

## Research Article

# Development and Optimization for a New Planar Spring Using Finite Element Method, Deep Feedforward Neural Networks, and Water Cycle Algorithm

Ngoc Le Chau,<sup>1</sup> Hieu Giang Le,<sup>1</sup> Van Anh Dang,<sup>2</sup> and Thanh-Phong Dao <sup>3,4</sup>

<sup>1</sup>Faculty of Mechanical Engineering, Ho Chi Minh University City of Technology and Education, Ho Chi Minh City, Vietnam

<sup>2</sup>Faculty of Mechanical Engineering, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

<sup>3</sup>Division of Computational Mechatronics, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam

<sup>4</sup>Faculty of Electrical & Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

Correspondence should be addressed to Thanh-Phong Dao; [daothanhphong@tdtu.edu.vn](mailto:daothanhphong@tdtu.edu.vn)

Received 17 March 2021; Revised 5 May 2021; Accepted 31 May 2021; Published 14 June 2021

Academic Editor: Noorbakhsh Amiri Golilarz

Copyright © 2021 Ngoc Le Chau et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The gravity balance mechanism plays a vital role in maintaining the equilibrium for robots and assistive devices. The purpose of this paper was to optimize the geometry of a planar spring, which is an essential element of the gravity balance mechanism. To implement the optimization process, a hybrid method is proposed by combining the finite element method, the deep feedforward neural network, and the water cycle algorithm. Firstly, datasets are collected using the finite element method with a full experiment design. Secondly, the output datasets are normalized to eliminate the effects of the difference of units. Thirdly, the deep feedforward neural network is then employed to build the approximate models for the strain energy, deformation, and stress of the planar spring. Finally, the water cycle algorithm is used to optimize the dimensions of the planar spring. The results found that the optimal geometries of the spring include the length of 45 mm, the thickness of 1.029 mm, the width of 9 mm, and the radius of 0.3 mm. Besides, the predicted results determined that the strain energy, the deformation, and the stress are 0.01123 mJ, 33.666 mm, and 79.050 MPa, respectively. The errors between the predicted result and the verifying results for the strain energy, the deformation, and the stress are about 1.87%, 1.69%, and 3.06%, respectively.

## 1. Introduction

A device is balanced when it can maintain equilibrium in any configuration or position without the need for external forces or actuators [1]. Generally, when a device achieves a gravity balance, it can move like in a gravity-less environment [2, 3]. For a device to reach equilibrium, scientists have researched balancers to compensate for the gravity of machines. Gravity balances can be classified in the following ways: active balance and passive balance, and gravity balancer mechanisms can adjust loads or cannot adjust loads [4]. Besides, some other mechanisms employ counterweights, springs, or coupling of counterweights and springs [2]. Among them, a gravity balance mechanism with springs is of great interest for research by scientists because of its

suitability in applications and simple adjustment [5, 6]. These types of mechanisms are capable of adjusting loads via using traditional springs [7, 8].

Previously, a gravity balance mechanism with adjusted loads was designed by the combination of a compliant spring and a torsion spring [9]. Besides, a compliant rotary joint for the gravity balance mechanism was designed by creating a torque that balances the torque of the load [10]. The spring and the joint were built based on advantages of compliant mechanisms; however, the stiffness of the spring is not high [11].

Although scientists have had great success in studying many of the different types of mechanisms, a large amount of energy is still required during the adjustment process. Therefore, the present study proposes a new gravity balance

mechanism according to the principle of compliant mechanisms. Compliant mechanism is selected to create the planar spring (PS) because it is a monolithic mechanism, which offers less lubricant, no friction, and minimal cost of manufacturing [12]. The gravity balance mechanism is a combination of a torsion spring and a planar spring. This combination allows the spring to easily adjust its stiffness when adjusting the load. In the proposed balancing mechanism, the PS is a very important detail. The PS is deformed during the working process, and it creates a strain energy and elastic force to compensate for the gravity generated by the mass. In order to work well, the strain energy of the PS should offer as large as possible and the stiffness of the PS should be ensured to create an elastic force that balances with the gravity due to the mass. In addition, the stress concentration must be less than the permissible stress.

In order to serve for a practical application, the PS should concern a full set of performances, including deformation, strain energy, and stress. However, these desirable properties have mutual contradictions, and these properties are very sensitive to PS geometrical dimensions. Therefore, optimization of the PS is necessary to balance its properties. In this study, an optimization process is performed to maximize or minimize one or more properties of the PS. It is noted that the proposed PS is designed based on the principle of compliant mechanisms. It is therefore difficult to build precisely mathematical models that show the relationship between deformation, stress, and strain energy with its geometrical factors. Therefore, in this study, a hybrid method of finite element method (FEM), deep feedforward neural network (DFNN), and water cycle algorithm (WCA) is aimed to build surrogate models and optimize the geometry of the PS.

Nowadays, the FEM is a widely used method to solve complex arithmetic problems [13, 14]. This method is capable of accurately predicting the phenomena and behavior of complex problems with little cost and time. Until now, the FEM has been used effectively to simulate the behavior of compliant mechanism [15, 16], healthy human intervertebral discs [17], and double-row tapered roller bearing [18]. Therefore, this study adopts the FEM to predict the behaviors of the PS. The DFNN is extended to model the performances [19]. The DFNN has the ability to approximate the relationships between the independent input variables to the output responses of highly nonlinearly complex models. However, the accuracy of the approximate model depends greatly on the structure of the DFNN [20]. Therefore, before building the approximate models, the structure optimization of the DFNN needs to be performed. This process is to find the training function, the number of hidden layers, the number of nodes in each hidden layer, and the data division rate that best matches the dataset. The proposed PS is desired to achieve the maximum strain energy but must ensure the required stiffness and durability when working. This is an optimization problem with many constraints. The WCA is capable of solving the optimization with multiple constraints [21, 22]. Therefore, the WCA is used to optimize the geometry of the PS. Although there have been studies on many

different gravity balance mechanisms, a full consideration of the performances of the spring has not been considered yet.

In summary, the present paper is aimed to optimize the geometry factors of a planar spring, which is used for the gravity compensation mechanism. To perform the optimization process, the FEM method is used to simulate and collect data. Next, regression models are built by the DFNN. Before using the DFNN, the structure of the DFNN is selected by optimizing the controllable parameters of the DFNN by using the Taguchi method. Finally, the WCA is applied to optimize the geometry of the planar spring.

## 2. Mechanical Design

*2.1. Design of Gravity Balancing Mechanism.* Figure 1 shows a design of a gravity balance mechanism. It consists of a bar that carries an object. The bar is connected to the planar spring ( $k_2$ ) through the pulley and cable. The planar spring is combined with the pulley and cable to create a zero-free-length spring. At the hinge joint, a torsion spring with stiffness  $k_1$  is considered as a rotation spring. During the working process, when the bar rotates clockwise by angle  $\varphi$ , the torsion spring is also deformed by an angle  $\varphi$ , and the planar spring is stretched by an amount equal to  $l$ .

Then, the moment balance equation of the mechanical system is expressed as follows:

$$(0.5m_1 + m_2)Lg \sin \varphi = k_1\varphi + k_2l, \quad (1)$$

where  $m_1$  is the mass of the bar,  $m_2$  is the mass of the object,  $L$  is the length of the bar,  $k_1$  is the stiffness of torsion spring,  $k_2$  is the stiffness of a spring of zero free length, and  $l$  is deformation of a spring of zero free length.

$$l = 2a^2 \sin \varphi, \quad (2)$$

where  $a$  is the distance from the rotating center to the connection point of a spring of zero free length.

In the proposed design, the bar is made of aluminum alloy with  $m_1 = 0.5$  kg, the mass  $m_2$  of the object allows to be adjusted in the range from 0.5 kg to 3 kg. The distance  $a$  is 65 mm and the length  $L$  of the bar is 400 mm. To ensure working requirements, the stiffness of the torsion spring is selected as 200 N/mm and the stiffness of the planar spring is adjustable in the range from 0.325 N/mm to 1.468 N/mm. Based on the required stiffness, a planar spring is designed in Section 2.2.

*2.2. Design Planar Spring.* To ensure an adjustable stiffness in the range of 0.325 N/mm to 1.468 N/mm, deformation of the spring is required from 0 to 30 degrees, and the planar spring is designed with 31 component springs that are connected in series, as shown in Figure 2. The component springs are formed by arranging the leaf springs in parallel and series styles, as shown in Figure 3. The geometrical parameters of the PS include the thickness  $t$ , length  $L$ , width  $w$ , radius  $r$ , height  $H$ , and width  $a$ .

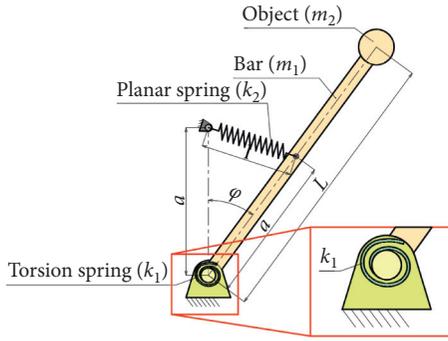


FIGURE 1: Structure of gravity balance mechanism.

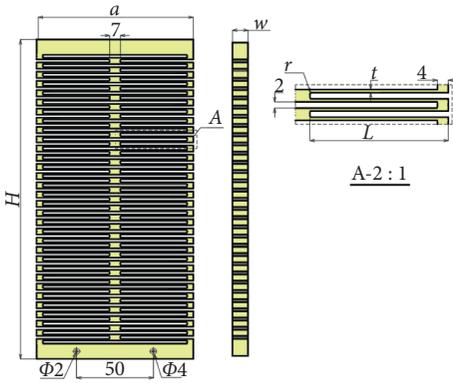


FIGURE 2: Structure of planar spring.

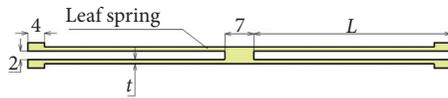


FIGURE 3: Structure of component spring.

### 3. Formulation of Optimization Problem

In order to offer an efficient work for the gravity balance mechanism, the performances of the PS should be enhanced because the efficiency of the overall mechanism is highly dependent on the properties of the PS. Meanwhile, the performances of the PS are very sensitive to the geometrical dimensions of the leaf springs, as given in Figures 2 and 3. The main design parameters include the length  $L$ , width  $w$ , thickness  $t$ , and radius  $r$  of the leaf spring. Therefore, if there is an expectation to improve the characteristics of the spring, the geometry of the leaf springs should be optimized. In this gravity balance mechanism, the spring should be met the following requirements:

- (i) The generated deformation has to be large enough so that the bar rotates at an angle from 0 to 30 degrees, and creates a moment for balancing.
- (ii) The generated stress must be less than the yield strength of the material. In this study, a maximum

load of 3 kg and a stiffness of planar spring of 1.436 N/mm are utilized.

- (iii) The strain energy is as large as possible so that the PS can do highly efficient work.
- (iv) The fatigue should also be considered. It is remarked that the performances of the PS often contradict each other. Therefore, the question arises as to how to balance these properties. So, the WCA optimization algorithm is applied to solve the optimization with multiple constraints.

**3.1. Design Variables.** Based on the proposed design in Figures 2 and 3, the deformation, stress, and strain energy of the PS depend on the leaf springs' properties. The leaf spring's properties are very sensitive to length  $L$ , width  $w$ , thickness  $t$ , and radius  $r$ . Therefore, these dimensions are chosen as the design variables. The limited dimensions of the design variables are selected based on the machining technology capabilities, the desired size of the system, and the designer's experience. These are represented as follows:

$$\begin{cases} 45 \text{ mm} \leq L \leq 55 \text{ mm}, \\ 1 \text{ mm} \leq t \leq 1.4 \text{ mm}, \\ 9 \text{ mm} \leq w \leq 14 \text{ mm}, \\ 0.1 \text{ mm} \leq r \leq 0.3 \text{ mm}. \end{cases} \quad (3)$$

**3.2. Objective Function.** As stated in the formulation of optimization problem above, the PS should possess a large strain energy to store and release a good elastic deformation. Therefore, the strain energy  $f_1(\mathbf{X})$  is chosen as the objective function of the optimization process.

**3.3. Constraint Functions.** In order to ensure the gravity balance mechanism safely, the PS must create a large enough deformation so that the mechanism can work in the range of 0–30 degrees. In addition, the deformation of the PS should be sufficient to create a moment that is equal to the moment generated by the mass. In addition, the stress of the PS should be less than the yield strength when the mechanism is operating at maximum load to ensure that the spring works safely. Hence, the deformation  $g_1(\mathbf{X})$  and stress  $g_2(\mathbf{X})$  are selected as the constraints of the optimization process.

The optimization problem for the PS is briefly stated as follows:

$$\text{Find: } \mathbf{X} = [r, l, t, w]$$

$$\max f_1(\mathbf{X}). \quad (4)$$

Subject to constraints,

$$\begin{cases} g_1(\mathbf{X}) = 33.646 \text{ mm}, \\ g_2(\mathbf{X}) \leq 120 \text{ MPa}. \end{cases} \quad (5)$$

## 4. Proposed Hybrid Optimization Approach

As discussed in Section 3, in order to ensure the efficient working of the gravity balancing mechanism, the PS must have three properties (deformation, stress, and strain energy). However, these properties have mutual contradiction. Hence, a hybridization of experiment design, FEM, DFNN, and WCA is developed. First, the experimental design is built by full factor experiment for collecting data. Then, the DFNN was used to establish the approximate models. Before establishing an approximate model, the structure of the DFNN needs to be optimized to get the best structure of the DFNN. Finally, the WCA is employed to optimize the geometrical sizes of the spring. This method is carried out through 4 stages, as shown in Figure 4.

*4.1. Stage 1: Mechanical Design.* The mechanical design process is performed by the following steps:

Step 1: identify the problem

When the working load is changed, the balance condition will be broken. Besides, when the load is adjusted, the gravity balance mechanism needs to be changed to maintain the equilibrium condition. One of the popular methods for adjusting the gravity balance mechanism is an adjustment of the stiffness of the spring. In this study, the proposed planar spring is designed to adjust easily the number of active leaf springs. So, the stiffness of PS can be adjusted without extra energy. To meet work requirements of the gravity balance mechanism, the design and optimization of the PS is essential.

Step 2: original design

Based on the requirements of the problem, the original gravity balance mechanism is designed, as shown in Figure 1. The principle of this mechanism is presented in Section 2.1. This study proposes the structure of PS as shown in Figure 2.

Step 3: define design variables, objective functions, and constraints

The PS is designed to meet technical requirements. A deformation must be large enough, the stress must be less than the yield strength, and the strain energy is large. The objective function, constraint, and design variable are presented in Section 3.

*4.2. Stage 2: General Numerical Data.* The purpose of this stage is to create datasets to build the approximate models, and the datasets were generated by the numerical simulation in ANSYS 18.2 software. The sequence of steps is as follows:

Step 4: experiment design

Experimental design is the statistical technique that is widely used in product development. To accurately evaluate the effect of each design variable on the performance of a product, the full factor experiment is used.

Step 5: simulate and collect data

Simulation is a technique of predicting the behavior of the structure. It is applied in engineering to reduce the costs of experimentation. A 3D model is built, and the boundary conditions and load are set up. The simulation process is performed to collect data.

Step 6: normalize data

The properties of a planar spring have different units; to avoid the effect of the different units on the optimal result, the output response should be normalized. This normalization only changes the value of the response, but does not change the nature of the data. Moreover, this process can evaluate fairly. In this study, the properties of PS need to be standardized to unify the units. The standardization process is performed by the following formula:

$$z_i = \frac{y_i - \min(y_i)}{\max(y_i) - \min(y_i)}, \quad (6)$$

where  $z_i$  is the normalized value and  $y_i$  is the real value.

*4.3. Stage 3: Optimize the Structure of the DFNN.* To approximate the relationship between the geometry sizes and the output characteristics of the PS, the DFNN is applied to create regression models. The working diagram of the DFNN is shown in Figure 5. Net input ( $z_i^j$ ) of node  $j_{th}$  of layer  $i_{th}$  is calculated according to the following formula:

$$z_i^j = \sum_{i=1}^n (w_{i,j}^j a_{i-1}^j) + b_i^j, \quad (7)$$

where  $w_i^j$  is the weighted input of the node  $j_{th}$  of the hidden layer  $i_{th}$ ,  $b_i^j$  is a constant value called bias, and  $a_{i-1}^j$  output of node  $j_{th}$  of layer  $(i-1)_{th}$ .

The net input of node  $j_{th}$  of layer  $i_{th}$  is then passed through an activation function to produce the output of node  $j_{th}$  of layer  $i_{th}$ , which is then transmitted to other neurons.

$$a_i^j = g(z_i^j), \quad (8)$$

where  $g$  is the active function.

The approximate models are greatly dependent on the structure of the DFNN and the actual data. To get the exact approximate models, the structure of the DFNN is optimized in this study. The optimization process for the structure of the DFNN is performed as follows:

Step 7: define the objective function of the DFNN structure.

The accuracy of the approximate model is usually assessed through coefficient of determination ( $R^2$ ) and mean squared error (MSE). In practice, if the value of MSE is smaller, the value of  $R^2$  will be larger and the predicted model is more accurate. Therefore, in this study, MSE is selected as the objective function to optimize the structure of the DFNN.

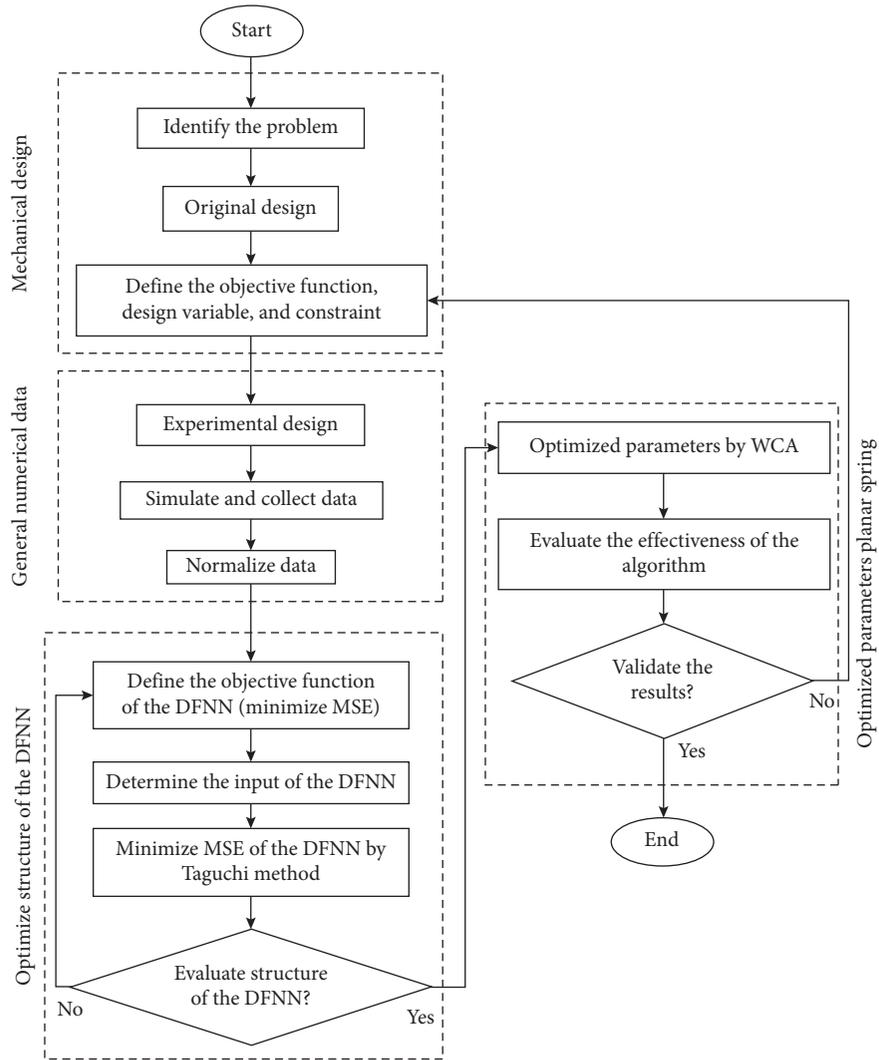


FIGURE 4: Flowchart of the hybrid approach.

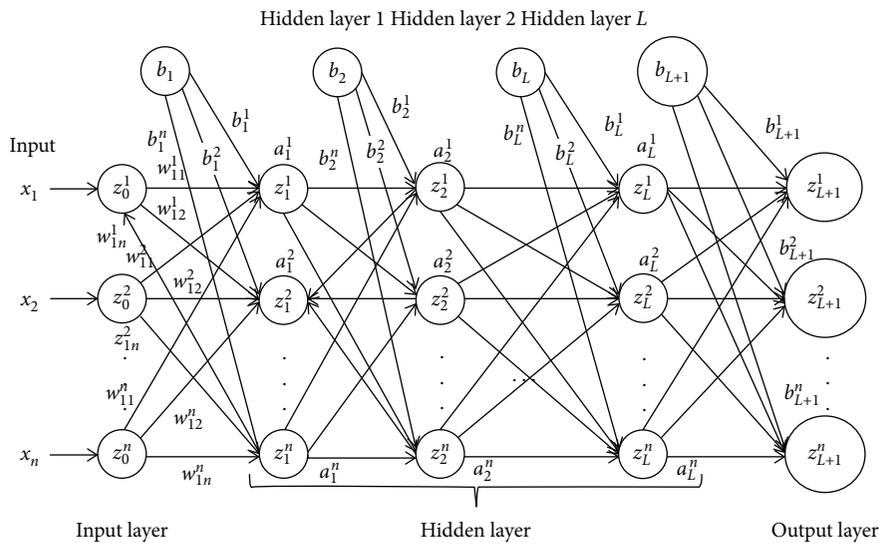


FIGURE 5: Structure of deep feedforward neural network.

The MSE and  $R^2$  are calculated by using the following formula:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (9)$$

$$R^2 = 1 - \frac{\text{residual sum of squares}}{\text{total sum of squares}},$$

where  $y_i$  is the  $i$ th observed value  $\hat{y}_i$  is the  $i$ th predicted value and  $n$  is the experiment number.

Step 8: select the input parameters of the DFNN structure

A basic structure of the DFNN includes the number of hidden layers, the number of nodes in each hidden layer, training function, activation function, bias coefficient, ratio of division of data for training, validation, and testing. However, in this study, the number of hidden layers, the number of nodes in each hidden layer, the transfer function and the ratio of division of data are chosen as controllable variables.

Normally, researchers usually use training functions: trainlm, trainbr, and trainscg. However, trainbr only uses 2 datasets: training and validating while the other transfer functions use 3 datasets: training, validating, and testing. Therefore, to evaluate equity, this study proposes to choose 3 training functions: trainlb, traincgs, and trainscg.

The number of nodes in hidden layers can be selected differently based on the position of the hidden layer and the number of nodes of the input layer. According to Chen [23], the number of nodes in the hidden layer can be calculated as follows:

$$\begin{aligned} \text{number node of hidden layer } 1 &= 2x + 1, \\ \text{number node of hidden layer } i + 1 &= n(2x + 1), \end{aligned} \quad (10)$$

where  $x$  is the number of input variables, and  $n$  is the number of output variables. In this study,  $x = 4$  and  $n = 1$ , so the number of nodes in each hidden layer is selected around 9.

According to Seo et al. [20], the accuracy of the approximate models can be improved as the number of hidden layers increases. However, the complexity of the algorithm also increases. The authors also indicate that when the number of hidden layers is equal or larger than 4, the accuracy increased slightly. For the data that have a small signal to noise, the approximate model has a good accuracy as well when using 2 or 3 hidden layers. Therefore, in this study, the number of hidden layers is selected as 2, 3, and 4. The ratio of division of data for training, validating, and testing is chosen with the ratios of 60:20:20, 70:15:15, and 80:10:10.

Step 9: collect data and optimize the structure of the DFNN

Taguchi is often used to improve a product's quality [24]. Nowadays, the Taguchi method is widely used to solve technical problems, especially for the single-objective problem with discrete variables [25, 26]. Hence,

it is chosen to optimize the structure of the DFNN. The Taguchi method uses two major tools, including signal-to-noise ratio ( $S/N$ ) and orthogonal array. There are three quality characteristics: "nominal is best," "smaller is better," and "larger is better." Their mathematical formula can be presented as follows:

Nominal is the best:

$$S/N = 10 \log_{10} \left[ \left( \frac{\bar{y}}{s} \right)^2 \right]. \quad (11)$$

The smaller is better:

$$S/N = -10 \log_{10} \left( \frac{\sum_{i=1}^N y_i^2}{N} \right). \quad (12)$$

The bigger is better:

$$S/N = -10 \log_{10} \left( \frac{\sum_{i=1}^N (1/y_i^2)}{N} \right), \quad (13)$$

where  $y_i$  is responses, and  $N$  is repetition of  $i^{\text{th}}$  experiment.

The Taguchi method used orthogonal arrays to reduce the number of experiments required but ensured that design variables are evaluated independently. The purpose of this step is to find the structure of the DFNN that best matches the existing dataset by minimizing MSE by using equation (12).

Step 10: evaluate the value of MSE

After optimizing the structure of the DFNN, the value of MSE has to be evaluated to find the most suitable structure for the dataset. If the value of MSE is not satisfied, it means that the most suitable structure has not been found, then go back to step 8. If the value of MSE is satisfied, then go to the next step.

**4.4. Stage 4: Optimization Using the WCA.** The WCA is an optimal algorithm inspired by the water cycle. It was developed by Eskandar [21] and widely used to solve technical problems [27–29]. However, this algorithm has not been applied to solve the problem of optimizing strain energy of the PS. The flowchart of the WCA is presented, as shown in Figure 6.

Step 11: choose the initial parameters of the WCA

Initial rain drops ( $N_{\text{pop}}$ —initial population), number of rivers and sea ( $N_{\text{sr}}$ ), maximum distance between streams and rivers, between rivers and sea  $d_{\text{max}}$ , and the maximum iteration must be preselected.  $N_{\text{sr}}$  and  $N_{\text{raindrop}}$  are calculated as follows:

$$\begin{aligned} N_{\text{sr}} &= \text{number of rivers} + \text{sea}, \\ N_{\text{raindrop}} &= N_{\text{pop}} - N_{\text{sr}}, \end{aligned} \quad (14)$$

where  $N_{\text{sr}}$  is the number of sea and rivers,  $N_{\text{pop}}$  is the number of the initial population, and  $N_{\text{raindrop}}$  is the number of streams.

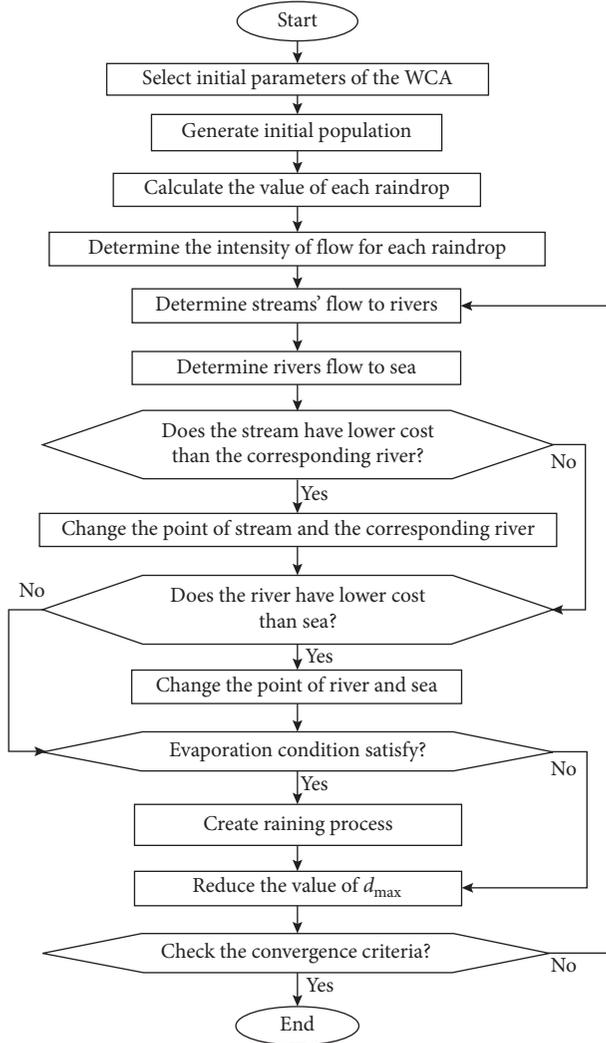


FIGURE 6: Flowchart of the WCA.

Step 12: generate random initial population

Initial population is generated by random variables as follows:

$$\text{population of raindrop} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N_{\text{var}}}^1 \\ x_1^2 & x_2^2 & \dots & x_{N_{\text{var}}}^2 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ x_1^{N_{\text{pop}}} & x_2^{N_{\text{pop}}} & \dots & x_{N_{\text{var}}}^{N_{\text{pop}}} \end{bmatrix}, \quad (15)$$

where  $x_i^j$  is the value of the variable  $i_{\text{th}}$  of the raindrop  $j_{\text{th}}$ ,  $i$  is the number of variables,  $j$  is the number of the initial population, and  $N_{\text{var}}$  is the number of variables.

Step 13: calculate the value of the initial raindrop

In order to select raindrops representing the sea and rivers, the value of the initial raindrop needs to be

calculated and they are calculated according to the following formula:

$$P_i = \text{cost}_i = f(\mathbf{X}_i), \quad i = 1, 2, 3, \dots, N_{\text{pop}}, \quad (16)$$

where  $P_i$  is the initial raindrop value.  $i$  is raindrop.

Step 14: determine the intensity of flow for rivers and sea

The raindrops flow to the rivers or sea depending on the intensity of the flow. The intensity of flow for rivers and sea are calculated by the following formula:

$$NS_n = \text{round} \left\{ \left\lfloor \frac{P_n}{\sum_{i=1}^{N_{\text{sr}}} P_i} \right\rfloor \cdot N_{\text{raindrop}} \right\}, \quad n = 1, 2, \dots, N_{\text{sr}}, \quad (17)$$

where  $NS_n$  is the number of streams flowing directly into the rivers or sea,  $P_n$  is the value of sea or rivers  $n_{\text{th}}$

Step 15: the streams flow to the rivers

During the streams' flow to rivers, the positions of streams are continuously updated. The new position for the stream is determined as follows:

$$X_{\text{stream}}^{i+1} = X_{\text{stream}}^i + \text{rand} \times Cx(X_{\text{river}}^i - X_{\text{stream}}^i), \quad (18)$$

where rand is a random number that is chosen between 0 and 1,  $X_{\text{stream}}^i$  is the location of the stream  $i_{\text{th}}$ ,  $X_{\text{river}}^i$  is the location of the river  $i_{\text{th}}$ , and  $C$  is a value between 1 and 2.

Step 16: the rivers flow to the sea

Similar to a stream, when rivers flow to the sea, the position of the river is always updated, and the new position of the river is determined as follows:

$$X_{\text{river}}^{i+1} = X_{\text{river}}^i + \text{rand} \times Cx(X_{\text{sea}}^i - X_{\text{river}}^i), \quad (19)$$

where rand is a random number that is chosen between 0 and 1,  $X_{\text{river}}^i$  is the location of the stream  $i_{\text{th}}$ ,  $X_{\text{sea}}^i$  is the location of the river  $i_{\text{th}}$ , and  $C$  is a value between 1 and 2.

Step 17: exchange positions of the river with a stream

To offer the best solution, the new cost of a stream is updated. If the solution given by a stream is better than its connecting river, the positions of the river and stream are exchanged

Step 18: exchange positions of the sea with a river

Similar to streams, if the solution of the river is better than the sea, the positions of the sea and the river will be swapped

Step 19: check the evaporation condition

Evaporation is one factor that prevents the algorithm from being optimized locally. As seen in nature, water from ponds, lakes, rivers, streams, seas, and the ground evaporates into clouds. Clouds fly up high then condense in the cold air to form water particles. Water particles fall down and form rain. Raindrops will form

new streams that flow into rivers and the sea. In the WCA, the condition evaporation is determined as follows:

$$\left| X_{\text{sea}}^i - X_{\text{river}}^i \right| < d_{\text{max}}, \quad i = 1, 2, 3, \dots, N_{sr} - 1, \quad (20)$$

where  $d_{\text{max}}$  is small number (close to zero),

When the distance from the river to sea is less than  $d_{\text{max}}$ , it means the river has flowed into the sea. Then, the evaporation process is applied. After the evaporation is completed, rain will begin to create a new cycle.

Step 20: reduce the value of  $d_{\text{max}}$

In the WCA, the value of  $d_{\text{max}}$  controls the search intensity near the sea. If the value of  $d_{\text{max}}$  is large, then the search density will reduce while a small value for  $d_{\text{max}}$  will increase the search density near the sea. Therefore, the value for  $d_{\text{max}}$  has to adaptively decrease as:

$$d_{\text{max}}^{i+1} = d_{\text{max}}^i - \frac{d_{\text{max}}^i}{\text{max iteration}}. \quad (21)$$

Step 21: check the convergence criteria

Like other optimization algorithms, the stop condition of the WCA algorithm can be based on the maximum number of iterations, the CPU time, or the error between the last two results less than a certain tolerance value. If the stopping condition is satisfied, the optimization will be finished, otherwise return to step 5.

## 5. Results and Discussion

**5.1. Collecting and Processing Data.** To perform simulation and collect data, firstly, the experimental design was constructed by a full experimental design. The four design variables of PS include  $r$ ,  $L$ ,  $t$ , and  $w$ , which are divided into three levels, as shown in Table 1. A matrix of experimental designs is formed, as seen in Table 2. Then, 3D models of the PS are drawn by Inventor software as shown in Figure 2. Next, the simulation for data collection is performed through ANSYS 18.2.

In this study, a nonlinear FEM is applied for the simulation process. The simulation process is set up as follows: the boundary condition and loads are given as shown in Figure 7. The PS is fixed with two holes, and a force of 10.935 N is applied to the top surface of the PS. Al T63-7075 is selected for the PS. The mechanical properties of the material are given in Table 2. The mesh is divided by using the sizing method. The element size greatly affects the behavior of the PS. So, before the simulation, an exploratory meshing was performed to ensure the good results at the lowest computational cost. The results show that the smaller the mesh size, the larger the processing time and memory used for the simulation. However, if the mesh size is 1 mm, the deformation, stress, and strain energy change very little. Hence, a mesh size of 1 mm is chosen for this study. The result meshes have 101190 elements and 632792 nodes. The quality of the mesh greatly affects the simulation results.

Especially, the PS has large deformation. Therefore, in this study, two mesh quality assessment standards are used at the same time. Skewness standard and Orthogonal Quality standard are utilized as two criteria to evaluate the quality of meshing. The quality of the mesh is assessed according to Skewness criteria. The results indicate that the average value of the meshing metric is about 0.21141, and the standard deviation of the meshing metric is approximately 0.194, as shown in Figure 8. According to the Skewness criteria, this value proved that the accuracy of meshing models is excellent. The quality of the mesh is assessed according to Orthogonal Quality criteria. The results indicate that the average value of the meshing metric is about 0.93498, and the standard deviation of the meshing metric is approximately 0.0751, as given in Figure 8. According to the Orthogonal Quality criteria, this value proved that the accuracy of meshing models is very good.

In this study, strain energy is considered an objective function; therefore, strain energy is a factor that needs data collection. In addition, deformation and stress are considered constraints so they also need to be collected as data. The results of data collection are shown in Table 3.

The properties of the PS have different units. Therefore, in this study, datasets are normalized to eliminate the units of measurement for data. This normalization makes the evaluation process fair. Data are normalized using (6). Normalization results are presented in Table 4.

**5.2. Optimization of the DFNN Structure.** As presented in Section 4, the DFNN is used to approximate the relationship between the design variable and the output characteristics. The accuracy of the approximated model depends on the structure of the DFNN. In this study, four factors of the DFNN structure are selected: training function, number of hidden layers, number of nodes in each hidden layer, and the division ratio of the dataset to create the most accurate approximated model possible. The factors are divided with 3 levels as illustrated in Table 5 and the orthogonal array  $L_9$  is used to establish the experimental plan as shown in Table 6.

As shown in Section 4, normally, the approximation model's accuracy is measured through two indices MSE and  $R^2$ . However, when the MSE is small, the  $R^2$  is large and the approximation model has high accuracy. Therefore, in this study, MSE is selected as the objective function for the structural optimization of the DFNN. The approximate results depend on a weighted input that selects random as equation (7). So, in order to optimize the structure of the DFNN, each experiment is performed 10 times to collect data for MSE. MSE data collection results for deformation, stress, and strain energy are presented in Tables 7–9.

Based on MSE results obtained for deformation, stress, and strain energy, the S/N calculation and analysis for each experiment are performed, then ranking process for each input parameter of the DFNN structure are performed.

The S/N analysis results for deformations (Figure 9) show that, the training function "Trainlm" has the smallest error and the function "Trainscg" gives the largest error. The structure of the DFNN consists of 2 hidden layers for the

TABLE 1: Design variables with three levels.

Variables	Level 1	Level 2	Level 3
$t$ (mm)	1.0	1.2	1.4
$L$ (mm)	45	50	55
$w$ (mm)	9	11.5	14
$r$ (mm)	0.1	0.2	0.3

TABLE 2: Mechanical properties of the material.

Density	Yield strength (MPa)	Poisson's ratio	Young modulus (MPa)
2810 kg/m <sup>3</sup>	503	0.33	71700

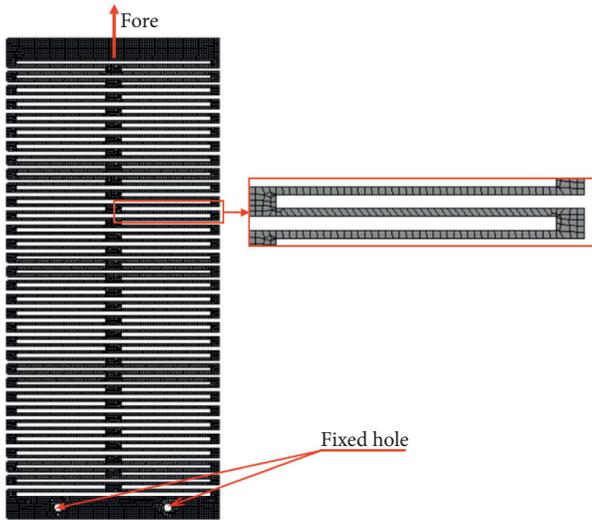


FIGURE 7: Meshing model and boundary conditions.

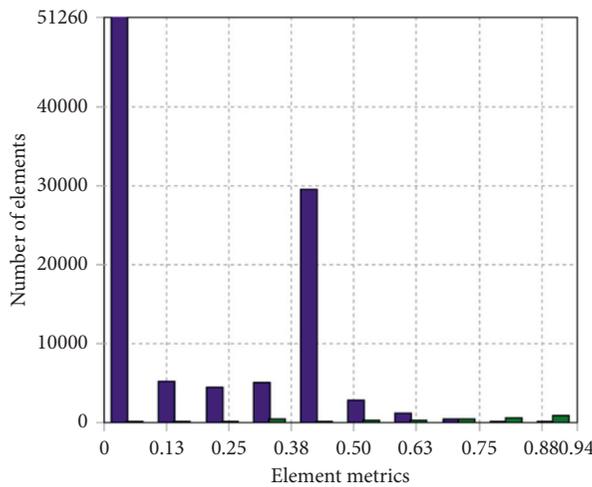


FIGURE 8: Evaluating the quality of the meshing.

smallest MSE value and 4 hidden layers with the largest error. The results found that when the number of nodes is 7 nodes, the error of the DFNN model is the smallest. Meanwhile, the when the number of nodes is 9 nodes, the error of the DFNN model is the largest. Besides, when the data are divided into the ratio of 60:20:20, the error of the DFNN model is the smallest; while the data ratio is 80:10:10,

the error of the DFNN model is the largest. The analysis results also show that the training function in the DFNN structure has the greatest impact on the accuracy of the approximate model and the number of nodes in each hidden layer has the least effect on the accuracy of the equivalent model as shown in Table 10.

For the stress dataset, Figure 10 shows that the training function “Trainlm” gives the largest S/N value and the training function “Trainscg” the minimum S/N value. Thus, the accuracy of the approximate model is best achieved using the training function “Trainlm.” At the same time, the number of hidden layers is 2, the number of nodes in a hidden layer is 7, and the data division ratio 70:15:15 gives the largest S/N. This means that the approximate model has the highest accuracy when the structure has 2 hidden layers, the number of nodes in each hidden layer is 7, and the data division ratio is 70:15:15. Table 11 also shows that, with data for stress, the number of layers hidden in the DFNN structure has the most influence on the accuracy of the approximate model.

Figure 11 depicts the calculation and analysis S/N results for the strain energy dataset. The results show the DFNN structure with the training function of “Trainlm,” the number of hidden layers of 2, the number of nodes in each hidden layer of 7, and the data division ratio of 60:20:20 give the best accurate approximate model. Table 12 shows that the number of hidden layers is ranked the highest. This means that among the four factors of the DFNN structure, the number of hidden layers has the highest influence on the accuracy of the approximate model.

Table 13 shows that for the deformation dataset, the optimal structure of the DFNN is: The training function is “trainlm,” the number of hidden layers is 2, the number of nodes in hidden layers is 7, and the data division ratio is 60:20:20. For the stress dataset, the optimal structure of the DFNN is: The training function is “trainlm,” the number of hidden layers is 2, the number of nodes in hidden layers is 11, and the data division ratio is 70:15:15. For the strain energy dataset, the optimal structure of the DFNN is: The training function is “trainlm,” the number of hidden layers is 2, the number of nodes in hidden layers is 9, and the data division ratio is 60:20:20.

To evaluate the effectiveness of the structure of the DFNN, the appropriateness of the approximate model built by the DFNN and the linear model is compared with each other.

TABLE 3: Experimental design and simulation data.

No.	$t$ (mm)	$L$ (mm)	$w$ (mm)	$r$ (mm)	Deformation (mm)	Stress (MPa)	Strain energy (mJ)
1	1	45	9	0.1	37.648	143.88	0.01136
2	1	45	9	0.2	37.359	126.56	0.011931
3	1	45	9	0.3	37.181	91.524	0.01208
4	1	45	11.5	0.1	29.158	110.5	0.0075272
5	1	45	11.5	0.2	28.933	97.469	0.0073316
6	1	45	11.5	0.3	28.794	70.283	0.0074207
7	1	45	14	0.1	23.743	89.134	0.004959
8	1	45	14	0.2	23.559	78.896	0.0052279
9	1	45	14	0.3	23.445	56.8	0.0052896
10	1	50	9	0.1	52.955	135.96	0.015283
11	1	50	9	0.2	52.482	183.47	0.015191
12	1	50	9	0.3	52.271	103.04	0.01534
13	1	50	11.5	0.1	41.035	105.25	0.010162
14	1	50	11.5	0.2	40.666	141.99	0.01011
15	1	50	11.5	0.3	40.502	79.32	0.010206
16	1	50	14	0.1	33.423	85.454	0.0067236
17	1	50	14	0.2	33.121	115.14	0.0066931
18	1	50	14	0.3	32.987	64.151	0.0067545
19	1	55	9	0.1	72.04	175.37	0.018836
20	1	55	9	0.2	71.362	203.9	0.018837
21	1	55	9	0.3	71.012	114.56	0.019014
22	1	55	11.5	0.1	55.856	135.11	0.012552
23	1	55	11.5	0.2	55.326	158.02	0.012564
24	1	55	11.5	0.3	55.053	88.312	0.012679
25	1	55	14	0.1	45.508	109.33	0.008325
26	1	55	14	0.2	45.075	128.3	0.0083395
27	1	55	14	0.3	44.852	71.513	0.008414
28	1.2	45	9	0.1	22.22	96.941	0.0065149
29	1.2	45	9	0.2	22.13	66.215	0.0063295
30	1.2	45	9	0.3	21.933	65.468	0.0062087
31	1.2	45	11.5	0.1	17.205	74.773	0.0040134
32	1.2	45	11.5	0.2	17.135	51.129	0.003901
33	1.2	45	11.5	0.3	16.981	50.323	0.003827
34	1.2	45	14	0.1	14.007	60.582	0.0028689
35	1.2	45	14	0.2	13.95	41.457	0.0027898
36	1.2	45	14	0.3	13.824	40.669	0.0027374
37	1.2	50	9	0.1	31.178	108.37	0.0082735
38	1.2	50	9	0.2	31.011	90.501	0.0080522
39	1.2	50	9	0.3	30.784	73.985	0.0079012
40	1.2	50	11.5	0.1	24.156	83.785	0.0055193
41	1.2	50	11.5	0.2	24.025	69.716	0.0053739
42	1.2	50	11.5	0.3	23.848	57.009	0.005274
43	1.2	50	14	0.1	19.671	67.908	0.0036635
44	1.2	50	14	0.2	19.565	56.337	0.003569
45	1.2	50	14	0.3	19.419	46.096	0.0035029
46	1.2	55	9	0.1	42.206	125.93	0.010211
47	1.2	55	9	0.2	42.035	82.775	0.010003
48	1.2	55	9	0.3	41.738	81.78	0.0098
49	1.2	55	11.5	0.1	32.717	97.869	0.0068261
50	1.2	55	11.5	0.2	32.584	64.17	0.0066893
51	1.2	55	11.5	0.3	32.353	63.11	0.0065549
52	1.2	55	14	0.1	26.652	79.533	0.0045419
53	1.2	55	14	0.2	26.543	52.124	0.004453
54	1.2	55	14	0.3	26.354	51.099	0.0043642
55	1.4	45	9	0.1	14.31	70.515	0.0028812
56	1.4	45	9	0.2	14.21	71.532	0.0023315
57	1.4	45	9	0.3	14.106	66.273	0.0021754
58	1.4	45	11.5	0.1	11.078	54.279	0.0017708
59	1.4	45	11.5	0.2	11	55.096	0.0014267
60	1.4	45	11.5	0.3	10.919	51.038	0.0013319

TABLE 3: Continued.

No.	$t$ (mm)	$L$ (mm)	$w$ (mm)	$r$ (mm)	Deformation (mm)	Stress (MPa)	Strain energy (mJ)
61	1.4	45	14	0.1	9.0172	43.868	0.0012658
62	1.4	45	14	0.2	8.9536	44.557	0.0010189
63	1.4	45	14	0.3	8.8876	41.251	0.00095013
64	1.4	50	9	0.1	20.005	79.204	0.0036771
65	1.4	50	9	0.2	19.885	73.889	0.002852
66	1.4	50	9	0.3	19.747	74.814	0.0027744
67	1.4	50	11.5	0.1	15.496	61.129	0.0024474
68	1.4	50	11.5	0.2	15.403	57.082	0.0018876
69	1.4	50	11.5	0.3	15.295	57.768	0.0018403
70	1.4	50	14	0.1	12.618	49.426	0.0016241
71	1.4	50	14	0.2	12.541	46.18	0.0012489
72	1.4	50	14	0.3	12.453	46.705	0.0012188
73	1.4	55	9	0.1	27.052	88.311	0.0038068
74	1.4	55	9	0.2	26.893	75.666	0.0035977
75	1.4	55	9	0.3	26.726	81.191	0.0034145
76	1.4	55	11.5	0.1	20.967	68.245	0.0025272
77	1.4	55	11.5	0.2	20.843	58.445	0.002389
78	1.4	55	11.5	0.3	20.713	62.752	0.0022659
79	1.4	55	14	0.1	17.078	55.247	0.0016786
80	1.4	55	14	0.2	16.976	47.321	0.0015868
81	1.4	55	14	0.3	16.87	50.783	0.0015041

Figure 12 compares the suitability of the proposed model with a linear model. Figure 12(a) shows that the predicted points of the proposed model are more suitable with the experimental point than the linear model for the deformation dataset. The predicted points of the proposed model are distributed around the good fitness line. Similarly, the proposed model has better fitness between the predicted points and the experimental points than the linear model for stress and strain energy datasets, as shown in Figures 12(b) and 12(c).

In addition, to evaluate the accuracy of the proposed model, the value  $R^2$  and MSE of the proposed model are also compared with the value of the linear model. Table 14 compares the  $R^2$  value of the proposed model with the linear model for the 4 datasets: model, training, testing, and validation. The results indicate that the proposed model has much better  $R^2$  than the linear model. Table 15 also shows that the MSE of the proposed model is smaller than the error of the linear model for all three sets of deformation, stress, and strain energy data.

As illustrated in Figure 13(a), the MSE values of the training dataset of the deformation continuously decreased at the 9th epoch, but the MSE values of the testing and validating dataset still remained stable. As depicted in Figure 13(b), the MSE values of the training datasets of the stress still decreased, but the MSE values of the testing and validating datasets remained stable at 9th epoch. As depicted in Figure 13(c), the MSE values of the training datasets of the strain energy still decreased, but the MSE values of the testing and validating datasets remained stable at 8th epoch.

Figure 14 demonstrates that the errors of the three training, evaluation, and test datasets of all three sets of deformation, stress, and strain energy datasets are almost zero.

Figure 15 also demonstrates that the appropriateness of the approximate models is constructed by the DFNN through the value of  $R^2$  for the datasets.

**5.3. Optimal Results.** After optimizing the structure of the DFNN, an alternative model using the DNFF was established. Then, the WCA algorithm was carried out using Matlab R2018a to optimize the parameters of the planar spring. The parameters of the WCA were initialized as initial rain drops of 50, number of rivers and sea of 4, maximum distance between streams and rivers, between river and sea of  $10^{-5}$ , and the maximum iteration of 2000.

The optimal results were generated as Table 16. The evaporation-rate-based water cycle algorithm (ER-WCA) was developed based on the WCA to speed up convergence [29]. However, with the proposed PS, the WCA has higher convergence rate and stability than the ER-WCA. The results in Table 16 show that the optimal mean strain energy is 0.01123 when optimized by the WCA. It is higher than the ER-WCA optimal value. The standard deviation of energy when optimized by the WCA is also smaller than when optimized by the ER-WCA. The average search time also shows that the WCA has a shorter search time than the ER-WCA. The standard deviation of the search time of the WCA is also smaller than the ER-WCA. The optimal results show that the optimal parameters of the planar spring  $t$ ,  $L$ ,  $w$ , and  $r$  is 1.029 mm, 45 mm, 9 mm, 0.3 mm, respectively.

## 6. Verifications

Optimized results are used to build the 3D model. The model was then used for finite element analysis. Finite element analysis results are presented in Table 17 and Figure 16. FEM results show with optimal parameters. Characteristics of planar spring with strain energy, stress, deformation, and life are 0.01102 mJ, 34.236 mm, 81.475 MPa, and 299 million cycles, respectively. Comparison between the prediction results with the FEM results shows that the strain energy error is 1.87%, the strain error is 1.69%, and the stress error is 3.06%. With

TABLE 4: Data normalization results.

No.	$t$ (mm)	$L$ (mm)	$w$ (mm)	$r$ (mm)	$Z_{Y1}$	$Z_{Y2}$	$Z_{Y3}$
1	1	45	9	0.1	0.455413	0.6323	0.576281
2	1	45	9	0.2	0.450836	0.526193	0.607891
3	1	45	9	0.3	0.448018	0.311552	0.61614
4	1	45	11.5	0.1	0.320976	0.427805	0.364101
5	1	45	11.5	0.2	0.317413	0.347973	0.353273
6	1	45	11.5	0.3	0.315212	0.181424	0.358205
7	1	45	14	0.1	0.235231	0.296911	0.221928
8	1	45	14	0.2	0.232317	0.23419	0.236814
9	1	45	14	0.3	0.230512	0.098823	0.240229
10	1	50	9	0.1	0.697795	0.58378	0.793455
11	1	50	9	0.2	0.690305	0.87484	0.788362
12	1	50	9	0.3	0.686964	0.382103	0.796611
13	1	50	11.5	0.1	0.509045	0.395642	0.509961
14	1	50	11.5	0.2	0.503202	0.620722	0.507082
15	1	50	11.5	0.3	0.500605	0.236787	0.512397
16	1	50	14	0.1	0.388511	0.274366	0.319614
17	1	50	14	0.2	0.383729	0.456231	0.317926
18	1	50	14	0.3	0.381607	0.143857	0.321325
19	1	55	9	0.1	1	0.825217	0.990146
20	1	55	9	0.2	0.989264	1	0.990201
21	1	55	9	0.3	0.983722	0.452677	1
22	1	55	11.5	0.1	0.743731	0.578573	0.642269
23	1	55	11.5	0.2	0.735339	0.718926	0.642934
24	1	55	11.5	0.3	0.731016	0.291875	0.6493
25	1	55	14	0.1	0.579873	0.420637	0.408266
26	1	55	14	0.2	0.573017	0.536853	0.409069
27	1	55	14	0.3	0.569486	0.188959	0.413193
28	1.2	45	9	0.1	0.211115	0.344738	0.308061
29	1.2	45	9	0.2	0.20969	0.156502	0.297797
30	1.2	45	9	0.3	0.20657	0.151926	0.29111
31	1.2	45	11.5	0.1	0.131704	0.208931	0.16958
32	1.2	45	11.5	0.2	0.130595	0.064081	0.163358
33	1.2	45	11.5	0.3	0.128157	0.059143	0.159261
34	1.2	45	14	0.1	0.081064	0.121993	0.106221
35	1.2	45	14	0.2	0.080162	0.004828	0.101843
36	1.2	45	14	0.3	0.078166	0	0.098942
37	1.2	50	9	0.1	0.352962	0.414756	0.405415
38	1.2	50	9	0.2	0.350318	0.305285	0.393164
39	1.2	50	9	0.3	0.346723	0.204103	0.384805
40	1.2	50	11.5	0.1	0.241771	0.264141	0.252945
41	1.2	50	11.5	0.2	0.239696	0.17795	0.244896
42	1.2	50	11.5	0.3	0.236894	0.100104	0.239366
43	1.2	50	14	0.1	0.170752	0.166874	0.15021
44	1.2	50	14	0.2	0.169074	0.095987	0.144978
45	1.2	50	14	0.3	0.166762	0.033247	0.141319
46	1.2	55	9	0.1	0.527587	0.522333	0.512674
47	1.2	55	9	0.2	0.524879	0.257953	0.501159
48	1.2	55	9	0.3	0.520177	0.251858	0.489921
49	1.2	55	11.5	0.1	0.377332	0.350424	0.325289
50	1.2	55	11.5	0.2	0.375226	0.143974	0.317715
51	1.2	55	11.5	0.3	0.371568	0.13748	0.310275
52	1.2	55	14	0.1	0.281294	0.238092	0.198837
53	1.2	55	14	0.2	0.279568	0.070177	0.193916
54	1.2	55	14	0.3	0.276575	0.063897	0.189
55	1.4	45	9	0.1	0.085862	0.182845	0.106902
56	1.4	45	9	0.2	0.084279	0.189076	0.076471
57	1.4	45	9	0.3	0.082632	0.156857	0.06783
58	1.4	45	11.5	0.1	0.034684	0.083379	0.045432
59	1.4	45	11.5	0.2	0.033449	0.088384	0.026382
60	1.4	45	11.5	0.3	0.032167	0.063523	0.021134

TABLE 4: Continued.

No.	$t$ (mm)	$L$ (mm)	$w$ (mm)	$r$ (mm)	$Z_{Y1}$	$Z_{Y2}$	$Z_{Y3}$
61	1.4	45	14	0.1	0.002052	0.019598	0.017475
62	1.4	45	14	0.2	0.001045	0.023819	0.003807
63	1.4	45	14	0.3	0	0.003565	0
64	1.4	50	9	0.1	0.176041	0.236076	0.150963
65	1.4	50	9	0.2	0.174141	0.203515	0.105286
66	1.4	50	9	0.3	0.171955	0.209182	0.10099
67	1.4	50	11.5	0.1	0.104642	0.125344	0.082888
68	1.4	50	11.5	0.2	0.103169	0.100551	0.051898
69	1.4	50	11.5	0.3	0.101459	0.104753	0.049279
70	1.4	50	14	0.1	0.05907	0.053648	0.03731
71	1.4	50	14	0.2	0.057851	0.033762	0.01654
72	1.4	50	14	0.3	0.056457	0.036978	0.014873
73	1.4	55	9	0.1	0.287628	0.291869	0.158143
74	1.4	55	9	0.2	0.28511	0.214402	0.146567
75	1.4	55	9	0.3	0.282466	0.248249	0.136425
76	1.4	55	11.5	0.1	0.191274	0.168938	0.087305
77	1.4	55	11.5	0.2	0.18931	0.108901	0.079655
78	1.4	55	11.5	0.3	0.187252	0.135287	0.07284
79	1.4	55	14	0.1	0.129693	0.089309	0.040327
80	1.4	55	14	0.2	0.128077	0.040752	0.035245
81	1.4	55	14	0.3	0.126399	0.061961	0.030667

TABLE 5: DFNN parameters with 3 levels.

Variable	Level 1	Level 2	Level 3
Training function	Trainlm	Traincgb	Trainscg
Number of hidden layers	2	3	4
Number of nodes	7	9	11
Divide data	60:20:20	70:15:15	80:10:10

TABLE 6: Experimental design using  $L_9$ .

No.	Training function	Number of hidden layers	Number of nodes	Divide data
1	trainlm	2	7	60:20:20
2	trainlm	3	9	70:15:15
3	trainlm	4	11	80:10:10
4	traincgb	2	9	80:10:10
5	traincgb	3	11	60:20:20
6	traincgb	4	7	70:15:15
7	trainscg	2	11	70:15:15
8	trainscg	3	7	80:10:10
9	trainscg	4	9	60:20:20

TABLE 7: MSE results for deformation.

No.	MSE									
	1	2	3	4	5	6	7	8	9	10
1	0.0003	0.0000	0.0002	0.0006	0.0009	0.0002	0.0001	0.0001	0.0004	0.0002
2	0.0031	0.0072	0.0001	0.0004	0.0000	0.0007	0.0019	0.0048	0.0086	0.0006
3	0.0166	0.0029	0.0006	0.0008	0.0039	0.0058	0.0040	0.0026	0.0057	0.0013
4	0.0108	0.0036	0.0030	0.0010	0.0038	0.0150	0.0005	0.0131	0.0036	0.0110
5	0.0043	0.0038	0.0047	0.0058	0.0014	0.0079	0.0059	0.0158	0.0039	0.0061
6	0.0027	0.0110	0.0164	0.0069	0.0012	0.0093	0.0033	0.0064	0.0312	0.0104
7	0.0088	0.0018	0.0015	0.0032	0.0075	0.0045	0.0037	0.0061	0.0025	0.0057
8	0.0130	0.0239	0.0017	0.0141	0.0276	0.0028	0.0113	0.0034	0.0031	0.0106
9	0.0103	0.0121	0.0146	0.0134	0.0222	0.0109	0.0091	0.0068	0.0205	0.0075

TABLE 8: MSE results for stress.

No.	MSE									
	1	2	3	4	5	6	7	8	9	10
1	0.0051	0.0023	0.0181	0.0022	0.0116	0.0063	0.0035	0.0023	0.0053	0.0039
2	0.0165	0.0046	0.0100	0.0075	0.0143	0.0084	0.0140	0.0056	0.0034	0.0032
3	0.0091	0.0103	0.0091	0.0009	0.0171	0.0078	0.0032	0.0104	0.0302	0.0036
4	0.0090	0.0076	0.0069	0.0150	0.0068	0.0087	0.0157	0.0043	0.0088	0.0086
5	0.0066	0.0129	0.0104	0.0082	0.0066	0.0155	0.0079	0.0075	0.0073	0.0090
6	0.0144	0.0106	0.0109	0.0127	0.0166	0.0088	0.0153	0.0132	0.0114	0.0179
7	0.0015	0.0115	0.0115	0.0024	0.0086	0.0118	0.0078	0.0049	0.0096	0.0146
8	0.0542	0.0154	0.0151	0.0037	0.0125	0.0138	0.0375	0.0162	0.0303	0.0137
9	0.0160	0.0124	0.0110	0.0356	0.0048	0.0106	0.0173	0.0270	0.0100	0.0283

TABLE 9: MSE results for strain energy.

No.	MSE									
	1	2	3	4	5	6	7	8	9	10
1	0.0061	0.0001	0.0004	0.0003	0.0004	0.0001	0.0003	0.0037	0.0014	0.0016
2	0.0222	0.0037	0.0086	0.0007	0.0032	0.0024	0.0035	0.0164	0.0082	0.0008
3	0.0013	0.0011	0.0203	0.0011	0.0175	0.0019	0.0296	0.0005	0.0010	0.0034
4	0.0152	0.0060	0.0007	0.0093	0.0016	0.0072	0.0178	0.0049	0.0054	0.0036
5	0.0045	0.0150	0.0039	0.0067	0.0071	0.0058	0.0058	0.0018	0.0078	0.0051
6	0.0048	0.0043	0.0049	0.0058	0.0075	0.0103	0.0037	0.0107	0.0093	0.0043
7	0.0119	0.0118	0.0027	0.0143	0.0046	0.0025	0.0086	0.0039	0.0051	0.0067
8	0.0062	0.0047	0.0015	0.0063	0.0015	0.0020	0.0423	0.0073	0.0061	0.0273
9	0.0043	0.0034	0.0355	0.0061	0.0158	0.0357	0.0091	0.0047	0.0200	0.0104

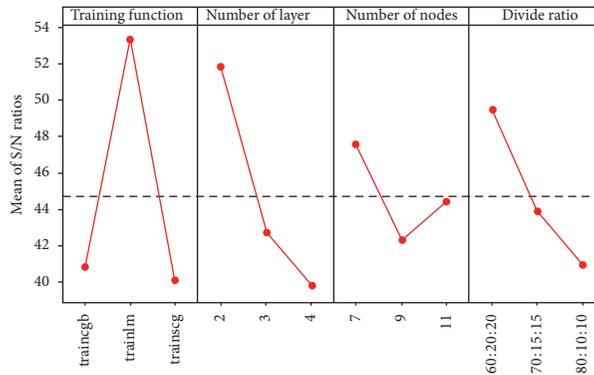


FIGURE 9: Results analysis S/N for deformation.

TABLE 10: Results of rank for deformation.

Level	Training function	Number of hidden layers	Number of nodes	Divide data
1	40.84	51.82	47.60	49.46
2	53.31	42.68	42.27	43.84
3	40.07	39.72	44.34	40.91
Delta	13.24	12.10	5.33	8.54
Rank	1	2	4	3

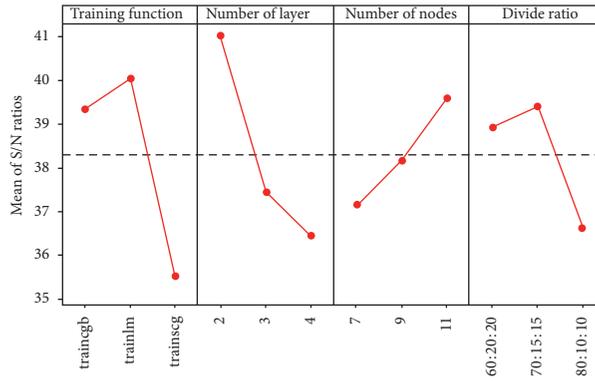


FIGURE 10: Results analysis S/N for stress.

TABLE 11: Results of rank for stress.

Level	Training function	Number of hidden layers	Number of nodes	Divide data
1	39.34	41.01	37.16	38.90
2	40.03	37.44	38.16	39.37
3	35.52	36.44	39.57	36.62
Delta	4.51	4.58	2.41	2.75
Rank	2	1	4	3

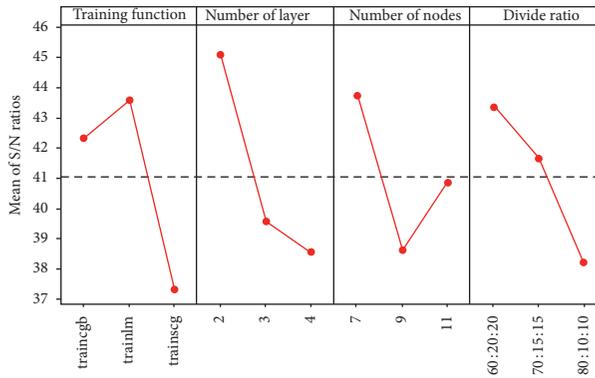


FIGURE 11: Results analysis S/N for strain energy.

TABLE 12: Results of rank for strain energy.

Level	Training function	Number of hidden layers	Number of nodes	Divide data
1	42.32	45.08	43.74	43.34
2	43.56	39.59	38.62	41.66
3	37.31	38.52	40.83	38.19
Delta	6.25	6.56	5.12	5.14
Rank	2	1	4	3

TABLE 13: The optimal structure of DFNN.

Function	Training function	Number of hidden layers	Number of nodes	Divide data
Deformation	trainlm	2	7	60:20:20
Stress	trainlm	2	11	70:15:15
Strain energy	trainlm	2	7	60:20:20

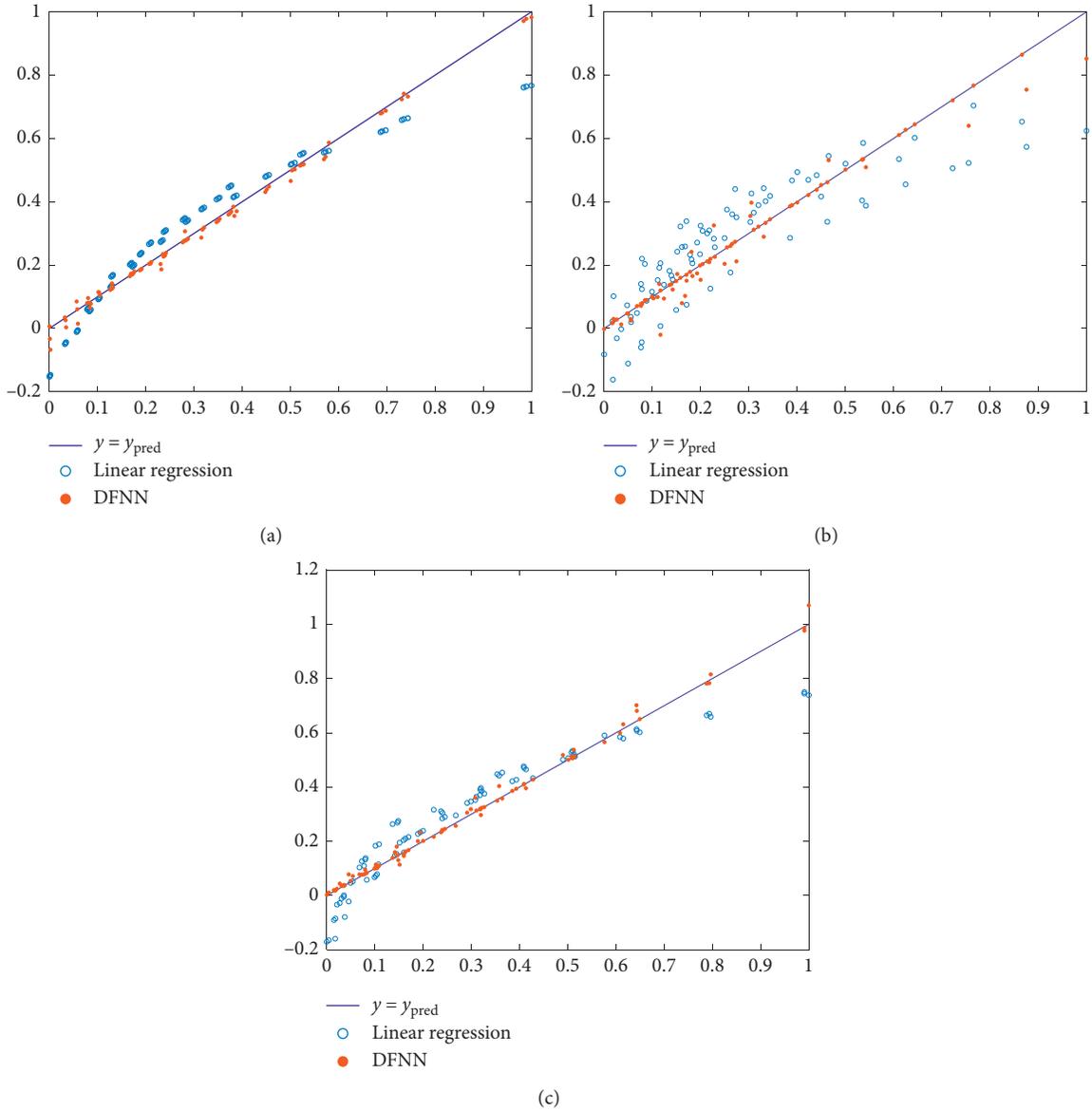


FIGURE 12: The good fitness of model (a) for deformation, (b) for stress, and (c) for strain energy.

TABLE 14: Compare the value  $R^2$  of the proposed model and the linear model.

Response	Model data		Training data		Validating data		Testing data	
	DFNN	Linear	DFNN	Linear	DFNN	Linear	DFNN	Linear
Deformation	0.99	0.91	0.99	0.91	1.0	0.92	0.99	0.96
Stress	0.97	0.76	0.90	0.73	0.95	0.78	0.95	0.82
Strain energy	0.99	0.89	0.99	0.97	1.0	0.89	0.99	0.93

TABLE 15: Compare the value MSE of the proposed model and the linear model.

Response	Model data		Training data		Validating data		Testing data	
	DFNN	Linear	DFNN	Linear	DFNN	Linear	DFNN	Linear
Deformation	$2.82E-4$	0.0050	$8.02E-4$	0.0053	$4.14E-5$	0.0046	$5.00E-4$	0.0014
Stress	0.0015	0.0121	0.0046	0.0099	$5.98E-6$	0.0127	0.0031	0.0112
Strain energy	$3.28E-4$	0.0069	$9.5E-4$	0.0065	$1.3E-5$	0.0061	$6.70E-4$	0.0049

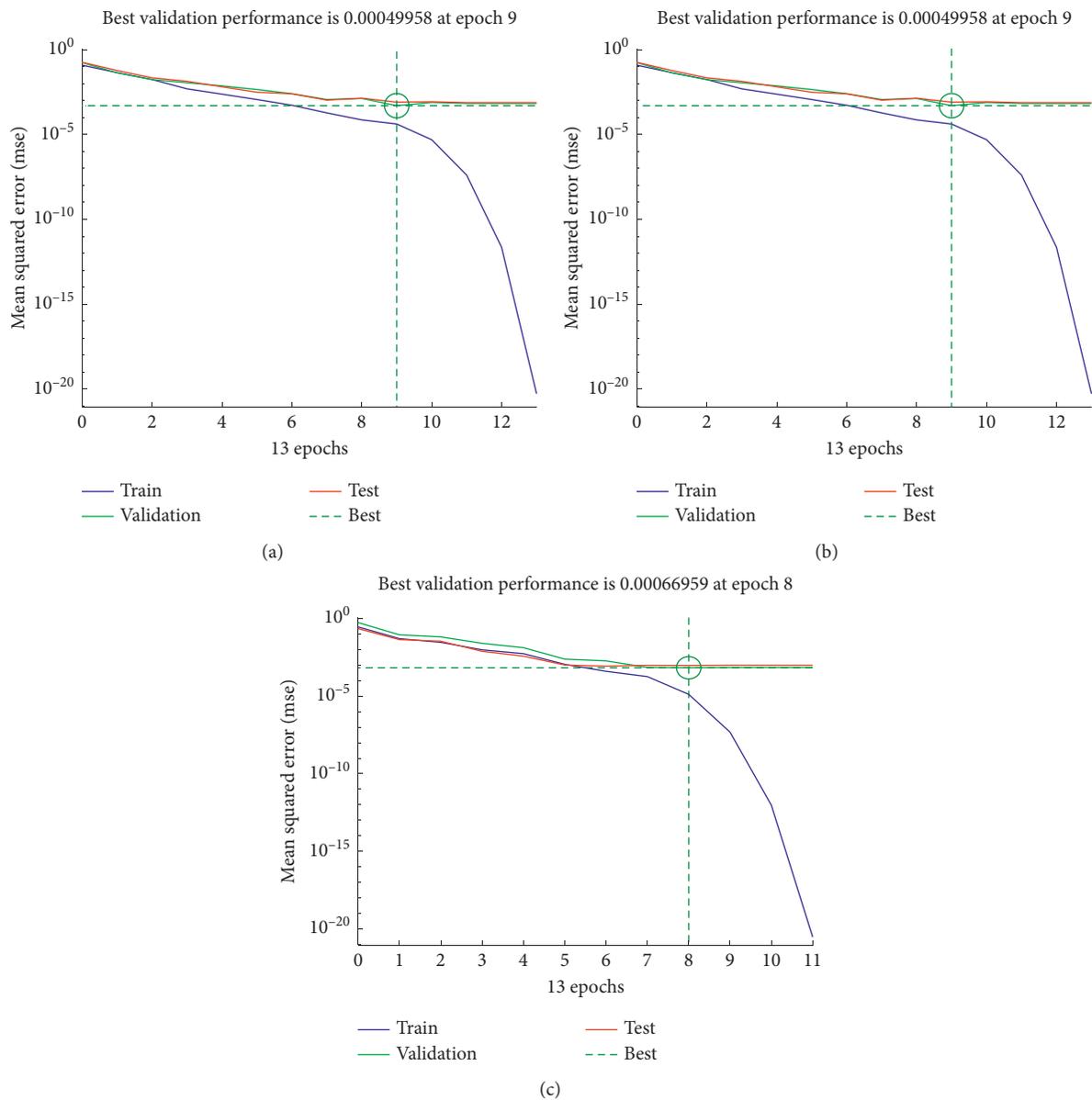


FIGURE 13: The best validation performance: (a) for deformation, (b) for stress, and (c) for strain energy.

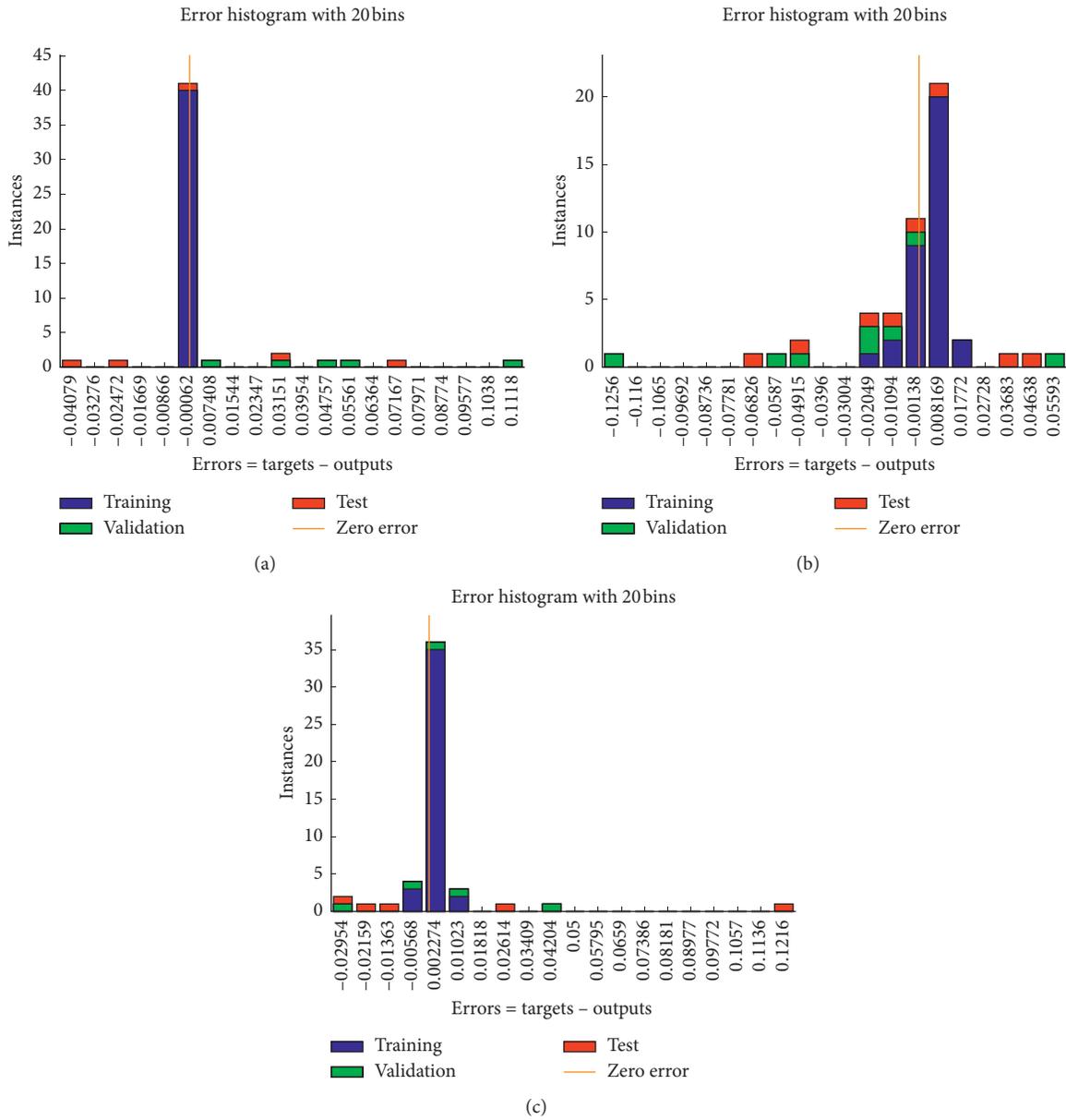
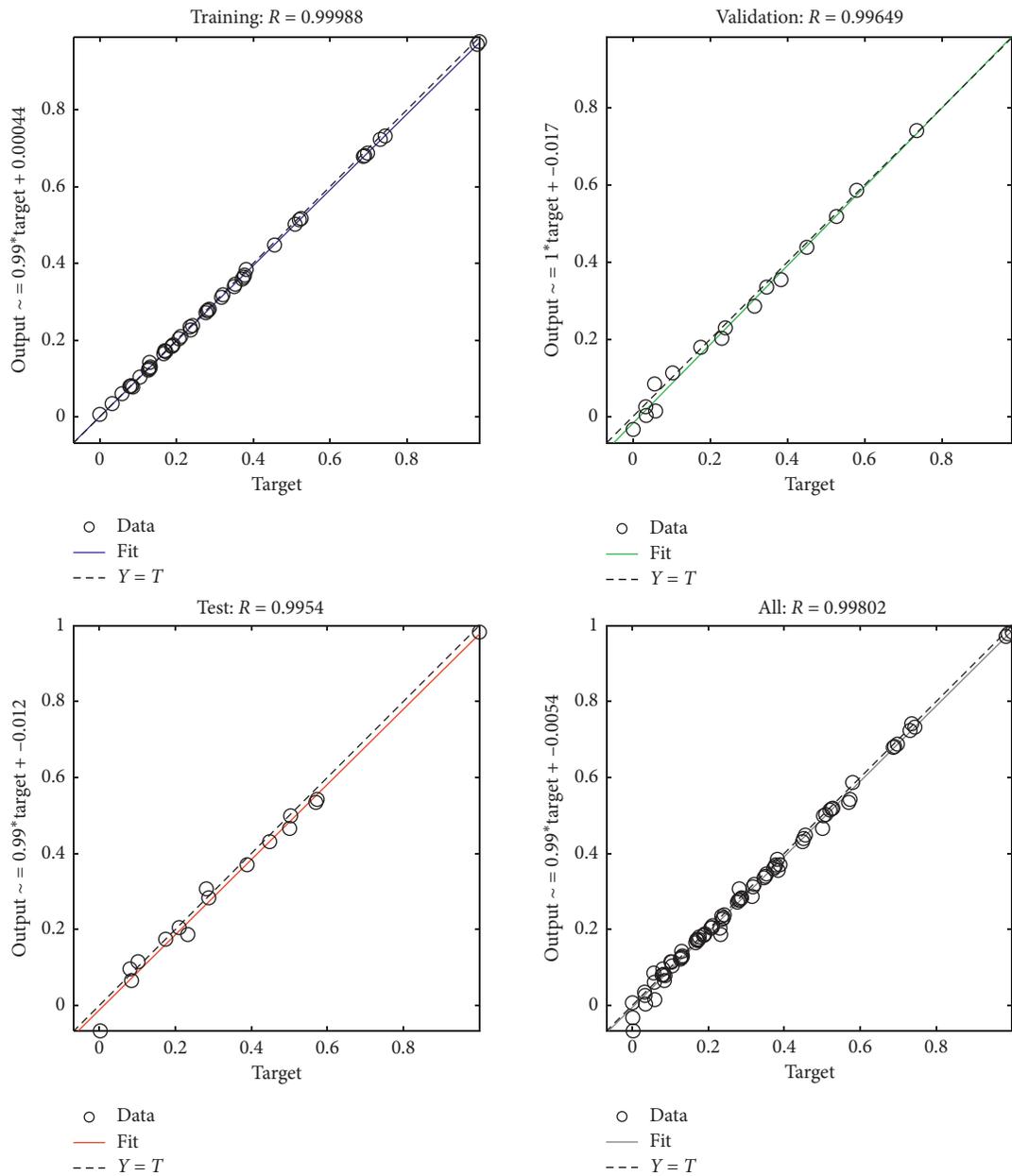
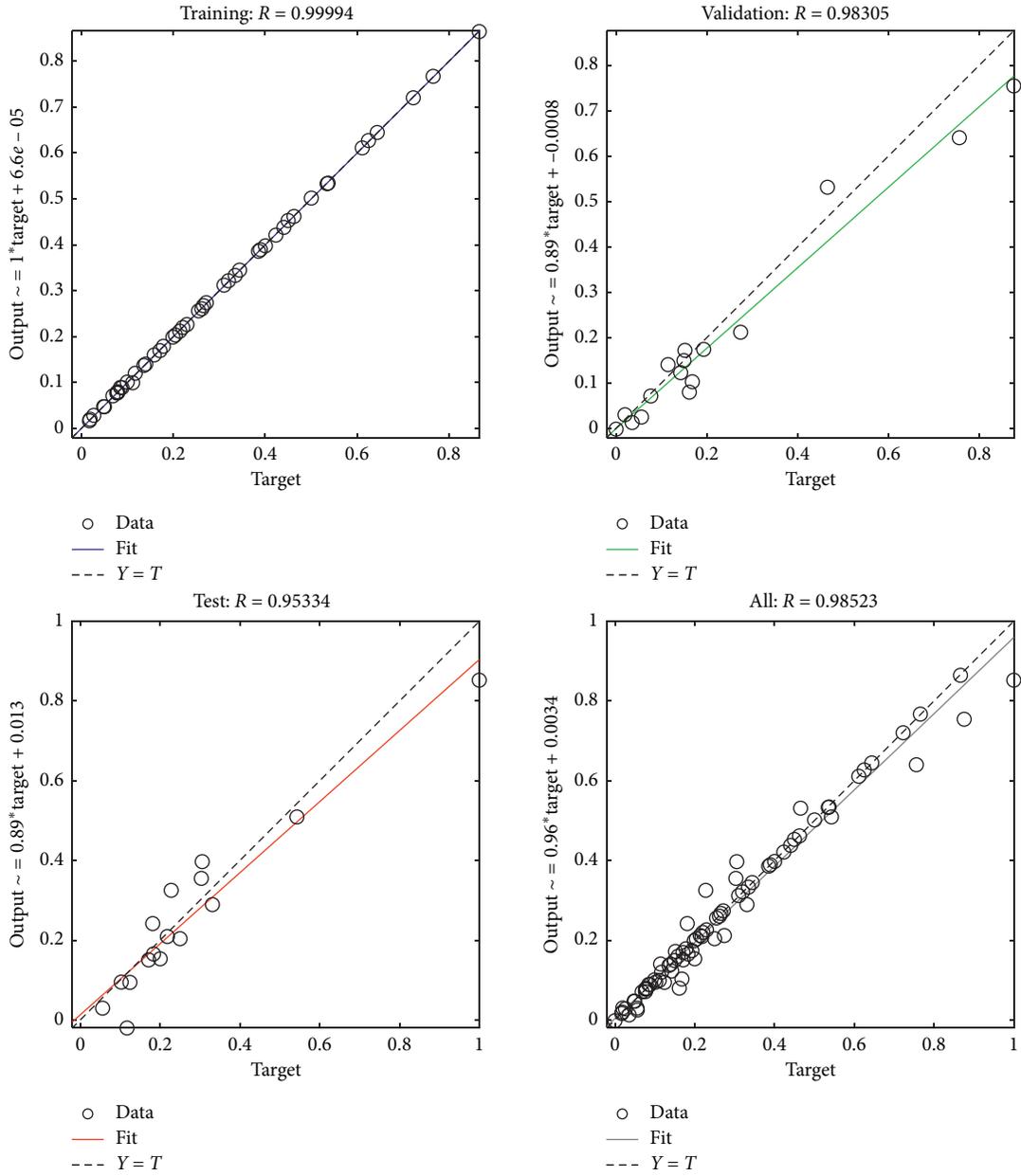


FIGURE 14: Error histogram (a) for deformation, (b) for stress, and (c) for strain energy.



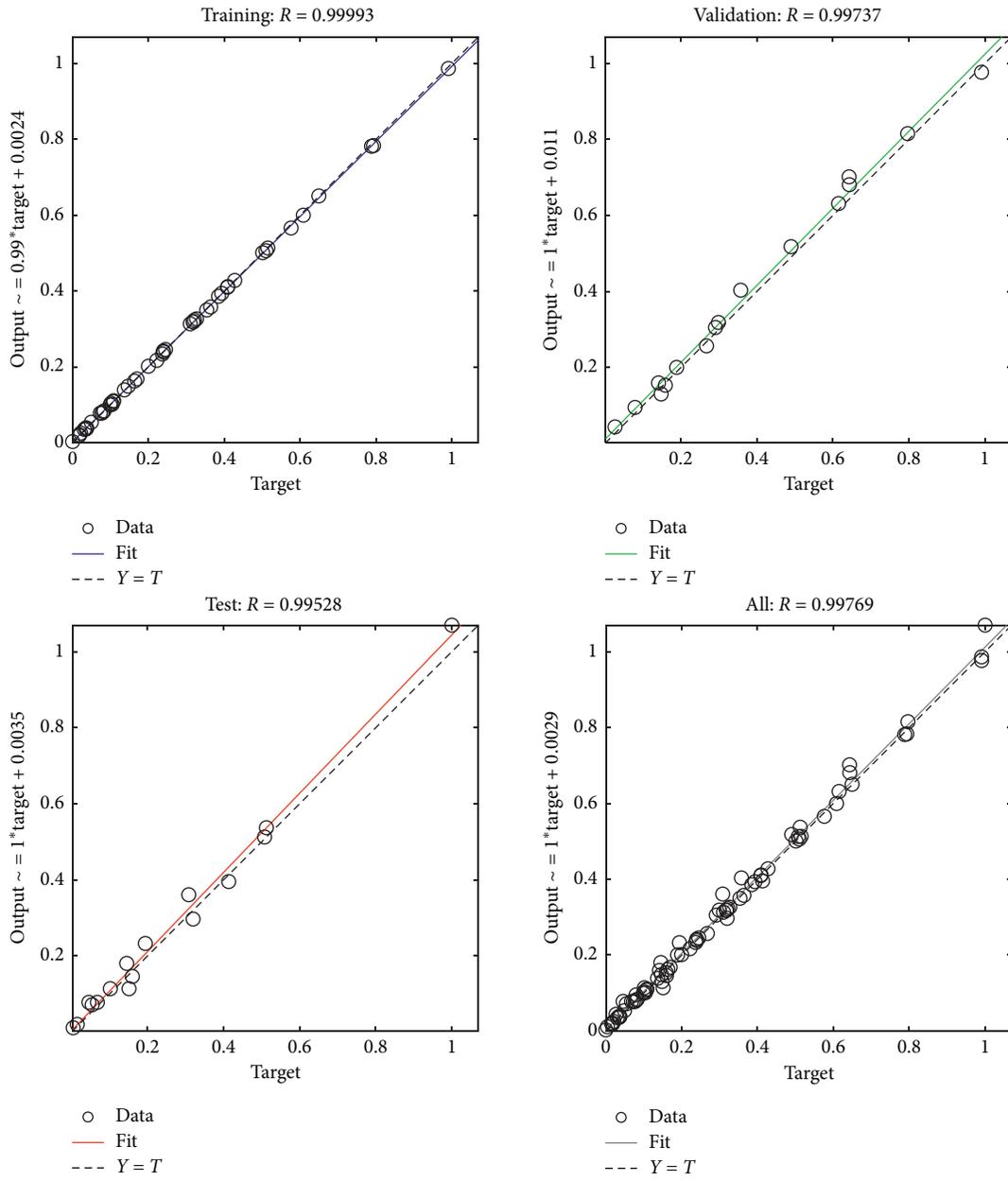
(a)

FIGURE 15: Continued.



(b)

FIGURE 15: Continued.



(c)

FIGURE 15: The fitness of model (a) for deformation, (b) for stress, and (c) for strain energy.

TABLE 16: Optimal results.

No.	WCA		ER-WCA	
	Strain energy (mJ)	Time (s)	Strain energy (mJ)	Time (s)
1	0.011509	2.62E+03	0.011509	3.05E+03
2	0.011509	2.59E+03	0.011509	3.08E+03
3	0.010654	2.63E+03	0.010654	2.84E+03
4	0.010654	2.63E+03	0.010654	2.90E+03
5	0.011127	2.62E+03	0.011127	2.96E+03
6	0.011314	2.63E+03	0.011314	3.61E+03
7	0.011364	2.63E+03	0.011364	3.56E+03
8	0.011509	2.62E+03	0.011509	3.57E+03
9	0.010717	2.63E+03	0.010717	3.17E+03
10	0.011509	2.63E+03	0.011509	3.07E+03
11	0.010675	2.65E+03	0.010675	3.03E+03
12	0.011509	2.33E+03	0.011509	2.90E+03
13	0.011509	2.67E+03	0.011509	2.98E+03
14	0.011509	2.67E+03	0.011509	3.05E+03
15	0.011509	2.90E+03	0.011509	3.02E+03
16	0.010716	2.68E+03	0.010716	3.05E+03
17	0.011509	2.68E+03	0.011509	2.89E+03
18	0.011509	2.72E+03	0.011509	2.85E+03
19	0.011509	2.85E+03	0.011509	2.94E+03
20	0.011509	2.80E+03	0.011509	2.85E+03
21	0.011257	2.95E+03	0.011257	2.86E+03
22	0.010708	2.69E+03	0.010708	2.93E+03
23	0.010656	2.94E+03	0.010656	3.02E+03
24	0.011509	2.94E+03	0.011509	3.06E+03
25	0.011509	2.96E+03	0.011509	3.09E+03
26	0.010675	2.63E+03	0.010675	3.19E+03
27	0.011509	2.65E+03	0.011509	3.05E+03
28	0.011509	2.68E+03	0.011509	3.05E+03
29	0.011509	2.95E+03	0.011509	3.05E+03
30	0.010716	2.96E+03	0.010716	2.84E+03
Average	0.01123	2717.755	0.01119	3057.928
Standard deviation	0.000373	149.2229	0.000568	202.8228

TABLE 17: Results of evaluation.

	Strain energy (mJ)	Deformation (mm)	Stress (MPa)	Life (10 <sup>6</sup> cycle)
Prediction	0.01123	33.666	79.050	
FEM	0.01102	34.236	81.475	299
Error (%)	1.87	1.69	3.06	

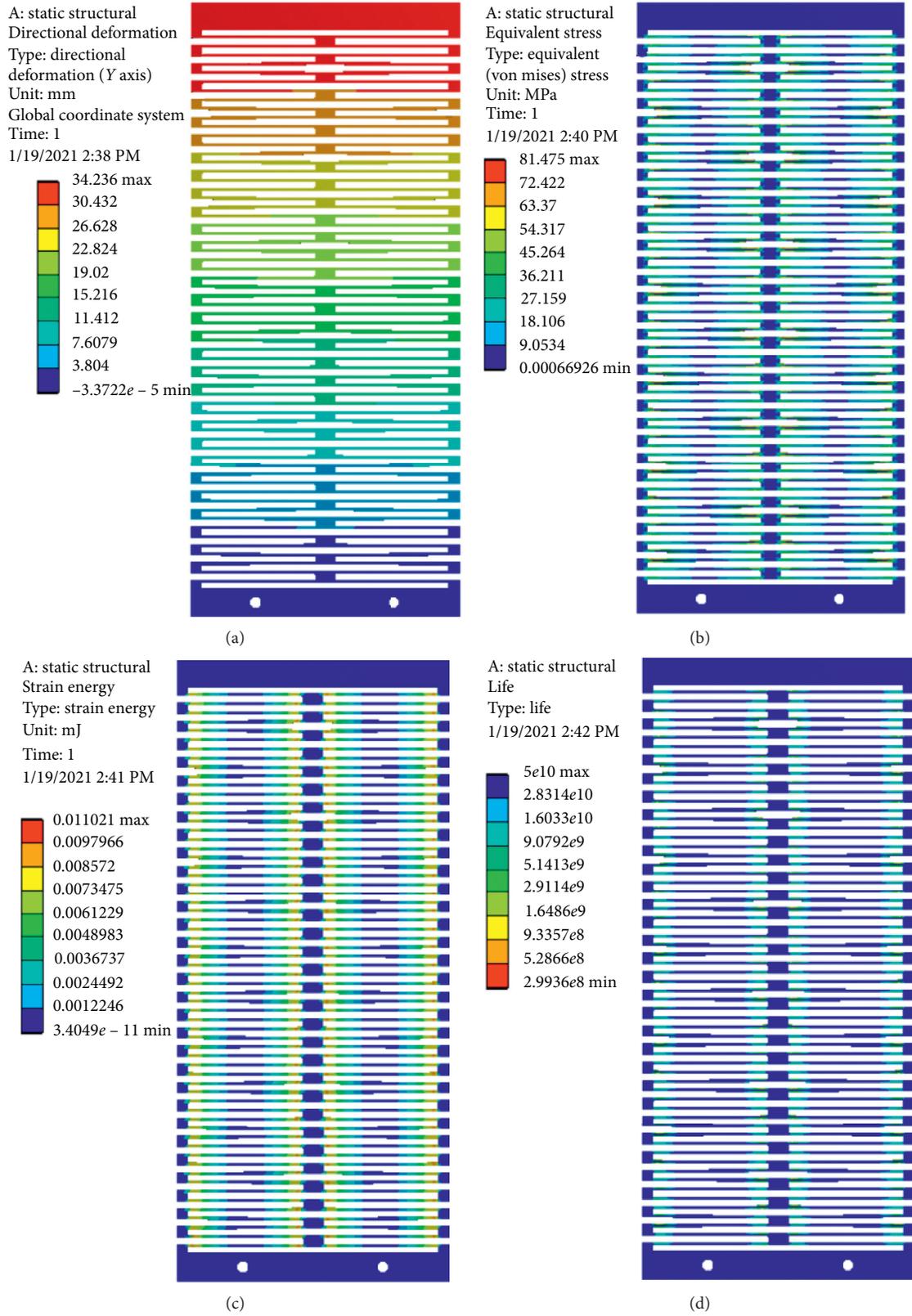


FIGURE 16: FEM results for (a) deformation, (b) stress, (c) strain energy, and (d) life.

small error values, it shows that the proposed optimal method has high reliability.

## 7. Conclusions

This paper proposes a method for the development and optimization of a planar spring used for a gravity balance mechanism. The proposed method is a combination of finite FEM, DFNN, and WCA. First, the FEM was used to collect data. Next, the approximated model was constructed using the DFNN, and finally, the geometry of the planar spring was optimized by the WCA. The effectiveness of the proposed method was tested by comparison with the optimal results of the ER-WCA. The comparison results show that the convergence speed and search stability of the proposed algorithm are better than the ER-WCA. The optimal parameters of PS are used to build 3D models. This model is used to be FEM. Comparing the FEM results with the optimal prediction results shows that the errors of energy strain, deformation, and stress are 1.87%, 1.69%, and 3.06%, respectively. This error shows that the proposed solution is highly robust. In addition, the life of the PS was also predicted with 299 million cycles. For future research, prototypes will be manufactured and measured to verify the numerical results. The optimization method will be utilized for other compliant mechanisms.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

The authors are thankful for the financial support from the HCMC University of Technology and Education, Vietnam, under Grant No. T2021-01NCS.

## References

- [1] S. K. Agrawal and A. Fattah, "Design of an orthotic device for full or partial gravity-balancing of a human upper arm during motion," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Cat. No. 03CH37453), pp. 2841–2846, Las Vegas, NV, USA, October 2003.
- [2] V. Arakelian, "Gravity compensation in robotics," *Advanced Robotics*, vol. 30, no. 2, pp. 79–96, 2016.
- [3] M. Uemura, Y. Mitabe, and S. Kawamura, "Simultaneous gravity and gripping force compensation mechanism for lightweight hand-arm robot with low-reduction reducer," *Robotica*, vol. 37, no. 6, pp. 1090–1103, 2019.
- [4] S. K. Agrawal, G. Gardner, and S. Pledgie, "Design and fabrication of an active gravity balanced planar mechanism using auxiliary parallelograms," *J. Mech. Des. Journal of Mechanical Design*, vol. 123, no. 4, pp. 525–528, 2001.
- [5] Y.-L. Chu and C.-H. Kuo, "A single-degree-of-freedom self-regulated gravity balancer for adjustable payload," *Journal of Mechanisms and Robotics*, vol. 9, 2017.
- [6] W. D.W. Van Dorsser, R. Barents, B. M.B. Wisse, M. Schenk, and J. L.J. Herder, "Energy-free adjustment of gravity equilibrators by adjusting the spring stiffness," *Proceedings of the institution of Mmechanical Eengineers, Part C: Journal of Mechanical Engineering Science*, vol. 222, no. 9, pp. 1839–1846, 2008.
- [7] B. M. Wisse, W. D. Van Dorsser, R. Barents, and J. L. Herder, "Energy-free adjustment of gravity equilibrators using the virtual spring concept," in *Proceedings of the 2007 IEEE 10th International Conference on Rehabilitation Robotics*, pp. 742–750, Noordwijk, The Netherlands, June 2007.
- [8] T. Rahman, R. Ramanathan, R. Seliktar, and W. Harwin, "A ssimple ttechnique to ppassively ggravity-bbalance aarticulated mmechanisms," *Journal of Mechanical Design*, vol. 117, no. 4, p. 655, 1995.
- [9] N. Le Chau, H.H. G. LeGiang Le, and T.-P. Dao, "A gGravity bBalance mMechanism uUsing cCompliant mMechanism," *Advances in Intelligent Systems and Computing*, in *Proceedings of the International Conference on Green Technology and Sustainable Development*, pp. 431–439, Nang City, Vietnam, November 2020.
- [10] N. L. Chau, H. G. Le, T.-P. Dao, M. P. Dang, and V. A. Dang, "Efficient hybrid method of FEA-based RSM and PSO algorithm for multi-objective optimization design for a compliant rotary joint for upper limb assistive device," *Mathematical Problems in Engineering*, vol. 2019, Article ID 2587373, 14 pages, 2019.
- [11] N. L.N. Le ChauChau, H. G. Le, T.-P. Dao, and V. A. Dang, "Design and optimization for a new compliant planar spring of upper limb assistive device using hybrid approach of RSM–FEM and MOGADesign and oOptimization for a nNew cCompliant pPlanar sSpring of uUpper lLimb aAssistive dDevice uUsing hHybrid aApproach of RSM-FEM and MOGA," *Arabian Journal for Science and Engineering*, vol. 44, no. 9, pp. 7441–7456, 2019.
- [12] L. L. Howell, *Compliant Mmechanisms*, John Wiley & Sons, Hoboken, NJ, USA, 2001.
- [13] K. J. Bathe, *Finite Eelement Mmethod*, Wiley Encyclopedia of Computer Science and Engineering, Hoboken, NJ, USA, 2007.
- [14] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Zhu, *The Finite Element Method*, Vol. 3, McGraw-Hill, London, UK, 1977.
- [15] N. Le Chau, T.-P. Dao, and V. A. Dang, "An efficient hybrid approach of improved adaptive neural fuzzy inference system and teaching learning-based optimization for design optimization of a jet pump-based thermoacoustic-Stirling heat engine," *Neural Computing and Applications*, vol. 32, no. 11, pp. 7259–7273, 2020.
- [16] N. Le Chau, N. T. Tran, and T.-P. Dao, "An oOptimal dDesign mMethod for cCompliant mMechanisms," *Mathematical Problems in Engineering*, vol. 2021, Article ID 5599624, 18 pages, 2021.
- [17] F.F. S. Gómez, R.R. L. Lorza, M.M. C. Bobadilla, and R.R. E. García, "Improving the process of adjusting the parameters of finite element models of healthy human intervertebral discs by the multi-response surface method," *Materials*, vol. 10, no. 10, p. 1116, 2017.
- [18] R. Lostado, R. F. Martinez, and B. J. Mac Donald, "Determination of the contact stresses in double-row tapered roller bearings using the finite element method, experimental

- analysis and analytical models,” *Journal of Mechanical Science and Technology*, vol. 29, no. 11, pp. 4645–4656, 2015.
- [19] M. F. Alghifari, T. S. Gunawan, and M. Kartiwi, “Speech emotion recognition using deep feedforward neural network,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 2, pp. 554–561, 2018.
- [20] J. Seo, J. Lee, and K. Kim, “Decoding of polar code by using deep feed-forward neural networks,” in *Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 238–242, Maui, HI, USA, March 2018.
- [21] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, “Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems,” *Computers & Structures*, vol. 110, pp. 110–111, pp. 151–166, 2012.
- [22] O. B. Haddad, M. Moravej, and H. A. Loáiciga, “Application of the water cycle algorithm to the optimal operation of reservoir systems,” *Journal of Irrigation and Drainage Engineering*, vol. 141, no. 5, Article ID 04014064, 2015.
- [23] W.W.-C. Chen, A.A. H. Lee, W.W.-J. Deng, and K.K.-Y. Liu, “The implementation of neural network for semiconductor PECVD process,” *Expert Systems with Applications*, vol. 32, no. 4, pp. 1148–1153, 2007.
- [24] S. Athreya and Y. Venkatesh, “Application of Taguchi method for optimization of process parameters in improving the surface roughness of lathe facing operation,” *International Refereed Journal of Engineering and Science*, vol. 1, pp. 13–19, 2012.
- [25] G. Sun, J. Fang, X. Tian, G. Li, and Q. Li, “Discrete robust optimization algorithm based on Taguchi method for structural crashworthiness design,” *Expert Systems with Applications*, vol. 42, no. 9, pp. 4482–4492, 2015.
- [26] W. Lin and Z. Ma, “Using Taguchi-Fibonacci search method to optimize phase change materials enhanced buildings with integrated solar photovoltaic thermal collectors,” *Energy*, vol. 106, pp. 23–37, 2016.
- [27] A. Sadollah, H. Eskandar, and J. H. Kim, “Water cycle algorithm for solving constrained multi-objective optimization problems,” *Applied Soft Computing*, vol. 27, pp. 279–298, 2015.
- [28] A. A. Heidari, R.R. A. AbbaspourAli Abbaspour, and A.A. R. JordehiRezaee Jordehi, “An efficient chaotic water cycle algorithm for optimization tasks,” *Neural Computing and Applications*, vol. 28, no. 1, pp. 57–85, 2017.
- [29] A. Sadollah, H. Eskandar, A. Bahreininejad, and J. H. Kim, “Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems,” *Applied Soft Computing*, vol. 30, pp. 58–71, 2015.