Hindawi
Journal of Electrical and Computer Engineering
Volume 2024, Article ID 9785715, 1 page
https://doi.org/10.1155/2024/9785715

Retraction

Retracted: CAD-Assisted 3D Assembly of Automobile Electrical Switch considering PSO-BP Neural Network Algorithm

Journal of Electrical and Computer Engineering

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] H. Zhang, "CAD-Assisted 3D Assembly of Automobile Electrical Switch considering PSO-BP Neural Network Algorithm," *Journal of Electrical and Computer Engineering*, vol. 2022, Article ID 6454942, 11 pages, 2022.

Hindawi
Journal of Electrical and Computer Engineering
Volume 2022, Article ID 6454942, 11 pages
https://doi.org/10.1155/2022/6454942

Research Article

CAD-Assisted 3D Assembly of Automobile Electrical Switch considering PSO-BP Neural Network Algorithm

Hanting Zhang

Hubei University of Technology, Wuhan 430068, China

Correspondence should be addressed to Hanting Zhang; 1910201408@hbut.edu.cn

Received 24 February 2022; Revised 19 March 2022; Accepted 4 April 2022; Published 5 May 2022

Academic Editor: Xuefeng Shao

The installation of automotive electrical switches is a complex three-dimensional space assembly project which has high requirements for installation accuracy. In order to improve the installation effect of automotive electrical switches, this paper applies the PSO-BP neural network algorithm to automotive electrical switches and integrates PSO and ELM algorithms. The training speed of the ELM model is fast, the model generalizes the data well, and the noise data have little effect on the model. Moreover, this article combines simulation research to evaluate the effect of this algorithm. After confirming the performance of the effect, this paper uses a case study to study the effect of the application of the PSO-BP neural network algorithm to the automotive electrical switch. The research results show that the CAD-assisted 3D assembly system of automobile electrical switch considering PSO-BP neural network algorithm has a good effect.

1. Introduction

As an outstanding engineering technology, CAD technology has been widely used in aerospace, automobiles, ships, machinery, electronics, chemicals, construction, and other industries because of its huge economic benefits. [1] Mechanical automotive products are ever-changing, and users will have the characteristics of their own enterprises when they are applied. Therefore, after users introduce commercial CAD software, they need to carry out different degrees of secondary development on the selected CAD software platform for specific objects. They need to design a CAD system for special car products with a friendly interface and easy to use so that the CAD software can play a full role in the enterprise [2]. In the computer field, with the rapid development of computer hardware performance and interactive graphics technology, people have shifted the focus from program efficiency to user efficiency, and the user interface is a key factor affecting user efficiency. In the development process of the application system, the design of the user interface often reflects the whole idea of the realization of the system function and is the basis for establishing the entire programming framework. Therefore, the user interface design and development work occupies a large proportion of the entire system development process [3].

In the design stage of automotive products, digital assembly modeling is very necessary, and it is an important part of automotive product design. The typical automotive product modeling process in the current commercial 3D CAD design software is as follows: first, the designer interactively specifies the geometric constraint relationship information of the assembly between the parts; then, the 3D software system automatically calculates the position and pose of the parts. The transformation matrix is used to determine its specific position information in the assembly; finally, the coordinate transformation matrix is used to realize the positioning of the part to the assembly position. In the research process of digital assembly technology, the intelligence and knowledge of the assembly system is an urgent problem for researchers to solve [3]. An important way to realize automatic assembly is to use 3D software as a platform to realize auto assembly automation through running programs and reduce the proportion of manual assembly operations when designing auto products. In the process of realizing automatic assembly, how to effectively express and transmit assembly information so that designers

can perform automatic assembly accurately, intuitively, and quickly has always been the research content of domestic and foreign scholars. Foreign research scholars have done a lot of work in assembling navigation. The highlight of the PSO-BP neural network model is the setting of the parameter and weight solution module. The traditional algorithm uses the gradient descent method. Here, a more accurate and fast particle swarm algorithm is used cleverly to enhance the learning ability between individuals and the information between systems. Sharing ability makes the search faster and then obtains the global optimal solution.

This paper combines the PSO-BP neural network algorithm to construct the CAD-assisted three-dimensional assembly system of automotive electrical switches to improve the effectiveness of subsequent automotive electrical switch installation.

2. Related Work

Foreign research on 3D assembly technology began in the 1990s. The United States was the first to research related technologies and made great breakthroughs and cooperated with many research institutions and well-known domestic universities to conduct research on virtual assembly technology. Literature [4] developed a three-dimensional auxiliary assembly system that can assemble products during product design. The system can share data with parametric CAD software. Three-dimensional models and structure trees can be automatically transferred from CAD, product assembly information can be obtained, and assembly sequence and path planning can be carried out according to the obtained data. Literature [5] studies the application of virtual technology in assembly and develops a system that simulates the actual assembly operation through the operation of the workers in the virtual environment on the three-dimensional model of the product. It is used to analyze and establish the assembly sequence of the product and calculate the cost and time of assembly. Literature [6] combined virtual technology with artificial intelligence technology and developed a virtual assembly system called CODY. The biggest feature of this system is the man-machine interactive operation mode. Literature [7] constructed a prototype system called VPW (Virtual Process Week). The main feature of this system is to use virtual technology to test the assembly process of automobiles.

Literature [8] uses virtual technology to establish a three-dimensional assembly planning prototype system called VDVAS. Literature [9] builds a system called VASS (Virtual Assembly Support System) that can implement digital pre-assembly through a three-dimensional digital model of the product in the product design stage. It is very intuitive to verify the geometric feasibility of the product. Literature [10] developed a web-based collaborative assembly process planning system Web CAPS to realize 3D assembly process design and planning in a network environment. Literature [11] used the browser plug-in Cortona to build a VRML (virtual reality modeling language) three-dimensional virtual assembly prototype system. Literature [12] developed a 3D digital assembly process planning system. Literature [13] developed a 3D assembly process design system for internal combustion engine assembly based on the SolidWorks platform. Literature [14] developed a 3D CAD-based visual assembly CAPP system, which can perform 3D assembly process design for gearboxes. The gearbox implements three-dimensional assembly process design. Literature [15] developed a three-dimensional assembly system based on UG (Unigraphics NX) as a development platform by studying the technical difficulties of the three-dimensional assembly process system.

The system constructed in [16] can analyze the assembly information of complex assemblies, focus on the geometry and data structure of the assembly, describe the degree of freedom of the parts, and classify the assembly methods in detail to simplify the complexity of assembly analysis. The virtual constructor CODY in [18] is a knowledge-based interactive assembly workbench on which assembly or disassembly is performed directly using a mouse or similar input device. The user selects an object and moves to another object nearby, and the knowledge-based system will complete the assembly. In addition, users can use natural language to manipulate the system. The research object is standardized reusable parts to construct complex assemblies. The Archimedes system proposed in [18] is a software tool for plan generation and visualization. It can realize the functions of generating, optimizing, verifying, and checking the mechanical assembly sequence of the three-dimensional model. Archimedes was originally used as a vertically integrated system to generate programs from CAD models. Literature [19] studied the data exchange between virtual assembly system and computer-aided design system. The small plane model is evenly cut into three vertical directions, and a contour-based surface model is created. Then, the boundary points are determined from the contour model. The network boundary curve is further derived from the previously determined boundary points from the original VR model of the vertices of the adjacent polygons. The entire surface object of the boundary curve is divided into a single surface area and the topological form of the object. The type is further identified by a single surface.

3. Algorithm Improvement Based on the PSO-BP Neural Network

The scale of the hidden layer unit of the neural network determines the model's ability to map data. The learning ability of the model increases with the increase in the size of the hidden layer unit. When the number of neurons in the hidden layer of the network is fixed, as the number of hidden layers of the network increases, the learning ability of the network model becomes stronger. When the hidden layer of the network is too large, it will cause the model to overlearn the training samples. When the scale of the hidden layer of the network is too small, it will lead to insufficient learning of the training samples by the model. Moreover, the empirical formula for selecting the number of neurons in the hidden layer of the model n_1 is

$$n_1 = \sqrt{m+n} + a. \quad (1)$$

Here, m and n are the number of nodes in the input layer and output layer of the network model, and a is a random

number between 1 and 10. The neural network has a limited capacity to load information, and the BP neural network tends to forget the old samples when training new samples. By increasing the scale of the hidden layer of the network, the information-carrying capacity of the model can be increased. The size of the network will reduce the generalization of the model.

The improvement strategies of the BP algorithm are discussed in the following.

3.1. Additional Momentum. The additional momentum method realizes that when the network parameters are corrected, a correction value proportional to the last parameter is added. The correction process of the neural network parameters is as follows:

$$\begin{aligned}\Delta w_{ij}(t+1) &= (1-m_c)\eta\delta_j u_i + m_c\Delta w_{ij}(t),\\ \Delta\theta_j(t+1) &= (1-m_c)\eta\delta_j + m_c\Delta\theta_j(t).\end{aligned} \tag{2}$$

Here, t represents the training times of the model and m_c is the momentum factor of parameter correction, δ represents the error signal of the model, and w and $\Delta\theta$, respectively, represent the correction amount of the neural network weight and threshold.

The gradient descent of additional momentum does not change the original algorithm mode of gradient descent. The adjustment of network parameters is still in the direction of the gradient of the error function. When the model parameters are adjusted to the flat area of the error function slope and the value of the error signal, the neuron weight and threshold are corrected to $\Delta w_{ij}(t+1) = m_c\Delta w_{ij}(t)$ and $\Delta\theta_j(t+1) = m_c\Delta\theta_j(t)$, respectively. The additional momentum method effectively avoids the risk of parameter optimization falling into local extreme values. However, the gradient descent of the additional momentum is also sensitive to the selection of the initial parameters of the network. When the direction of the model parameter adjustment is consistent with the direction of the global minimum movement, the gradient descent of the additional momentum can work better.

When the artificial neural network model is trained, although the gradient descent method and the gradient descent method with additional momentum are highly adaptable to the model, the convergence speed of the parameters to be optimized is slow, and the training time of the neural network model is longer.

3.2. Adaptive Learning Rate. The selection of the learning rate n value affects the training results of the BP algorithm model. A learning rate that is too small will increase the training time of the model, and a learning rate that is too large will oscillate the convergence of the error function. The adaptive learning rate method can improve this phenomenon. Moreover, the calculation formula of the adaptive learning rate o is as follows:

$$\eta^{t+1} = \eta^t \frac{E^t}{E^{t+1}}. \tag{3}$$

Here, η is the learning rate, t is the number of iterations, and E is the training error of the model.

When the t+1th iteration error E+1 is greater than the tth iteration error E, the learning rate η^{t+1} value of the t+1th iteration decreases. The t+1th iteration error E^{t+1} is less than the tth iteration error. At the iteration error E^t, the learning rate value of the t+1th iteration increases, and the value of the learning rate η^{t+1} is dynamically adjusted according to the model training error.

3.3. Deformation of the Error Function. The error function during model training can be expressed as

$$E = \frac{1}{2}\sum_{y=1}^{P}\sum_{k=1}^{M}\left(d_k^{(p)} - y_k^{(p)}\right)^2. \tag{4}$$

Here, d_k is the actual value of the training sample, and y_k is the predicted value of the model. Baum, Wileze, and others proposed a new model training error function. The conventional error function training is easy to stagnate. The improved model training error function is as follows:

$$E = \sum\left[\frac{1}{2}(1+y_k)\log\left(\frac{1+y_k}{1+d_k}\right) + \frac{1}{2}(1-y_k)\log\left(\frac{1-y_k}{1-d_k}\right)\right]. \tag{5}$$

When the predicted value of the training model is equal to the actual value d of the training sample, the error of model training is 0. When the predicted value 9 of the model is +1 or −1, the function diverges. This overcomes the phenomenon of stagnant model training.

When using the hyperbolic tangent function tanh as the model activation function, that is, f(t) = tanh(t), the error signal during model training changes from $\delta = f(t)(y_k - d_k)$ to $\delta = y_k - d_k$. At this time, the correction of model weight and threshold are, respectively, $\Delta w_{jk} = \eta\delta y_k$ and $\Delta\theta_j = \eta\delta$. Since the adjustment of model parameters lacks the control of f(t), the adjustment process of the parameters is prone to oscillation.

During the foraging process, the ant colony releases "pheromone" to carry out the information interaction between ant individuals, and the individual ants make route selection according to the pheromone concentration of the route to be selected. The pheromone concentration of the path to be selected is inversely proportional to the distance of the ant's foraging path. The ant colony "double-bridge" experiment is shown in Figure 1.

Among them, point B is the colony of ants, point A is the target food, and B-C-D-A or B-E-D-A is an alternative path for foraging. In the early stage of foraging, the pheromone concentrations of the two paths are the same, and the ants have the same path selection probability. In the later stage of foraging, the shorter path B-C-D-A will leave more pheromones. Because the ants make path selection based on the path pheromone concentration, the probability of the ants choosing the shorter path B-C-D-A is greater; that is, the shortest path B-C-D-A is found.

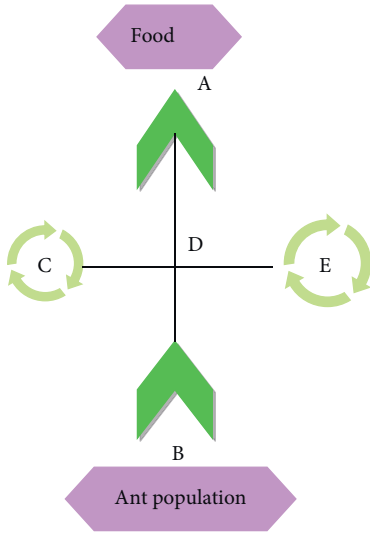


Figure 1: The ant colony "double-bridge" experiment.

Taking Figure 1 as an example, we set b(t) to represent the number of ants on node i at time t, τ_{ij} to represent the pheromone concentration of path i to i, and n to be the cutoff of the problem to be solved. The total amount of ants in the ant colony m is expressed as $m = \sum_{i=1}^{n} b_i(t)$. The taboo table $Tabu_k$ represents all the paths that ant k has traversed, Tabu_k changes dynamically with the path traversed by ant k, and $p_{ij}^k(t)$ represents the path transition probability of ant k from i to j at time t.

$$p_{ij}^k(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum_s \left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}. \tag{6}$$

Here, s represents the candidate path of ant k, and n (t) is the heuristic function of ant state transition, which represents the expected degree of the ant's transition from path i to j. The heuristic function of state transition can be expressed, where d represents the distance from path i to j. a is the information heuristic factor, which indicates the importance of the pheromone concentration to the path selection of ants, and ants tend to choose the path with high pheromone concentration. $ß$ is the expected heuristic factor, which indicates how important the heuristic information is to the ant's path selection. Since the value of the heuristic function is inversely proportional to the distance of the path to be selected, ants are more inclined to choose a shorter path.

Ant colony path optimization is a process based on the positive feedback of path residual pheromone concentration. Excessively high concentration of path pheromone will obscure the role of the heuristic function. The introduction of a pheromone volatilization mechanism can improve the above situation and update the path residual pheromone, and the formula is as follows:

$$\begin{aligned} \tau_{ij}(t+n) &= (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \\ \Delta\tau_{ij}(t) &= \sum_{k=1}^{m} \Delta\tau_{ij}^k. \end{aligned} \tag{7}$$

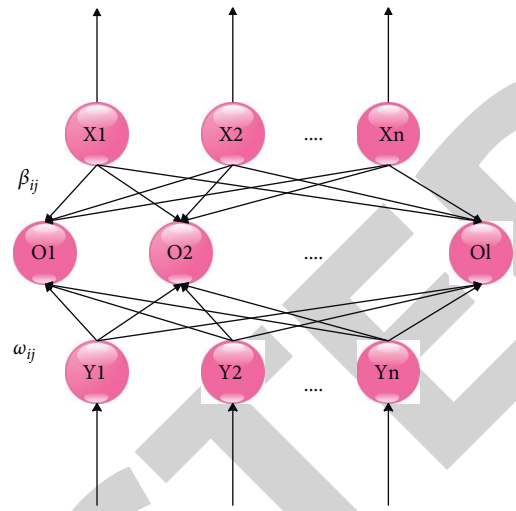


Figure 2: Feedforward neural network model.

Here, ρ is the pheromone volatilization factor, $(1-\rho)$ is the residual factor of the path pheromone, and $\Delta\tau_{ij}$ is the increase in the path pheromone.

The feedforward neural network model is shown in Figure 2. It is composed of an input layer, an output layer, and a hidden layer. The input layer of the model has n neurons, which represent the n input parameters of the model. The output layer of the model has m neurons, which represent the m output parameters of the model. The hidden layer of the model has l neurons, and the model realizes the mapping from input to output through this layer.

w_{ij} is the connection weight of the ith neuron in the input layer and the jth neuron in the hidden layer, and the weight matrix ω is

$$\omega = \begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ \omega_{21} & \omega_{21} & \cdots & \omega_{21} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{l1} & \omega_{l2} & \cdots & \omega_{\text{ln}} \end{bmatrix}_{l\times n}. \tag{8}$$

β_{jk} is the connection weight between the jth neuron in the hidden layer of the model and the kth neuron in the output layer, the weight matrix from the hidden layer to the output layer is β, and b is the threshold value of the lth neuron in the hidden layer of the model.

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1n} \\ \beta_{21} & \beta_{21} & \cdots & \beta_{21} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{l1} & \beta_{l2} & \cdots & \beta_{\text{ln}} \end{bmatrix}_{l\times n} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{bmatrix}_{l\times 1}. \tag{9}$$

We set the number of training samples of the model to Q, and the activation function of the hidden layer unit to g(z), then the output matrix H of the model is

$$H(\omega_1, \omega_2 \ldots \omega_l; b_1, b_2 \ldots b_l; x_1, x_2 \ldots x_l) = \begin{bmatrix} g(\omega_1 x_1 + b_1) & g(\omega_2 x_1 + b_2) & \cdots & g(\omega_l x_1 + b_l) \\ g(\omega_1 x_2 + b_1) & g(\omega_2 x_2 + b_2) & \cdots & g(\omega_l x_2 + b_l) \\ \vdots & \vdots & \vdots & \vdots \\ g(\omega_1 x_Q + b_1) & g(\omega_2 x_Q + b_2) & \cdots & g(\omega_l x_Q + b_l) \end{bmatrix}_{Q \times l}. \tag{10}$$

The target matrix T of the model is

$$T = [\, t_1 \;\; t_2 \;\; \cdots \;\; t_Q \,]_{m \times Q}. \tag{11}$$

Here, there are

$$t_j = \begin{bmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{mj} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{t} \beta_{i1} g(\omega_i x_j + b_i) \\ \sum_{i=1}^{t} \beta_{i2} g(\omega_i x_j + b_i) \\ \vdots \\ \sum_{i=1}^{t} \beta_{im} g(\omega_i x_j + b_i) \end{bmatrix}, j = 1, 2, \ldots, Q, \tag{12}$$

$$\omega_i = [\, \omega_{i1} \;\; \omega_{i2} \;\; \cdots \;\; \omega_{in} \,]; x_j = [\, x_{1j} \;\; x_{2j} \;\; \cdots \;\; x_{nj} \,]^T.$$

In summary, the target matrix T of the single hidden layer feedforward neural network model can be expressed as T' = Hβ, where T is the transpose of T.

The characteristics of a single hidden layer feedforward neural network are as follows:

(1) When the feedforward neural network has N training samples (x_i, t_i), where $i = 1, 2, \ldots, N$, when the number of hidden layer units of the model is N and the activation function g(z) is infinitely differentiable in the solution interval, and when the hidden layer connection weight w and threshold value are randomly assigned, the output matrix H of the hidden layer is invertible and $\|H\beta - T\| = 0$. Among them, β is the output weight matrix of the hidden layer, and T is the target matrix output by the model.

(2) When the feedforward neural network model has N training samples (x_i, t_i), where $i = 1, 2, \ldots, N$, when the number of hidden layer units of the model is K ($K \le N$) and the activation function g(z) is infinitely differentiable in the solution interval, and when the connection weight w and the threshold b of the hidden layer are randomly assigned, we give an arbitrary small error ε and $\varepsilon > 0$, and there is $\|H\beta - T\| < \varepsilon$.

It can be seen from the above conclusion that the feedforward neural network model has N training samples

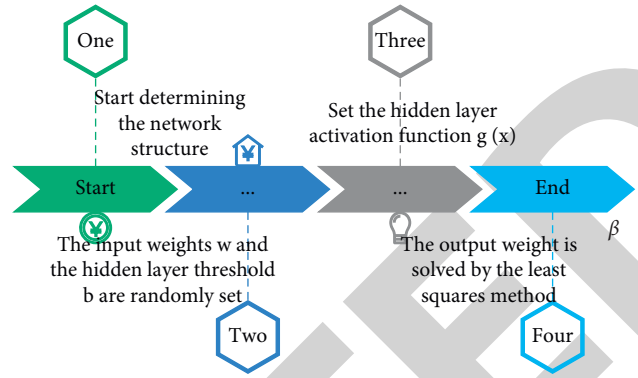


Figure 3: ELM algorithm flow chart.

(x_j, t_j), where $j = 1, 2, 3, \ldots, N$. The number of neurons in the hidden layer of the model is L, the actual output of the training model is o_j, and the minimum error of the model is $\sum_{j=1}^{N} \|O_j - t_j\| = 0$ This needs to calculate the input weight surface of the model, the hidden layer threshold b, and the hidden layer-output weight β and make it satisfy $\|H(\widehat{\omega}_i, \widehat{b}_i), \widehat{\beta}_i - T\| = \min \omega, b, \beta \|H(\omega_i, b_i), \beta_i - T\|$. This is equivalent to minimizing the loss function $E = \sum_{j=1}^{N} \sum_{i=1}^{L} \|\beta_i g(\omega_i, x_j + b_i) - t_j\|$.

Once the connection weight ω_i and hidden layer threshold b_i from the input layer of the extreme learning machine to the hidden layer are determined, the output matrix H of the hidden layer of the model will also be uniquely determined.

The training of the single hidden layer feedforward neural network is transformed into the solution of H$\beta = T$, and the output weight β of the hidden layer of the model is determined from this, that is, $\widehat{\beta}_i = H^+ T$, where H^+ is the Moore–Penrose generalized inverse matrix of the output matrix H.

The training steps of the extreme learning machine model are as follows:

(1) The algorithm determines the model structure: determine the network structure of the input layer, output layer, and hidden layer of the model.

(2) The algorithm assigns the parameters including the connection weight ω from the model input layer to the hidden layer and the hidden layer threshold b randomly assigned values.

(3) The algorithm sets the activation function. The hidden layer of the model sets an activation function g(z) with an infinite differentiable solution interval.

(4) The algorithm calculates the parameter $\widehat{\beta}$. According to the equation $\min_{\beta} \|H\beta - T'\|$, the connection weight $\widehat{\beta}_i = H^+ T'$ from the hidden layer of the model to the output layer is obtained.

The flow chart of the ELM algorithm is shown in Figure 3.

The extreme learning machine model and the BP neural network model are set to the same network structure. The extreme learning machine model has 5 input parameters, 1 output parameter, and 10 hidden layer neurons, and the activation function g(z) of the hidden layer neurons is set to tanh.

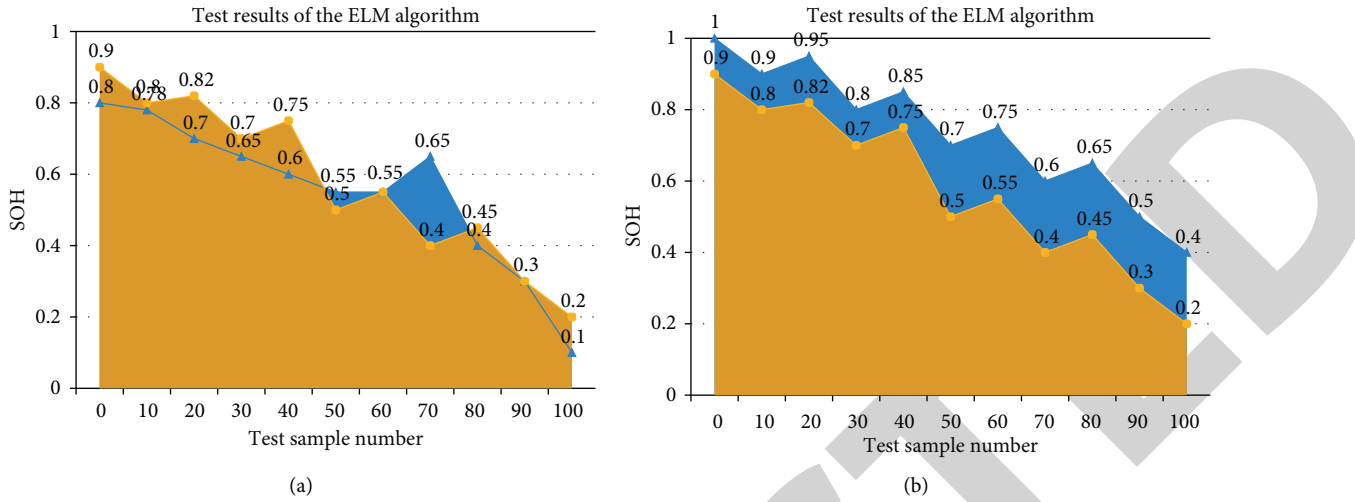


Figure 4: The output result table of the ELM algorithm. (a) The mean square error (MSE) of the prediction result and (b) goodness of fit (R^2).

Table 1: The output error of the ELM algorithm.

	a	B
MSE	0.0005380	0.000834100
R^2	0.9733800	0.9549600

4. Algorithm Performance Test

The two prediction results of the extreme learning machine model are shown in Figure 4, and the mean square error MSE and the goodness of fit R of the prediction results are shown in Table 1.

The training of the extreme learning machine model does not require repeated iterations. The training speed of the extreme learning machine is fast. By comparing the prediction results of the BP, ACO-BP, and ELM algorithm models, it can be seen that the prediction accuracy of the ELM algorithm model is higher than that of the BP and ACO-BP algorithm models. The ELM algorithm improves the long iteration period of the BP algorithm, and the parameter optimization is easy to fall into the local minimum l. As shown in Figures 4(a) and 4(b), although the training and test data used by the model are the same, the prediction results of the model are not completely consistent.

The algorithm changes the fitness of the particles in the solution space by updating the spatial position and velocity of the particles. If the dimension of the problem to be solved is n, $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ is the position vector of particle i, $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ is the velocity vector of particle i, and the update formula of the particle's velocity V and spatial position X is as follows:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(Pbset_i^k - X_i^k\right) + c_2 r_2 \left(Pbset^k - X_i^k\right),$$
$$X_i^{k+1} = X_i^k + V_i^{k+1}. \tag{13}$$

Here, k is the number of iterations of the particle swarm algorithm, w is the inertia weight of the particle velocity update, and X is the position vector of the particle i. $Pbset$ is the optimal position vector of the particle i, $Gbset$ is the optimal position vector of the particle swarm, c_1 and c_2 are the acceleration factors of the particles, and r_1 and r_2 are random numbers between 0 and 1.

In order to prevent particles from arbitrarily searching in the solution space, the particle velocity V and particle position X need to be set in a reasonable interval.

The flow of the algorithm is shown in Figure 5.

Although the training rate and generalization ability of the extreme learning machine model can be better than that of the BP algorithm, the extreme learning machine also has the problem of uncertain hidden layer scale settings. If the number of hidden layer units is set too large, it will cause the model to learn overfitting, and if the number of hidden layer units is set too small, it will cause the model to learn underfitting.

The arbitrariness of the initial parameter setting of the ELM algorithm model leads to the instability of model learning. Because the PSO algorithm has a better global search ability, in view of this defect of the extreme learning machine, the algorithm fusion of PSO and ELM is carried out. This paper uses the PSO algorithm to optimize the ELM model parameters usnxt and be, finds the generation results according to the algorithm, and then trains the ELM torture, as shown in Figure 6.

The inertia weight w is an important parameter of the particle swarm algorithm. The size of the inertia weight value affects the space search ability of particles. A larger inertia weight is conducive to the global search of particles, and a smaller inertia weight is conducive to the local search of particles. Particles need a larger inertia weight in the early stage of the search and a smaller inertia weight in the later stage of the search. The inertia weight of the algorithm is

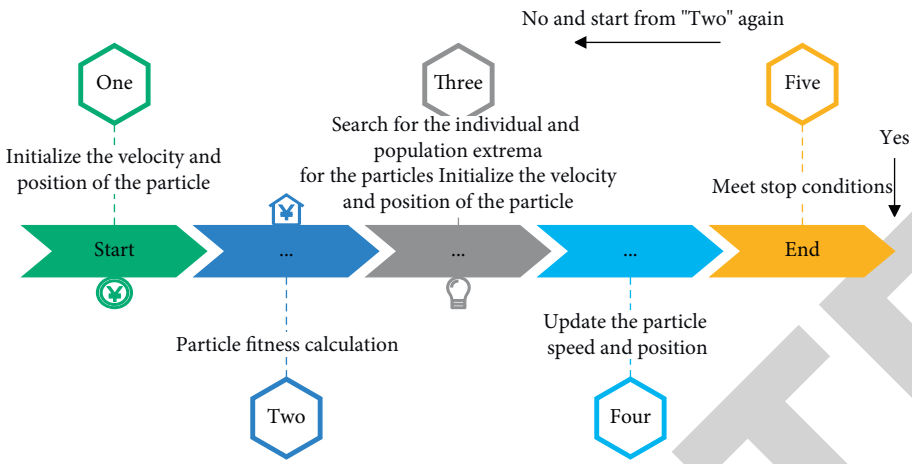


Figure 5: Flow chart of particle swarm algorithm.

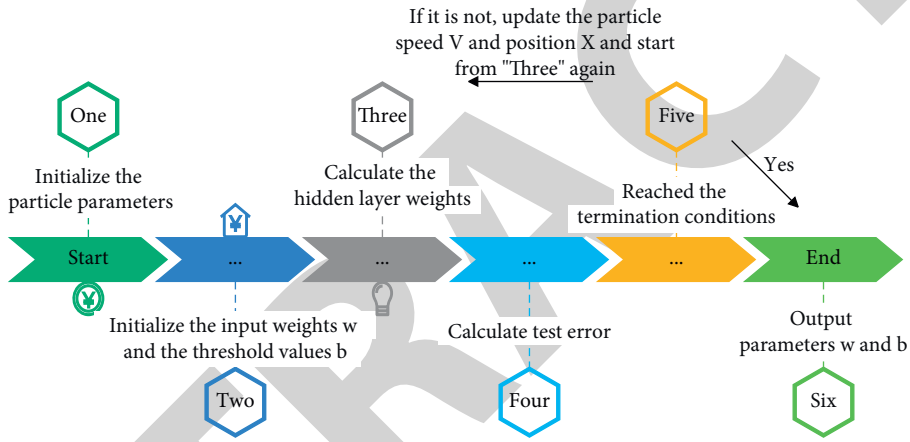


Figure 6: PSO-ELM algorithm flow chart.

Table 2: Parameter settings of particle swarm algorithm.

Number of particles	Maximum number of iterations k_{max}	Value range of inertia weight	Acceleration factors c_1 and c_2	Parameter division range	Maximum particle velocity V_{max}
10	50	[0.4, 0.9]	2	[−1, 1]	0.3

updated in a linearly decreasing manner, and the update formula of the inertia weight is as follows:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}}, k. \quad (14)$$

Here, ω_{max} and ω_{min} are the maximum and minimum values of the inertia weight, k_{max} is the maximum number of iterations of the algorithm, and k is the current number of iterations of the algorithm. The value range of the model parameters of the particle swarm algorithm is $[X_{min}, X_{max}]$, the acceleration factors are c_1 and c_2, and the maximum velocity of the particles is V_{max}. The parameter settings of the particle swarm algorithm are shown in Table 2.

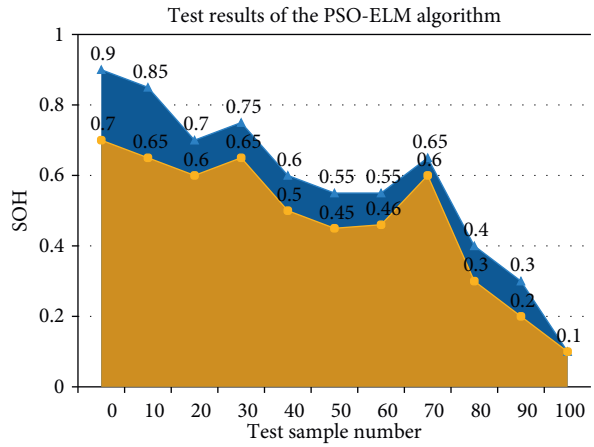
The prediction results of the PSO-ELM algorithm are shown in Figure 7.

Figure 7(a) is the prediction output of the PSO-ELM algorithm, the mean error of the model prediction is 0.0000112, and the goodness of fit of the model output is 0.99848. Figure 7(b) shows the iterative error of the algorithm, which stabilizes after 40 iterations.

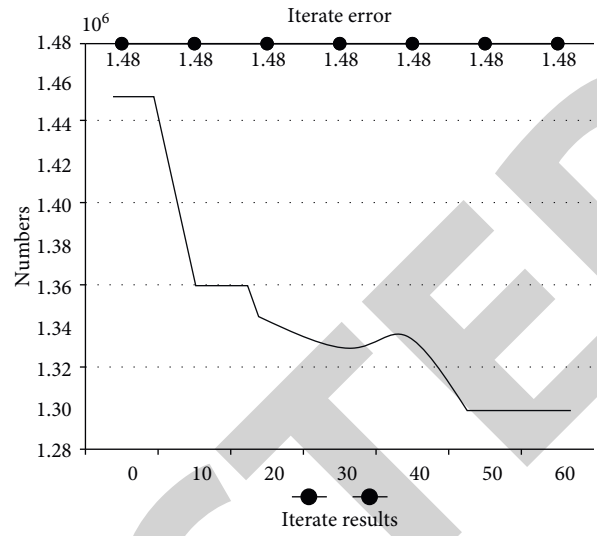
The main indicator of the particle swarm algorithm is the number of particles, which determines the efficiency of the particle search in space. Different particle numbers are set for the PSO algorithm to explore the impact of the particle number on the PSO-ELM model.

The parameter settings of the modified particle swarm algorithm are shown in Table 3.

Figures 8 and 9 show the iterative results of the PSO-ELM model when the number of particles is 50 and 2, respectively. Compared with the model iteration when the number of particles is 10, it can be seen that increasing the number of particles in the algorithm can effectively improve the prediction accuracy of the model. When the number of particles increases to 50, the prediction accuracy of the model still has a



(a)



(b)

Figure 7: The prediction results of the PSO-ELM algorithm. (a) Test results of PSO-ELM algorithm and (b) iteration error.

Table 3: The parameter settings of the modified particle swarm algorithm.

Number of particles	Maximum number of iterations k_{max}	Value range of inertia weight	Acceleration factors c_1 and c_2	Parameter division range	Maximum particle velocity V_{max}
50	50	[0.4, 0.9]	2	[−1, 1]	0.3
2	50	[0.4, 0.9]	2	[−1, 1]	0.3

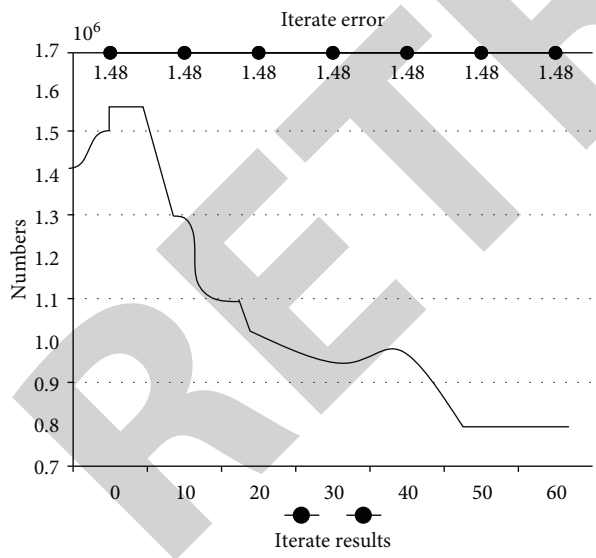


Figure 8: Iterative results of the PSO-ELM algorithm model (par-50).

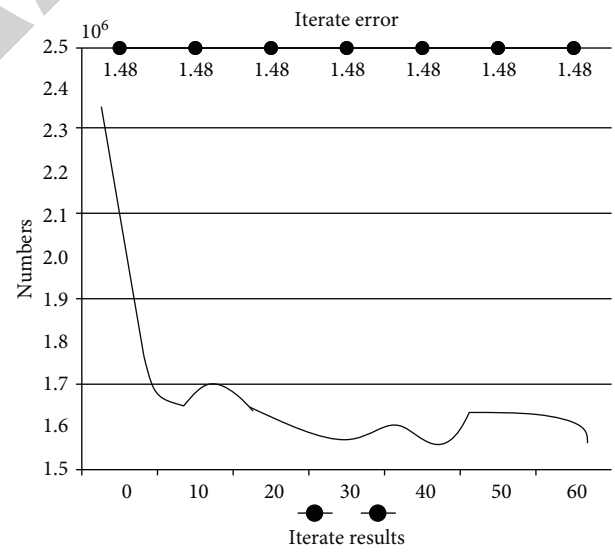


Figure 9: Iterative results of the PSO-ELM algorithm model (par-2).

large room for improvement. When the number of particles decreases to 2, the model quickly converges to a larger value.

Table 4 shows the time required for the PSO-ELM algorithm model to complete 50 iterations. Compared with the ACO-ELM algorithm model, the iteration cycle (time) of the algorithm model is significantly shorter, and the iteration cycle has an approximately linear relationship with the number of particles.

The particle swarm algorithm is a simple heuristic algorithm. Both the particle swarm algorithm and the genetic algorithm belong to the bionic optimization algorithm. The parameter setting of the particle swarm algorithm is easier than that of the genetic algorithm, and the iteration efficiency of the particle swarm algorithm is higher than that of the genetic algorithm. The particle swarm optimization BP neural network (PSO-BP model) realizes the detection of the health state of the electric switch of the automobile. The experimental data of the article are selected from NASA's public data set, and the relative error of the model prediction is about

Table 4: Iteration cycle statistics of PSO-ELM algorithm.

Number of particles	Model training cycle (s)
50	23
2	5.2
10	11.19

Table 5: Performance comparison between prediction models for auxiliary installation of automotive electrical switches.

Algorithm model	Mean square error (MSE)	Model training time (s)
SVR	1.480×10^{-3}	8.90
GPR	1.530×10^{-4}	10.40
BP	7.060×10^{-2}	13.20
ACO-BP	9.970×10^{-4}	120.50
ELM	5.390×10^{-4}	3.60
GA-ELM	3.160×10^{-5}	25.70
PSO-ELM	1.130×10^{-5}	23.10

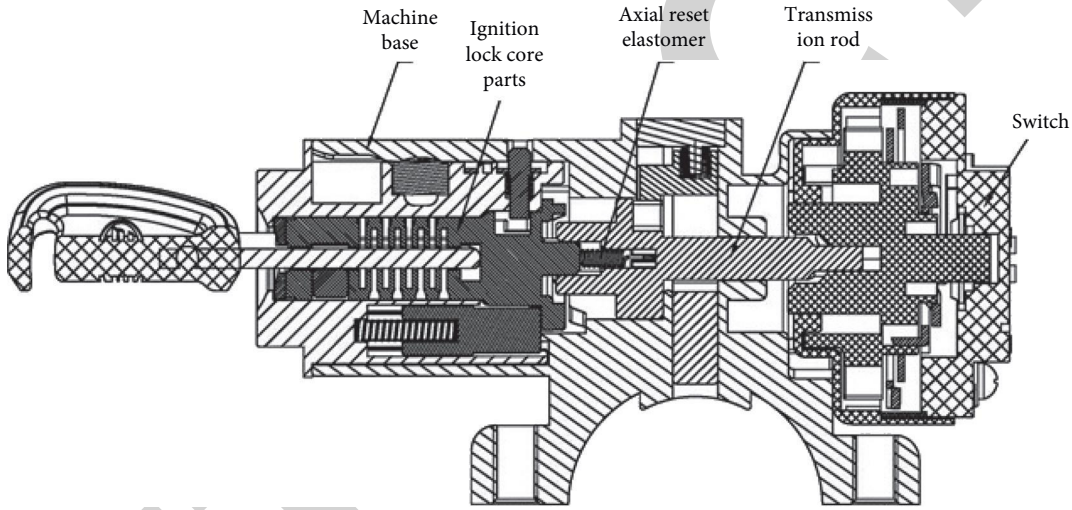


Figure 10: Automotive electrical switch.

0.025. In this paper, only the average temperature and voltage are used as the input of the model, and other parameters of the automobile electrical switch are not analyzed. By constructing a prediction model for auxiliary installation of automotive electrical switches based on SVR and GPR algorithms, the prediction accuracy MSE of the model in the test set is 0.0014 and 0.00015, respectively, which are lower than the prediction accuracy of the fusion algorithm model used in this paper. Table 5 shows the performance comparison between different algorithm models of the auxiliary installation prediction of automobile electrical switches.

5. CAD-Assisted 3D Assembly of Automobile Electrical Switch considering PSO-BP Neural Network Algorithm

This article takes the structure shown in Figure 10 as the research object and explores the effect of CAD-assisted 3D assembly of automobile electrical switches taking into account the PSO-BP neural network algorithm. Through the algorithm of this paper, the simulation is carried out, the design effect of the process and the installation effect are counted, and the results are shown in Tables 6 and 7.

Table 6: Process design effect.

No.	Process design	No.	Process design	No.	Process design
1	94.76	13	94.96	25	95.99
2	94.71	14	96.94	26	91.94
3	94.48	15	93.91	27	95.80
4	95.17	16	93.06	28	95.30
5	91.45	17	94.11	29	94.57
6	95.05	18	91.32	30	94.92
7	95.10	19	92.55	31	95.15
8	91.91	20	95.16	32	91.02
9	91.41	21	95.02	33	96.35
10	93.47	22	91.58	34	94.52
11	93.31	23	95.36	35	91.35
12	92.07	24	91.20	36	92.19

From the above results, it can be seen that the CAD-assisted 3D assembly system of automobile electrical switches taking into account the PSO-BP neural network algorithm has good effects.

Table 7: Auxiliary installation effect.

No.	Auxiliary installation	No.	Auxiliary installation	No.	Auxiliary installation
1	91.77	13	93.87	25	90.36
2	88.93	14	89.07	26	93.65
3	91.16	15	93.13	27	90.42
4	88.39	16	92.31	28	92.84
5	90.47	17	89.83	29	92.45
6	88.88	18	90.41	30	88.50
7	91.81	19	90.37	31	88.85
8	92.19	20	90.38	32	93.26
9	89.41	21	93.99	33	93.04
10	93.50	22	93.58	34	89.59
11	89.10	23	93.44	35	90.55
12	90.61	24	92.62	36	92.75

6. Conclusion

Automobile electrical switches are used in large quantities and have many varieties, which require sensitive and reliable work, and can adapt to harsh working environments such as large temperature differences and large vibrations. As long as the relationship between the components is defined as insertion or merging, the simulated three-dimensional assembly can be completed, and it can be used for interference checking of related components and analysis and calculation of mass and inertia. The application of computer-aided design can give full play to the graphics processing functions of the computer. By calling the prebuilt electrical switch parts library, the three-dimensional simulation assembly of automobile electrical switches can be realized so that the assembly process and realistic effects of the product can be seen in the design stage, which reduce the need for repeated trial production of the product and greatly improve the design quality and efficiency of automotive electrical switches. This paper combines the PSO-BP neural network algorithm to construct the CAD-assisted 3D assembly system of automotive electrical switches to improve the effectiveness of subsequent automotive electrical switch installation. [20].

Data Availability

The labeled data set used to support the findings of this study is available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This study was sponsored by Hubei University of Technology.

References

[1] G. Xie, Z. Gang, L. Yan, Z. Jia, R. Li, and K. Li, "Fast functional safety verification for distributed automotive applications during early design phase," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 99, p. 1, 2017.

[2] X. Sun, J. Liu, B. Lu, P. Zhang, and M. Zhao, "Life cycle assessment-based selection of a sustainable lightweight automotive engine hood design," *International Journal of Life Cycle Assessment*, vol. 22, no. 9, pp. 1373–1383, 2017.

[3] G. Mostyn, "Meeting advanced automotive design challenges with mems-based timing devices," *Electronics World*, vol. 124, pp. 20–23, 2018.

[4] B. R. Diez-Caballero, J. Alfonso-Beltran, I. J. Bautista, and C. B. Pitarque, "Occupational risk factors for shoulder chronic tendinous pathology in the Spanish automotive manufacturing sector: a case-control study," *BMC Musculoskeletal Disorders*, vol. 21, no. 818, pp. 1–8, 2020.

[5] P. Polverino, E. Frisk, D. Jung, M. Krysander, and C. Pianese, "Model-based diagnosis through structural analysis and causal computation for automotive polymer electrolyte membrane fuel cell systems," *Journal of Power Sources*, vol. 357, pp. 26–40, 2018.

[6] X. Wei, H. Yuan, H. Wang, and Y. Chen, "Intelligent design for automotive interior trim structures based on knowledge rule-based reasoning," *International Journal of Automotive Technology*, vol. 21, no. 5, pp. 1149–1167, 2020.

[7] R. A. Poshekhonov, G. A. Arutyunyan, S. A. Pankratov, A. S. Osipkov, D. O. Onishchenko, and A. I. Leontyev, "Development of a mathematical model for optimizing the design of an automotive thermoelectric generator taking into account the influence of its hydraulic resistance on the engine power," *Semiconductors*, vol. 51, no. 8, pp. 981–985, 2017.

[8] U. Faruk, "Phase-coded fmcw automotive radar: system design and interference mitigation," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 270–281, 2019.

[9] N. Geren, O. O. Akçalı, and M. Bayramoğlu, "Parametric design of automotive ball joint based on variable design methodology using knowledge and feature-based computer assisted 3d modelling," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 87–103, 2017.

[10] M. Mcharek, T. Azib, M. Hammadi, C. Larouci, and J.-Y. Choley, "Multiphysical design approach for automotive electronic throttle body," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6752–6761, 2020.

[11] Y. Xie, G. Zeng, R. Kurachi, X. Peng, and G. Xie, "Balancing bandwidth utilization and interrupts: two heuristic algorithms for the optimized design of automotive cps," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2382–2392, 2020.

[12] J. P. Trovao, "Digital transformation, systemic design, and automotive electronics [automotive electronics," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 149–159, 2020.

[13] K. Rohde-Brandenburger and C. Koffler, "Commentary on 'correction to: on the calculation of fuel savings through lightweight design in automotive life cycle assessments' by koffler and rohde-brandeburger (2018)," *International Journal of Life Cycle Assessment*, vol. 24, no. 3, pp. 397–399, 2019.

[14] E. D. Pasquale and D. Coutellier, "Wave propagation in head on bonnet impact: material and design issues," *International Journal of Automotive Technology*, vol. 18, no. 4, pp. 631–642, 2017.

[15] A. Nordelof, E. Grunditz, A. M. Tillman, T. Thiringer, and M. Alatalo, "A scalable life cycle inventory of an electrical automotive traction machine—part i: design and composition," *International Journal of Life Cycle Assessment*, vol. 23, no. 1, pp. 55–69, 2018.

[16] D. Xiao, C. Liu, and B. Tuan Le, *Detection Method of TFe Content of Iron Ore Based on Visible-Infrared Spectroscopy and IPSO-TELM Neural network*, Infrared Physics & Technology, vol. 97, no. 16, pp. 341–348, 2019.

[17] S. Kongwat, P. Jongpradist, and H. Hasegawa, "Lightweight bus body design and optimization for rollover crashworthiness," *International Journal of Automotive Technology*, vol. 21, no. 4, pp. 981–991, 2020.

[18] N. Yusof, S. M. Sapuan, M. Sultan, and M. Jawaid, "Conceptual design of oil palm fibre reinforced polymer hybrid composite automotive crash box using integrated approach," *Journal of Central South University*, vol. 27, no. 1, pp. 64–75, 2020.

[19] S. M. Hosseini, M. A. Rad, A. Khalkhali, and M. J. Saranjam, "Optimal design of the s-rail family for an automotive platform with novel modifications on the product-family optimization process," *Thin-Walled Structures*, vol. 138, pp. 143–154, 2019.

[20] I. J. C. Perez Olguin and J. Iván, "High power printed circuit board design for automotive fuse block usage," *Dyna*, vol. 84, no. 203, pp. 88–94, 2017.