applied sciences

Article

A Flight Parameter-Based Aircraft Structural Load Monitoring Method Using a Genetic Algorithm Enhanced Extreme Learning Machine

Yanjun Zhang [1], Shancheng Cao [2,*], Bintuan Wang [1] and Zhiping Yin [3]

[1] AVIC the First Aircraft Institute, Xi'an 710089, China
[2] School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China
[3] School of Civil Aviation, Northwestern Polytechnical University, Xi'an 710072, China
* Correspondence: shancheng.cao@nwpu.edu.cn

Abstract: High-precision operational flight loads are essential for monitoring fatigue of individual aircraft and are usually determined by flight parameters. To tackle the nonlinear relationship between flight loads and flight parameters for more accurate prediction of flight loads, artificial neural networks have been widely studied. However, there are still two major problems, namely the training strategy and sensitivity analysis of the flight parameters. For the first problem, the gradient descent method is usually used, which is time-consuming and can easily converge to a local solution. To solve this problem, an extreme learning machine is proposed to determine the weights based on a Moore–Penrose generalized inverse. Moreover, a genetic algorithm method is proposed to optimize the weights between the input and hidden layers. For the second problem, a mean impact value (MIV) method is proposed to measure the sensitivity of the flight parameters, and the neuron number in the hidden layer is also optimized. Finally, based on the measured dataset of an aircraft, the proposed flight load prediction method is verified to be effective and efficient. In addition, a comparison is made with some well-known neural networks to demonstrate the advantages of the proposed method.

Keywords: flight loads; flight parameters; extreme learning machine; genetic algorithm

check for updates

Citation: Zhang, Y.; Cao, S.; Wang, B.; Yin, Z. A Flight Parameter-Based Aircraft Structural Load Monitoring Method Using a Genetic Algorithm Enhanced Extreme Learning Machine. *Appl. Sci.* **2023**, *13*, 4018. https://doi.org/10.3390/app13064018

Academic Editors: Fei Kang, Jun Teng and Yu Tang

Received: 2 February 2023
Revised: 12 March 2023
Accepted: 20 March 2023
Published: 22 March 2023

1. Introduction

Flight loads (shear, bending moment, and torque) are indispensable and important data for structural health monitoring in producing the aircraft load spectrum, determining the aircraft remaining service life, and analyzing the aircraft structural reliability, as shown in Figure 1 [1–3]. The accuracy of the flight load data directly affects the correct assessment of the aircraft health status. Therefore, the question of how to obtain highly accurate flight load data has always been the focus of fatigue life monitoring of individual aircraft [4–6]. Normally, operational flight load can be obtained by strain measurement [7,8] or indirect flight parameter prediction methods [9,10].

In the flight parameter prediction method, the local stress is determined by converting the flight parameters, while in the direct strain measurement method, the local stress is determined based on the strain gauges [11,12]. Since the flight load data are directly acquired in the strain measurement method, the results of flight fatigue life monitoring using this method are theoretically more accurate than those of the parameter identification method [13,14]. A measured bending moment series of an aircraft wing during flight is shown in Figure 2. However, there are difficulties in sticking, calibration, and wire routing of strain gauges, as well as shortcomings such as drift, easy damage, and short life, which hinder the widespread application of strain measurements in engineering [15].

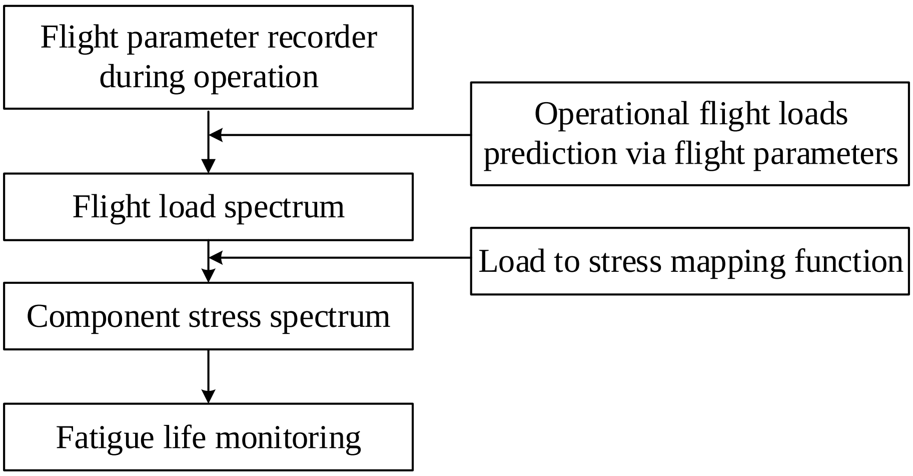
Flight parameter recorder during operation
Operational flight loads prediction via flight parameters
Flight load spectrum
Load to stress mapping function
Component stress spectrum
Fatigue life monitoring

Figure 1. Flowchart of individual aircraft fatigue life monitoring.

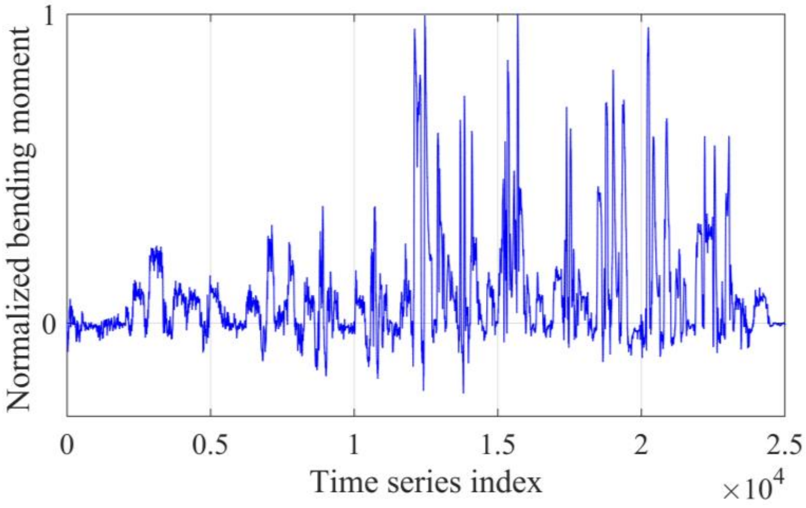

Figure 2. Demonstration of measured bending moment series of aircraft wing during flight.

Although the flight load determined by the flight parameter prediction method may degrade its accuracy, it is still an effective method that is widely used in practical engineering [16]. Currently, many advanced combat aircraft generally use a hybrid method that combines the flight parameter prediction and strain measurement method [17]. That is, most aircraft in a given fleet use flight-parameter-based flight load prediction monitoring, and a few aircraft have strain gauges attached to some key components. It is worth mentioning that the measured strain data are mainly used to establish, verify, and improve the flight-parameter-based flight load prediction method [18].

In the flight-parameter-based flight load prediction method, the multiple linear regression method is often used to establish the relationship between the flight parameters and loads, and then the operational flight load is determined based on the recorded flight parameters during the flight [19]. However, the prediction accuracy based on the multiple linear regression is low, and the nonlinear relationship between flight parameters and loads cannot be revealed and established [20,21]. To address this issue, various artificial neural networks have been studied to build flight load prediction models that solve the nonlinear mapping problem and improve the accuracy of the flight load prediction [22–24].

Based on the multilayer perceptron architecture, it was demonstrated that the correlation coefficient values between the predicted and measured load values of the key components of the outer and inner wings were 0.997 and 0.995, respectively, and the errors resulting from fatigue damage life were −3.1% and 2.7%, respectively [9,25]. Later, a model of flight parameters with fuselage frame and leading-edge flap loads was established based on neural network technology [3]. The average error between the life of the fuselage frame and leading-edge flap calculated by the predicted load and the measured life is 21% and 30%, respectively. In addition, a local model network was proposed to overcome the lack of transparency of conventional networks and the robust extrapolation behavior [16,26]. Based on this model, the flight loads of the bending and torsional moments on the vertical tail plane were accurately predicted using the flight parameters. From the above studies, it concludes that the traditional feedforward neural networks use the gradient descent method, which mainly has the following shortcomings and deficiencies: (1) the training speed is slow because the gradient descent method requires multiple iterations to achieve the purpose of correcting weights and thresholds; (2) it is easy to fall into a local minimum point and cannot promise reaching a global minimum; and (3) the choice of learning rate is sensitive, which has a great impact on the performance of the neural network, and a suitable learning rate must be chosen to obtain an ideal network [27–29].

To circumvent the above-mentioned problems, novel training algorithms are required, which aim at improving the performance of artificial neural networks [30–32]. In this paper, an improved extreme learning machine (ELM) is proposed. Compared with the common machine-learning methods, the ELM has the advantages of fast learning speed and good generalization performance [33–35]. In addition, the ELM has fewer tuning parameters and completes the training task without repeated optimization steps as it bypasses the iterative process [36–38]. Moreover, the ELM is a single-layer feedforward neural network and is popular for accurate classification or regression [39,40]. In the original ELM method, the weights between the input and hidden layers are randomly generated, which degrades its performance [40–42]. To solve this problem, the weights between the input and hidden layers are determined using a genetic algorithm (GA). The genetic algorithm, an optimization strategy based on natural selection and evolution, is well-suited to solve a variety of optimization problems [43]. Based on selection, mutation, and crossover operations, the GA can better explore the solution space. Additionally, techniques such as simulated annealing or tabu search can be also used to prevent the search from getting stuck in local optima [44,45].

Moreover, flight parameter recorders normally record tens or even hundreds of flight parameter data, but not all flight parameter data are suitable for building parameter-based flight load prediction models, which increases the complexity and time of neural networks [46]. Therefore, it is important to perform a sensitivity analysis of the flight parameter, which can help to determine the crucial flight parameters for flight load prediction. Moreover, for the proposed improved ELM, only the neuron number in the hidden layer needs to be determined, and cross-validation is used to evaluate the neuron number of the hidden layer.

With the aim of solving the above problems, the structure of this paper is as follows. First, an improved extreme learning machine method based on a genetic algorithm is presented in Section 2. In Section 3, a novel flight parameter sensitivity analysis method and a method for determining the neuron number of the hidden layer are proposed. In Section 4, the above-proposed methods are demonstrated to be effective and efficient. Finally, some important conclusions are drawn in Section 5.

2. Extreme Learning Machine Improved by Genetic Algorithm

The extreme learning machine is a typical single-layer feedforward neural network learning algorithm, as shown in Figure 3 [47,48]. The weight matrix $\mathbf{W}$ and offset $\mathbf{b}$ between the input and hidden layers of the traditional ELM neural network are randomly generated, and no adjustments are required in the subsequent computations [36]. In this case, only the neuron number of the hidden layer needs to be set to obtain the unique optimal solution.

In a traditional extreme learning machine, the weights between the input and hidden layers are randomly generated, and the weights between the hidden and output layers are efficiently determined based on the least-squares criterion instead of the traditional iterative gradient descent method. To improve the performance of the traditional ELM, the weights between the input and hidden layers are optimized using a genetic algorithm. More details are presented below.

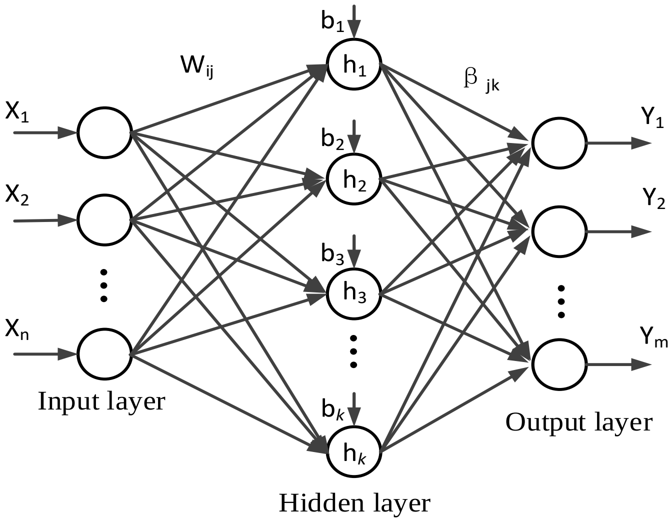

Figure 3. Illustration of single-layer feedforward neural network.

2.1. Extreme Learning Machine Neural Network

The structure of the neural network of an extreme learning machine is shown in Figure 3, which consists of an input layer, a hidden layer and an output layer. Suppose the input layer has n input variables, the hidden layer has K neurons, the output layer has m output variables and the input and output matrices of the training dataset have N samples. Then, the input and output of ELM are

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nN} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & \cdots & y_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mN} \end{bmatrix} \quad (1)$$

Assume that the connection weight matrix $\mathbf{W}$ between the input and hidden layers is

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K1} & w_{K2} & \cdots & w_{Kn} \end{bmatrix} \quad (2)$$

where w_{ji} represents the weight between the j-th neuron of the hidden layer and the i-th neuron of the input layer.

Suppose the connection weight matrix $\boldsymbol{\beta}$ of the output and hidden layers is

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{K1} & \beta_{K2} & \cdots & \beta_{Km} \end{bmatrix} \quad (3)$$

where β_{ji} represents the weight between the j-th neuron of the hidden layer and the i-th neuron of the output layer. In this case, the output of the ELM is $\mathbf{Z} = [z_1, z_2, \cdots, z_N]$

$$\boldsymbol{z}_j = \begin{bmatrix} z_{1j} \\ z_{2j} \\ \vdots \\ z_{mj} \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{K} \beta_{i1}\varphi(\boldsymbol{w}_i\boldsymbol{x}_j + b_i) \\ \sum\limits_{i=1}^{K} \beta_{i2}\varphi(\boldsymbol{w}_i\boldsymbol{x}_j + b_i) \\ \vdots \\ \sum\limits_{i=1}^{K} \beta_{im}\varphi(\boldsymbol{w}_i\boldsymbol{x}_j + b_i) \end{bmatrix} \tag{4}$$

in which $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \cdots, w_{in}]$, $\boldsymbol{x}_j = [w_{1j}, w_{2j}, \cdots, w_{nj}]^{\mathrm{T}}$, and j =1, 2,$\cdots$, N. Moreover, $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_K \end{bmatrix}^{\mathrm{T}}$ indicates the neuron offset of the hidden layer. In addition, the superscript T indicates the transpose operation. Equation (4) can be rearranged into

$$\mathbf{H\beta} = \mathbf{Z}^{\mathrm{T}} \tag{5}$$

in which $\mathbf{H}$ is the output matrix of the hidden layer and expressed as

$$\mathbf{H} = \begin{bmatrix} \varphi(\boldsymbol{w}_1 \cdot \boldsymbol{x}_1 + b_1) & \varphi(\boldsymbol{w}_2 \cdot \boldsymbol{x}_1 + b_2) & \cdots & \varphi(\boldsymbol{w}_K \cdot \boldsymbol{x}_1 + b_K) \\ \varphi(\boldsymbol{w}_1 \cdot \boldsymbol{x}_2 + b_1) & \varphi(\boldsymbol{w}_2 \cdot \boldsymbol{x}_2 + b_2) & \cdots & \varphi(\boldsymbol{w}_K \cdot \boldsymbol{x}_2 + b_K) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\boldsymbol{w}_1 \cdot \boldsymbol{x}_N + b_1) & \varphi(\boldsymbol{w}_2 \cdot \boldsymbol{x}_N + b_2) & \cdots & \varphi(\boldsymbol{w}_K \cdot \boldsymbol{x}_N + b_K) \end{bmatrix} \tag{6}$$

The ideal case is that the output of the network $\mathbf{Z}^{\mathrm{T}}$ is identical to $\mathbf{Y}^{\mathrm{T}}$, but in practice, the constraint condition is that $\|\mathbf{Y}^{\mathrm{T}} - \mathbf{Z}^{\mathrm{T}}\| < \varepsilon$ (ε is an arbitrary positive value). Then, the weight matrix $\boldsymbol{\beta}$ between the hidden and output layers is determined as

$$\min_{\beta} \|\mathbf{H\beta} - \mathbf{Y}^{\mathrm{T}}\| \tag{7}$$

Equation (7) can be effectively solved by a Moore–Penrose inverse according to the least-squares criterion, and the solution of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = \mathbf{H}^{+}\mathbf{Y}^{\mathrm{T}} \tag{8}$$

in which the superscript + represents the pseudoinverse operator.

For a given extreme learning machine with a fixed number of neurons in the hidden layer, the weight matrix $\boldsymbol{\beta}$ is uniquely determined without an iteration process, but the weight matrix $\mathbf{W}$ and offset vector $\mathbf{b}$ are randomly generated without optimization, which can severely affect the prediction accuracy and reliability. To overcome this issue, an improved ELM is proposed in which the weight matrix $\mathbf{W}$ and offset vector $\mathbf{b}$ are optimized by a genetic algorithm.

2.2. Genetic Algorithm Enhanced ELM

The genetic algorithm (GA) simulates the phenomena of replication, crossover, and mutation that occur in natural selection and inheritance [49]. Random selection, crossover, and mutation operations are used to generate a group of individuals that can better adapt to the environment, so that the group evolves to a better and better region in the search space [50]. Compared with the traditional optimization methods, the genetic algorithm mainly has the following characteristics:

1. The processing object of the GA is not the parameter itself of the optimization problem but the individual that encodes the parameter set;
2. The basic action object of the genetic algorithm is a set of multiple feasible solutions, not a single feasible solution;
3. The genetic algorithm only uses the fitness function value to evaluate the individual without the knowledge of the search space or other auxiliary information;

4. The genetic algorithm does not use deterministic rules but uses probabilistic transition rules to guide its search direction.

In the GA-enhanced ELM model, the genetic algorithm is used to optimize the weights of the ELM neural network model. In this work, the genetic algorithm is used to optimize and generate **W** and **b**. In this process, the fitness function is taken as the mean squared errors predicted by the ELM network, which is defined as

$$F = \sum_{ij} \left| z_{ij} - y_{ij} \right|^2 / N \tag{9}$$

in which z_{ij} is the output of the ELM network, y_{ij} is the practical value, and N is the sample number of the training set.

Finally, the optimization results of the genetic algorithm are returned to the ELM network, and the optimized **W** and **b** are used to establish the ELM model. The process is illustrated in Figure 4. In addition, the setting parameters of the genetic algorithm are tabulated in Table 1.

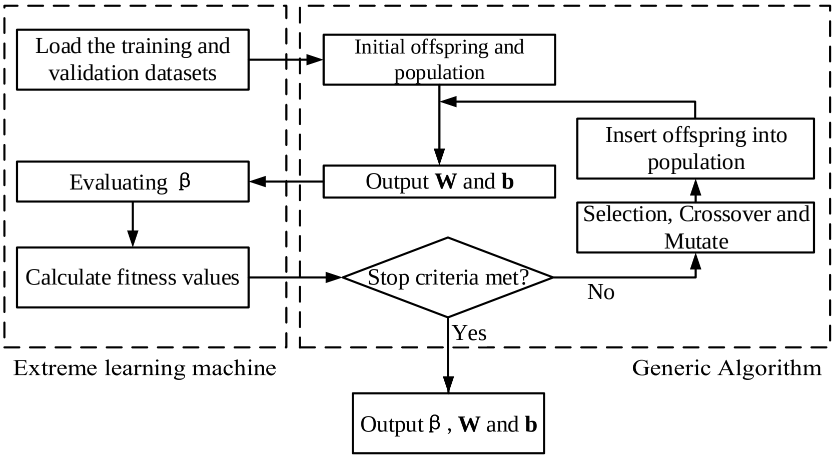
Load the training and validation datasets
Initial offspring and population
Insert offspring into population
Evaluating β
Output W and b
Selection, Crossover and Mutate
Calculate fitness values
Stop criteria met?
No
Yes
Extreme learning machine
Generic Algorithm
Output β , W and b

Figure 4. Flowchart of genetic-algorithm-enhanced extreme learning machine.

Table 1. Setting parameters of genetic algorithm.

Parameter Type	Parameter Setting
Population size	20
Max generation	200
Selection approach	Roulette wheel
Cross probability	0.4
Mutation probability	0.2
Stopping criteria	Maximum iteration steps

3. Determination of Input Parameters and Neuron Number of the Hidden Layer

With the connection weights **W**, **b,** and **β** of the ELM calculated in Section 2, the essential problem here is to determine the number of input flight parameters and the neuron number in the hidden layer. As for the number of input parameters, it can be determined by conducting flight parameter sensitivity analysis, and a mean impact value method is proposed to achieve this. As for the neuron number of the hidden layer, a cross-validation method is adopted to determine the proper value of K.

3.1. Mean Impact Value Method

The mean impact value (MIV) method is an effective approach to estimating the impact of the inputs on the outputs. The basic principle is to change the values of input flight parameters one by one while the trained neural network is fixed, and then calculate the corresponding prediction differences. Using the vertical load factor n_z as an example,

n_z will be increased by 10% ($1.1n_z$) and decreased by 10% ($0.9n_z$), respectively, while the other flight parameters remain unchanged. After this, $1.1n_z$ and $0.9n_z$ are put back into the training samples to replace the original n_z, and the corresponding outputs z_{ij}^{+} and z_{ij}^{-} of the neural networks are obtained. The mean impact value is calculated as the average value of their differences. Moreover, the mean impact values of different input parameters are normalized according to Equation (11).

$$MIV_i = \frac{1}{N}\sum_{j=1}^{N}\left(z_{ij}^{+} - z_{ij}^{-}\right) \tag{10}$$

$$C_i = |MIV_i| / \sum_{i=1}^{n}|MIV_i| \tag{11}$$

3.2. Determination of Neuro Number K in the Hidden Layer

For neural networks with a single hidden layer, the optimal prediction model can be easily determined based on the method proposed above for a given number of neurons of the hidden layer. Therefore, the number of neurons K in the hidden layer is an important parameter to be determined. When the number of neurons in the hidden layer is equal to the number of samples in the training set, the neural network can approach the output value of the training sample with zero residual errors for any weight and threshold of the input and hidden layers [29]. However, the generalization performance is the worst in this case. To reduce the computation cost and increase the generalization performance of the model, the number of neurons in the hidden layer is generally smaller than the number of training samples.

To determine the proper neuron number K in the hidden layer, the cross-validation (CV) method is adopted. CV is an effective method for evaluating model performance, which can avoid the occurrence of overlearning and under-learning in neural network models. In this method, the training dataset is divided into $\boldsymbol{L}$ groups, and each subset of data will be used as a verification dataset. With the rest of the subset data adopted as a training set, the neural network model is obtained, and the mean squared residual errors of the predicted flight loads of the verification set will be used as an indicator. The number of neurons in the hidden layer that corresponds to the minimum mean squared residual errors will be adopted. The flowchart of this process is shown in Figure 5.

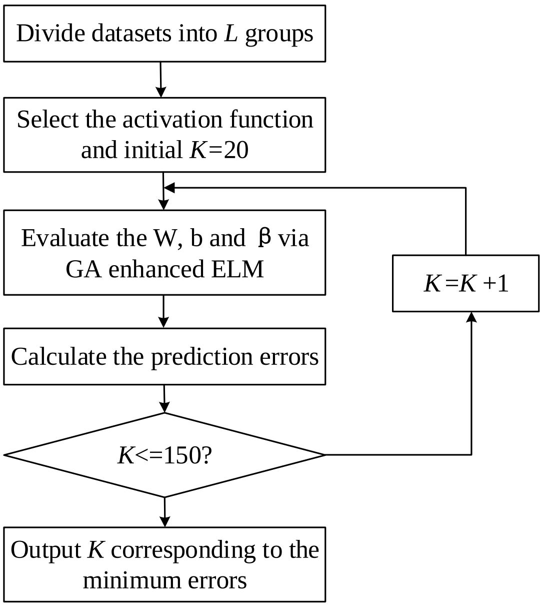
Divide datasets into L groups
Select the activation function and initial K=20
Evaluate the W, b and β via GA enhanced ELM
K=K +1
Calculate the prediction errors
K<=150?
Output K corresponding to the minimum errors

Figure 5. Flowchart of determining proper neuron number K in the hidden layer.

4. Flight Load Prediction Case Study

Taking the flight recorder data of an aircraft loop maneuver as the studying dataset, 14 loop maneuvers with a flight parameter size of 5645 × 13 are used as the training dataset, and 2 loop maneuvers with a flight parameter size of 851 × 13 are used as the testing dataset. To remove the outlier data points and reduce the effects of measurement noise, the original dataset is first processed by smooth filtering, and 13 flight parameters are initially adopted as input parameters, namely the altitude, Mach number, angle of attack, vertical load factor, roll rate, pitch rate, roll angle, angle of right outer aileron, angle of left front flap, angle of left outer aileron, angle of right inner aileron, angle of left inner aileron, and pitch angle. The wing bending moment is adopted as the output for the prediction. Based on a loop maneuver, the altitude, Mach number, vertical load factor, and wing bending moment are shown in Figure 6.

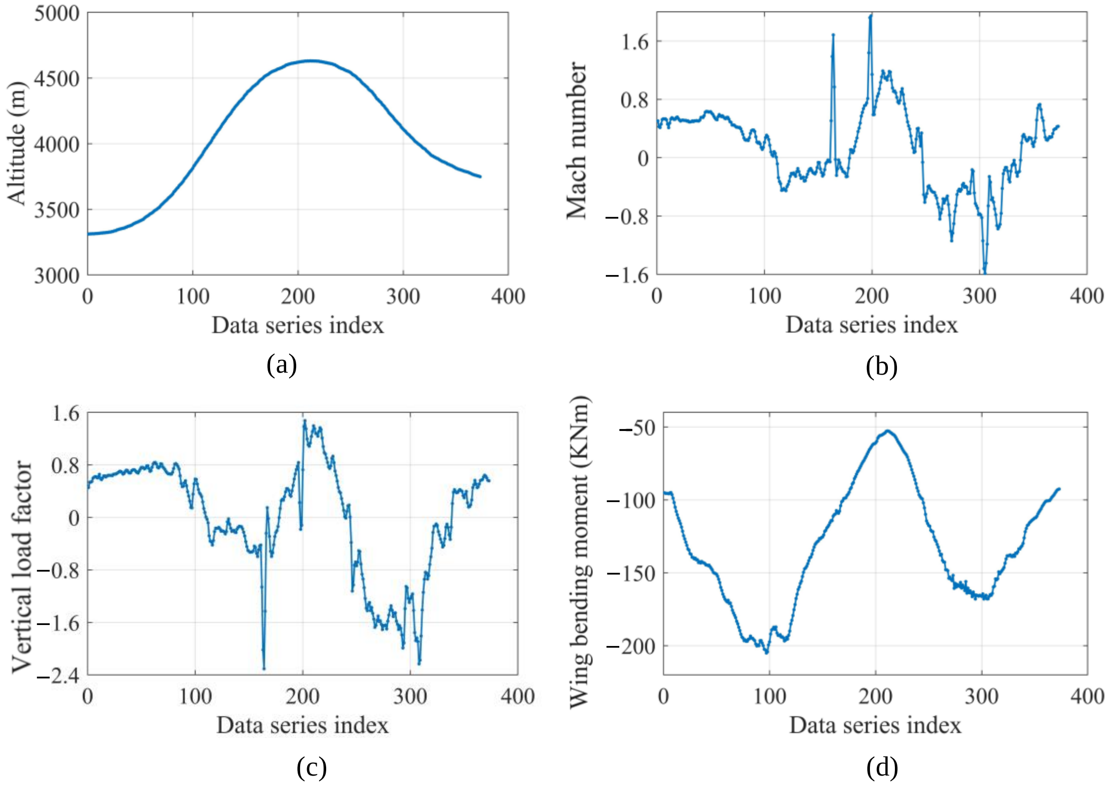

Figure 6. Demonstration of three flight parameters and wing bending moment of typical loop maneuver: (**a**) altitude, (**b**) Mach number, (**c**) vertical load factor, and (**d**) wing bending moment.

4.1. Investigation of Different Activation Functions

It is well-known that the performance of the artificial neural network varies with different activation function $\varphi(\cdot)$. Therefore, it is important to choose the proper activation function for more accurate prediction of flight loads. In this section, 5 different activation functions are investigated, namely the sigmoid function, sine function, hard-limit function, hyperbolic tangent function, and Morlet wavelet function (see Figure 7).

Based on the traditional extreme learning machine with 13 input flight parameters and 100 hidden layer neurons, different activation functions were investigated, and the accuracy of the wing bending moment prediction using the testing data is shown in Table 2.

From Table 2, it can be seen that the ELM achieves high prediction accuracy of the wing bending moment when the activation function is sigmoid and hyperbolic tangent function. In addition, the predicted wing bending moments based on the sigmoid activation function are presented in Figure 8 to compare with the actual measured wing bending moment.

Figure 8b shows that the correlation coefficient between the predicted and actual measured wing bending moments is 0.9991, indicating high prediction accuracy.

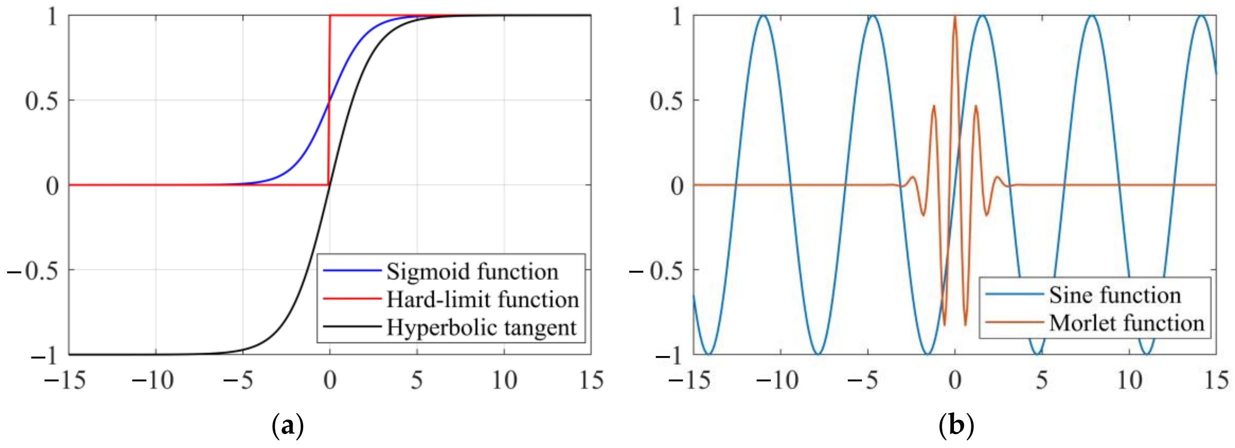

Figure 7. Demonstration of five investigated activation functions: (**a**) sigmoid function, hard-limit function, and hyperbolic tangent function; (**b**) sine function and Morlet wavelet function.

Table 2. Prediction accuracy of wing bending moments under different activation functions.

Activation Functions	Mean Relative Errors	Maximum Relative Errors	Mean Squared Errors $(\text{KN}\cdot\text{m})^2$
Sigmoid function	1.0546%	6.3417%	2.9702
Sine function	1.3321%	8.2345%	4.3269
Hard-limit function	6.3217%	32.0980%	86.852
Hyperbolic tangent function	1.1769%	4.5714%	3.1849
Morlet wavelet function	8.3282%	61.873%	144.51

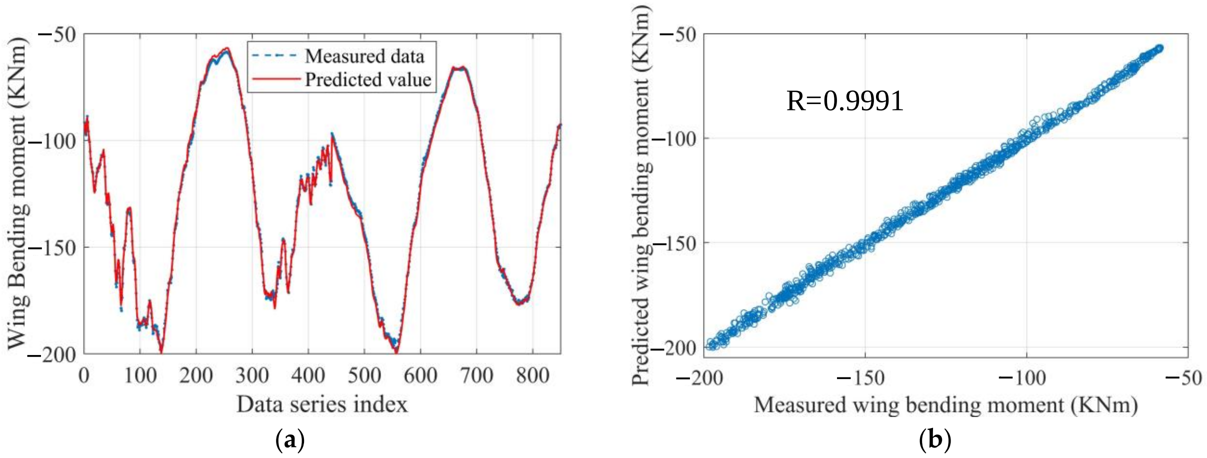

Figure 8. Wing bending moment prediction based on ELM (φ: sigmoid function; K = 100): (**a**) predicted value series; (**b**) the correlation coefficient between measured and predicted values.

4.2. Genetic Algorithm Enhanced ELM

Given 100 neurons of the hidden layer, the weight matrix $\mathbf{W}$ and offset $\mathbf{b}$ are optimized by a genetic algorithm. With the sigmoid activation function and parameter setting of the genetic algorithm in Table 1, the optimization process based on the training dataset is shown in Figure 9. In addition, a sensitivity analysis of the population size, crossover probability, and mutation probability is conducted, which confirms that the parameter setting in Table 1 is suitable for solving the proposed optimization problem.

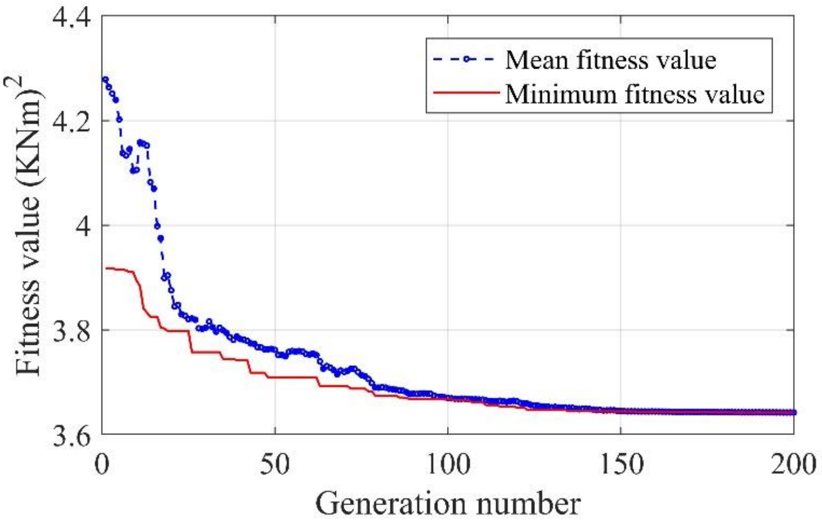

Figure 9. Optimization process of genetic algorithm.

In Figure 9, the mean and minimum fitness values of each generation based on the training dataset are plotted, and they gradually converge to the same value. After 200 generations, the optimal weight matrix **W** and offset **b** are obtained, and based on the testing dataset, the mean relative error, mean squared error, and maximum relative error between the predicted wing bending moments and measured ones are shown in Table 3.

Table 3. Flight load prediction accuracy via genetic-algorithm-enhanced ELM.

Activation Function	Mean Relative Errors	Maximum Relative Errors	Mean Squared Errors $(\mathbf{KN{\cdot}m})^2$
Sigmoid function	0.8376%	4.8627%	2.0044
Sine function	0.9163%	5.0318%	2.0247
Hard-limit function	4.7941%	31.943%	54.027
Hyperbolic tangent function	0.9799%	5.1406%	2.41
Morlet wavelet function	3.8387%	62.962%	37.824

In addition, based on the same optimization procedure, the weight matrix **W** and offset vector **b** are also optimized for other activation functions, whose mean relative error, mean squared error, and maximum relative error based on the testing dataset are presented in Table 3 as well. In comparison with Table 2, the mean relative and mean squared errors are all decreased, which manifests that the prediction accuracy of the wing bending moment has been improved by using the GA-enhanced ELM method. In this case, it shows that the optimization of **W** and **b** by using the genetic algorithm can effectively boost the performance of the ELM.

4.3. Determination of Neuron Number **K** of Hidden Layer

In the traditional ELM, the weight matrix **W** and offset vector **b** are randomly generated, which affects the performance of the ELM and makes it difficult to optimize the neuron number K in the hidden layer. In this case, the proposed GA-enhanced ELM is adopted to test the prediction performance for different neuron number K, and the results are graphed in Figure 10. Moreover, the activation function is set as a sigmoid function.

From Figure 10, as the number of neurons K increases, the prediction accuracy initially decreases, but after 80 neurons, the prediction accuracy reaches a stable level. To reduce the complexity and computational effects, the neuron number K is suggested to set as 80. However, in order to promise a high prediction accuracy, a large K such as 100 is also a good choice.

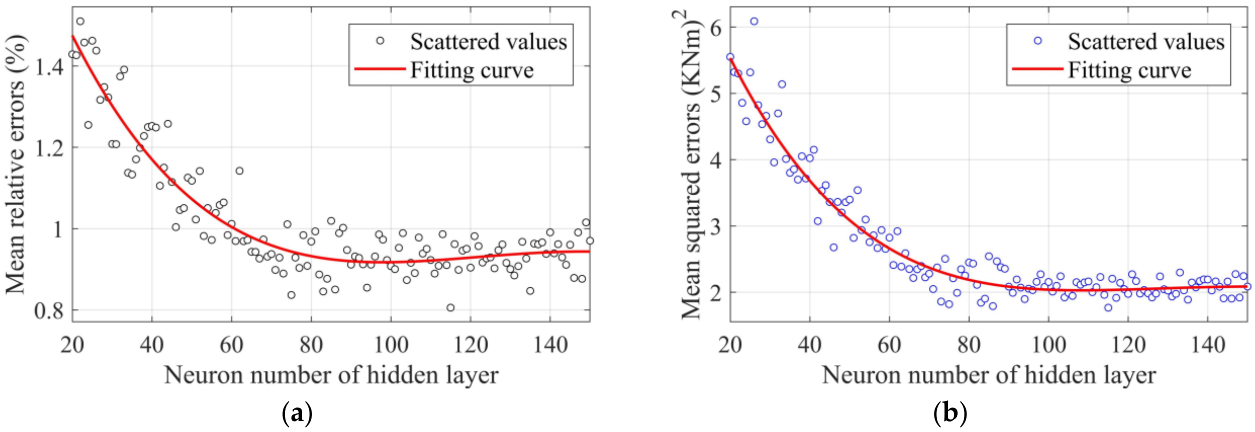

Figure 10. Wing bending moment prediction accuracy against different neuron number *K* of hidden layer: (**a**) mean relative errors and (**b**) mean squared errors.

4.4. Sensitivity Analysis of Flight Parameters

Based on the proposed mean impact value method and trained GA-enhanced ELM neural network, the MIVs of the 13 flight parameters are calculated and shown in Table 4.

Table 4. Normalized mean impact values of 13 flight parameters.

Flight Parameter	MIV	Flight Parameter	MIV
Altitude	0.0390	Angle of right outer aileron	0.0312
March number	0.0826	Angle of left front flap	0.0450
Angle of attack	0.1868	Angle of left outer aileron	0.0441
Vertical load factor	0.4449	Angle of right inner aileron	0.0648
Roll rate	0.0255	Angle of left inner aileron	0.0141
Pitch rate	0.0064	Pitch angle	0.0028
Roll angle	0.0127	-	-

Table 4 shows that the vertical load factor and angle of attack are strongly correlated with the wing bending moment, while the pitch angle and rate are almost uncorrelated with the wing bending moment. By removing the pitch rate and pitch angle from the input flight parameters, the mean relative error, mean squared error, and maximum relative error based on the testing dataset are 0.90858%, 1.9973, and 4.8335%, respectively, which demonstrates that the prediction accuracy is almost not affected by removing these two flight parameters.

4.5. Comparison with Other Neural Networks

In order to compare the prediction accuracy with other neural networks, a comparison study is conducted in this section. Apart from the tradition ELM, five well-known neural networks are studied, which are the backpropagation (BP) neural network, radial basis function (RBF) neural network, support vector machines (SVMs), long short-term memory (LSTM) neural network, and gate recurrent unit (GRU) neural network. In all neural networks, a single hidden layer is used as illustrated in Figure 3, and the neuron number in the hidden layer is set as 100. Moreover, all neural networks are trained by the same training dataset using a 5-cross-validation method, and the prediction accuracy for the testing dataset is tabulated in Table 5. In addition, to avoid unfair comparisons, BP, RBF, SVM, LSTM, and GRU are coded based on the neural network toolbox of MATLAB software, where the hyperparameters of those neural networks are set in an optimal way. It is worth noting that the training time of the ELM is 0.12 s, which is the shortest among the studied methods for the proposed problem in this work. Meanwhile, for the GA-enhanced ELM, it is the longest with a time of 164.30 s due to the optimization process of the generic algorithm with 200 generations.

Table 5. Prediction accuracy of different neural networks.

Neural Networks	Mean Relative Errors	Maximum Relative Errors	Mean Squared Errors $(\mathbf{KN{\cdot}m})^2$	Training Time (s)
BP	2.7909%	15.5331%	13.7988	7.2818
RBF	2.3006%	11.093%	13.769	3.6208
SVM [51]	2.2698%	9.3018%	13.3625	0.3033
LSTM [52]	1.3601%	23.2184%	4.6971	58.5525
GRU [53]	1.2323%	5.4111%	3.4670	58.5631
ELM	1.0546%	6.3417%	2.9702	0.1237
GA-enhanced ELM	0.8376%	4.8627%	2.0044	164.3014

From Table 5, it indicates that the LSTM and GRU networks outperform the traditional BP, RBF, and SVM. The reason is that the LSTM and GRU are both recurrent neural networks, which are good at extracting the sequence data features. However, the prediction accuracy of the LSTM and GRU is still lower when compared with that of the proposed GA-enhanced ELM method. The reason is that the Moore–Penrose inverse operation of the ELM is robust to the outlier values or measurement noise of the dataset, which improves the accuracy and noise robustness of the established nonlinear mapping relationship. Moreover, the proposed genetic algorithm further enhances the prediction accuracy of the ELM method by optimizing the weights between the input and hidden layers. However, the performance of the traditional ELM is outstanding as well, and it takes less training time compared with the GA-enhanced ELM. In addition, the GRU possesses high prediction accuracy over the LSTM neural network, which indicates that the flight-parameter-based flight load prediction problem is not a long-term time-series dependency problem.

5. Conclusions

An extreme learning machine (ELM) is investigated in this paper for better flight-parameter-based flight load prediction, which determines the connection weights based on the least-squares criterion instead of using the iterative gradient descent method. Moreover, a genetic algorithm is proposed to optimize the randomly generated weights between the input and hidden layers, which improves the prediction accuracy of the ELM method. In addition, a mean impact value method is adopted to select the flight parameters, and the neuron number of the hidden layer is effectively determined. Some other major conclusions are as follows:

(1) The ELM method is an effective tool to establish a flight-parameter-based flight load prediction model even when some weights are randomly generated.

(2) A genetic algorithm is an effective global optimization algorithm, which can be used to optimize the parameters of the neural networks.

(3) The mean impact value method can effectively select the highly correlated flight parameters for accurate flight load prediction, and it is applicable to all neural networks.

(4) The neuron number in the hidden layer should not be too small and is suggested to be a little large as long as the computation cost is affordable.

However, the Moore–Penrose inverse can be computationally expensive when dealing with very large datasets, and the optimized solutions of the genetic algorithm are not guaranteed to be globally optimal.

In the future, other efficient optimization algorithms can be explored to avoid the high computation cost of a genetic algorithm. Moreover, how to consider the time dependency property of a training dataset in the extreme learning machine is a promising research direction.

Author Contributions: Conceptualization, Y.Z. and S.C.; methodology, Y.Z.; validation, B.W. and S.C.; writing—original draft preparation, Y.Z.; writing—review and editing, S.C.; visualization, B.W.; supervision, Z.Y.; project administration, Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 12102346.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Kaneko, H.; Furukawa, T. Operational loads regression equation development for advanced fighter aircraft. In Proceedings of the ICAS 24th International Congress of the Aeronautical Sciences, Yokohama, Japan, 29 August–3 September 2004; pp. 1–9.
2. Zhu, S.; Wang, Y. Scaled sequential threshold least-squares (s2tls) algorithm for sparse regression modeling and flight load prediction. *Aerosp. Sci. Technol.* **2019**, *85*, 514–528. [CrossRef]
3. Tikka, J. Flight parameter based fatigue analysis approach for a fighter aircraft. *Aeronaut. J.* **2008**, *112*, 79–91. [CrossRef]
4. Raab, C.; Rohde-Brandenburger, K. Dynamic flight load measurements with mems pressure sensors. *CEAS Aeronaut. J.* **2021**, *12*, 737–753. [CrossRef]
5. Rui, J.; Xiaofan, H.; Yuhai, L. Individual aircraft life monitoring: An engineering approach for fatigue damage evaluation. *Chin. J. Aeronaut.* **2018**, *31*, 727–739.
6. de Camargo Branco, D.; de Silva Bussamra, F.L. Fatigue life monitoring system for aircraft to flexibilize operations and maintenance planning. *J. Aircr.* **2016**, *53*, 1298–1304. [CrossRef]
7. Jeong, S.H.; Lee, K.B.; Ham, J.H.; Kim, J.H.; Cho, J.Y. Estimation of maximum strains and loads in aircraft landing using artificial neural network. *Int. J. Aeronaut. Space Sci.* **2020**, *21*, 117–132. [CrossRef]
8. Nicolas, M.J.; Sullivan, R.W.; Richards, W.L. Large scale applications using fbg sensors: Determination of in-flight loads and shape of a composite aircraft wing. *Aerospace* **2016**, *3*, 18. [CrossRef]
9. Reed, S. Development of a parametric-based indirect aircraft structural usage monitoring system using artificial neural networks. *Aeronaut. J.* **2007**, *111*, 209–230. [CrossRef]
10. Sharan, A.; Vijayaraju, K.; James, D. Synthesis of in-flight strains using flight parameters for a fighter aircraft. *J. Aircr.* **2013**, *50*, 469–477. [CrossRef]
11. Dziendzikowski, M.; Kurnyta, A.; Reymer, P.; Kurdelski, M.; Klysz, S.; Leski, A.; Dragan, K. Application of operational load monitoring system for fatigue estimation of main landing gear attachment frame of an aircraft. *Materials* **2021**, *14*, 6564. [CrossRef]
12. Park, C.Y.; Ko, M.-G.; Kim, S.-Y.; Ha, J.-S. Flight test applications of an improved operational load monitoring device. *Int. J. Aeronaut. Space Sci.* **2020**, *21*, 970–983. [CrossRef]
13. Lin, M.; Guo, S.; He, S.; Li, W.; Yang, D. Structure health monitoring of a composite wing based on flight load and strain data using deep learning method. *Compos. Struct.* **2022**, *286*, 115305. [CrossRef]
14. Coates, C.W.; Thamburaj, P. Inverse method using finite strain measurements to determine flight load distribution functions. *J. Aircr.* **2008**, *45*, 366–370. [CrossRef]
15. Zhang, S.; Yang, J.; Li, Y.; Li, J. Identification of bearing load by three section strain gauge method: Theoretical and experimental research. *Measurement* **2013**, *46*, 3968–3975. [CrossRef]
16. Halle, M.; Thielecke, F.; Lindenau, O. Comparison of real-time flight loads estimation methods. *CEAS Aeronaut. J.* **2014**, *5*, 501–513. [CrossRef]
17. Cheung, C.; Valdés, J.J.; Li, M. Exploration of flight state and control system parameters for prediction of helicopter loads via gamma test and machine learning techniques. In *Real World Data Mining Applications*; Springer: Cham, Switzerland, 2015; pp. 359–385.
18. Kim, J.-H.; Park, Y.; Kim, Y.-Y.; Shrestha, P.; Kim, C.-G. Aircraft health and usage monitoring system for in-flight strain measurement of a wing structure. *Smart Mater. Struct.* **2015**, *24*, 105003. [CrossRef]
19. Candon, M.; Esposito, M.; Fayek, H.; Levinski, O.; Koschel, S.; Joseph, N.; Carrese, R.; Marzocca, P. Advanced multi-input system identification for next generation aircraft loads monitoring using linear regression, neural networks and deep learning. *Mech. Syst. Signal Process.* **2022**, *171*, 108809. [CrossRef]
20. Montel, M.; Thielecke, F. Efficient and accurate technology for aircraft loads estimation. *CEAS Aeronaut. J.* **2020**, *11*, 461–474. [CrossRef]
21. Krings, M.; Thielecke, F. A predictive envelope protection system using linear, parameter-varying models. *CEAS Aeronaut. J.* **2015**, *6*, 95–108. [CrossRef]

22. Dong, Y.; Tao, J.; Zhang, Y.; Lin, W.; Ai, J. Deep learning in aircraft design, dynamics, and control: Review and prospects. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2346–2368. [CrossRef]
23. Cao, X.; Sugiyama, Y.; Mitsui, Y. Application of artificial neural networks to load identification. *Comput. Struct.* **1998**, *69*, 63–78. [CrossRef]
24. Li, H.; Zhang, Q.; Chen, X. Deep learning-based surrogate model for flight load analysis. *Comput. Model. Eng. Sci.* **2021**, *128*, 605–621. [CrossRef]
25. Reed, S.C. Indirect aircraft structural monitoring using artificial neural networks. *Aeronaut. J.* **2008**, *112*, 251–265. [CrossRef]
26. Halle, M.; Thielecke, F. Local model networks applied to flight loads estimation. In Proceedings of the 31st Congress of the International Council of the Aeronautical Sciences, Belo Horizonte, Brazil, 9–14 September 2018; pp. 1–10.
27. Van Gerven, M.; Bohte, S. *Artificial Neural Networks as Models of Neural Information Processing*; Frontiers Media, S.A.: Lausanne, Switzerland, 2017; p. 114.
28. Guliyev, N.J.; Ismailov, V.E. On the approximation by single hidden layer feedforward neural networks with fixed weights. *Neural Netw.* **2018**, *98*, 296–304. [CrossRef]
29. Cao, W.; Wang, X.; Ming, Z.; Gao, J. A review on neural networks with random weights. *Neurocomputing* **2018**, *275*, 278–287. [CrossRef]
30. Huang, G.; Huang, G.-B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *61*, 32–48. [CrossRef] [PubMed]
31. Laudani, A.; Lozito, G.M.; Fulginei, F.R.; Salvini, A. On training efficiency and computational costs of a feed forward neural network: A review. *Comput. Intell. Neurosci.* **2015**, *2015*, 83. [CrossRef]
32. Ojha, V.K.; Abraham, A.; Snášel, V. Metaheuristic design of feedforward neural networks: A review of two decades of research. *Eng. Appl. Artif. Intell.* **2017**, *60*, 97–116. [CrossRef]
33. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef]
34. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
35. Huang, H.-X.; Li, J.-C.; Xiao, C.-L. A proposed iteration optimization approach integrating backpropagation neural network with genetic algorithm. *Expert Syst. Appl.* **2015**, *42*, 146–155. [CrossRef]
36. Huang, G.-B.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B* **2011**, *42*, 513–529. [CrossRef] [PubMed]
37. Tang, J.; Deng, C.; Huang, G.-B. Extreme learning machine for multilayer perceptron. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 809–821. [CrossRef] [PubMed]
38. Zhao, H.; Liu, H.; Xu, J.; Deng, W. Performance prediction using high-order differential mathematical morphology gradient spectrum entropy and extreme learning machine. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 4165–4172. [CrossRef]
39. Chen, H.; Zhang, Q.; Luo, J.; Xu, Y.; Zhang, X. An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2020**, *86*, 105884. [CrossRef]
40. Pan, Z.; Meng, Z.; Chen, Z.; Gao, W.; Shi, Y. A two-stage method based on extreme learning machine for predicting the remaining useful life of rolling-element bearings. *Mech. Syst. Signal Process.* **2020**, *144*, 106899. [CrossRef]
41. Wang, J.; Lu, S.; Wang, S.-H.; Zhang, Y.-D. A review on extreme learning machine. *Multimed. Tools Appl.* **2022**, *81*, 41611–41660. [CrossRef]
42. Chen, Z.; Gryllias, K.; Li, W. Mechanical fault diagnosis using convolutional neural networks and extreme learning machine. *Mech. Syst. Signal Process.* **2019**, *133*, 106272. [CrossRef]
43. Feng, K.; Ji, J.; Ni, Q. A novel adaptive bandwidth selection method for vold–kalman filtering and its application in wind turbine planetary gearbox diagnostics. *Struct. Health Monit.* **2022**, *22*, 1027–1048. [CrossRef]
44. Lee, S.; Kim, S.B. Parallel simulated annealing with a greedy algorithm for bayesian network structure learning. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1157–1166. [CrossRef]
45. Yuan, S.J.; Xu, Y.J.; Mu, B.; Zhang, L.L.; Ren, J.H.; Ma, S.Y.; Duan, W.S. An improved continuous tabu search algorithm with adaptive neighborhood radius and increasing search iteration times strategies. *Int. J. Artif. Intell. Tools* **2021**, *30*, 2150001. [CrossRef]
46. Veksler, B.Z.; Morris, M.B.; Krusmark, M.A.; Gunzelmann, G. Integrated modeling of fatigue impacts on c-17 approach and landing performance. *Int. J. Aerosp. Psychol.* **2023**, *33*, 61–78.
47. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
48. Ding, S.; Zhao, H.; Zhang, Y.; Xu, X.; Nie, R. Extreme learning machine: Algorithm, theory and applications. *Artif. Intell. Rev.* **2015**, *44*, 103–115. [CrossRef]
49. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
50. Ding, S.; Su, C.; Yu, J. An optimizing bp neural network algorithm based on genetic algorithm. *Artif. Intell. Rev.* **2011**, *36*, 153–162. [CrossRef]
51. Zhang, X.; Jiang, Y.; Zhong, W. Prediction research on irregularly cavitied components volume based on gray correlation and pso-svm. *Appl. Sci.* **2023**, *13*, 1354. [CrossRef]

52. Chen, Y.; Shi, G.; Jiang, H.; Zheng, T. Research on the prediction of insertion resistance of wheel loader based on pso-lstm. *Appl. Sci.* **2023**, *13*, 1372. [CrossRef]
53. Wu, T.; Wang, M.; Xi, Y.; Zhao, Z. Malicious url detection model based on bidirectional gated recurrent unit and attention mechanism. *Appl. Sci.* **2022**, *12*, 12367. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.