Interest: Artificial Intelligence, Lithium batteries health estimation, Renewable energy, Maximum power point tracking for PVs.

Email: jamsheer807@gmail.com

Second Author – Shi Yong Sheng is a professor in Shaanxi University of science and technology, China

Research area: Lithium batteries, Turbines and Converters

Third Author – Abdul Rehman graduated from "The University of Faisalabad", Pakistan. Currently doing master in electrical engineering from Harbin University of science and technology.

Fourth Author – Imran Afzal graduated from "The University of Faisalabad", Pakistan. Currently doing master in electronics engineering from Hunan University

Fifth Author – Shoaib ul Hassan got his B.S. degree from department of Computer Science and Information Technology in University of Sargodha, Sub Campus Bhakkar in Pakistan in 2018 and doing his MS degree from Department of Electronic Information and Artificial Intelligence in Shaanxi University of Science and Technology in China. His Research Interest is focused on Internet of things, Android Operating System, Blockchain and Cloud Computing.

