

Prediction of Array Antenna Assembly Accuracy Based on Auto-Encoder and Boosting-OSKELM

Miao Wang

Nanjing University of Science and Technology <https://orcid.org/0000-0001-7622-0497>

Yifei Tong (✉ tyf51129@aliyun.com)

Nanjing University of Science and Technology

Tong Zhou

Nanjing University of Science and Technology

Research Article

Keywords: Array antenna, Assembly accuracy, Auto-encoder, Boosting-OSKELM

Posted Date: March 21st, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1400540/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Title page

Prediction of Array Antenna Assembly Accuracy Based on Auto-Encoder and Boosting-OSKELM

Miao Wang, born in 1999, is currently a master candidate at *School of Mechanical Engineering, Nanjing University of Science & Technology, China*.

Tel: 17327990057; E-mail: 1063168093@qq.com

Yi-Fei Tong, born in 1981, received the PhD degree in Mechanical Engineering Science and now is a professor at *Nanjing University of Science & Technology, China*. His main research interests include advanced manufacturing system management & control and IoT.

E-mail: tyf51129@aliyun.com

Tong Zhou, born in 1997, is currently a master candidate at *School of Mechanical Engineering, Nanjing University of Science & Technology, China*.

E-mail: 1159187336@qq.com

Corresponding author: Yi-Fei Tong E-mail: tyf51129@aliyun.com

ORIGINAL ARTICLE

Prediction of Array Antenna Assembly Accuracy Based on Auto-Encoder and Boosting-OSKELM

Miao Wang¹ • Yi-Fei Tong¹ • Tong Zhou¹

Abstract: As a complex mechanical and electrical product, array antenna involves many parts and requires high precision, so its assembly process is complex, changeable and difficult to control. As a result, the assembly accuracy often fails to meet the design requirements and needs to be reassembled repeatedly, resulting in the delay of delivery date. In order to realize the accurate and efficient prediction of array antenna assembly accuracy, a prediction method based on auto-encoder and online sequential kernel extreme learning machine with boosting (Boosting-OSKELM) is proposed in this paper. The method is mainly divided into two steps: Firstly, the auto-encoder with fine-tuning trick is used for training and representation reduction of the data. Then, the data is taken as the input of Boosting-OSKELM to complete the initial training of the model. When new sample data is generated, Boosting-OSKELM can realize the online correction of the model through rapid iteration. The test shows that this method has strong robustness in prediction accuracy and online learning ability.

Keywords: Array antenna • Assembly accuracy • Auto-encoder • Boosting-OSKELM

1 Introduction

Electronic information industry is an important driving force for today's economic and social development. It is a strategic, basic and leading pillar industry of the national economy, which plays an important role in promoting economic growth, industrial structure, changing development mode and maintaining national security. In recent years, high-precision integrated antenna (array antenna), as an indispensable part of early warning and detection systems, has become the core of the national major project "Space-Earth Integration Network". Taking the new exploration satellite as an example, its array antenna is composed of 500 subarrays and more than 1

✉ Yi-Fei Tong
tyf51129@aliyun.com

¹ School of Mechanical Engineering, Nanjing University of Science & Technology, Nanjing 210094, China

million parts, with more than 100 million assembly welding points. The size of the satellite is more than 50 meters and the price is more than 300 million. It needs to be in trouble-free service for more than 8 years in the harsh space environment, which features extremely high requirements for reliability.

The basic structure of the array antenna body is shown in figure 1. Due to the complexity of the structure, array antennas are mostly assembled manually or with the help of mechanical equipment. With the uncertainty of assembly activities, the assembly process is often unqualified or the performance of the mechanical and electrical cannot meet the design standard, which results in an average of more than 12 months of repeated adjustment, seriously reducing the assembly efficiency. Therefore, in order to shorten the assembly cycle of array antenna, it is urgent to introduce an effective method of assembly accuracy prediction.

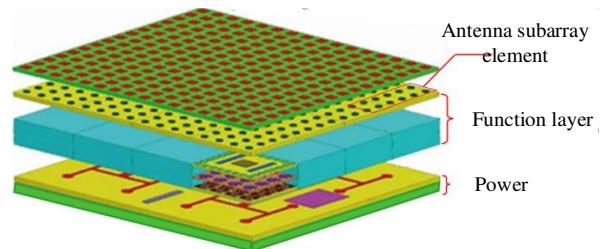


Figure 1 Basic structure of array antenna body

There have been many related researches in this field. In order to study the effect of assembly error on the antenna gain, Guo et al. [1] proposed an accurate gain prediction model using an improved XGBoost algorithm and the transfer learning method, based on the simulation data and experience. Combined with the principle of maximum entropy, Zhang et al. [2] proposed the probability density function model of the relationship between micro force and displacement, which can promote process accuracy prediction of micro-assembly. Mu et al. [3] studied the construction method of

composite processing components considering the manufacturing error and deformation factors of parts, and proposed a new prediction method of aero-engine high-pressure rotor system. In order to predict and control the variation in assembly process, Hong et al [4] developed a mathematical analysis method based on state space model, the validity of which is proved by a simulation.

The above research either established a data-driven model in regard of historical data and simulation data, or built a mechanism model based on physical principles. Both them are off-line prediction mode, which is difficult to achieve rapid prediction in some complex cases. Moreover, it also lacks the ability of model iteration, leading to a low efficiency in dealing with new sample data for online correction of the model.

In recent years, digital twin technology has been widely studied and applied in advanced manufacturing. As a virtual-physical fusion technology, it can realize the virtual-physical interaction, data fusion, decision analysis and iterative optimization of the whole assembly process by using the twin data of assembly context, based on virtual assembly information model and quantitative calculation of assembly quality [5]. Obviously, the implement of digital twin technology is inseparable from artificial intelligence technology, which is equipped with the ability of high-performance data analysis and real-time prediction. However, the assembly process of array antenna has the following features, which bring difficulties to assembly accuracy prediction:

(1) The assembly process of array antenna is complex and variable, and the data of its assembly samples are mostly in high-dimensional space, so single sample may have redundant or even contradictory information between sample features, which is a disadvantage to the prediction of assembly accuracy.

(2) Array antenna belongs to the small batch production mode, and its historical sample data has limitations, which will disturb the training of machine learning model.

As to the problem of high-dimension, data mining technology can be used to extract valuable information. The commonly used data mining method is representation reduction, which is a dimension reducing or feature extraction method. It can reduce the distance between sample points in high-dimensional space and preserve the valuable information of samples as much as possible, which is conducive to improving the inference speed and accuracy of prediction algorithm [6]. In traditional machine learning, the commonly used feature extraction methods are mainly divided into two parts, that is, linear

dimension reduction represented by PCA [7] and nonlinear dimension reduction based on manifold learning. PCA may bring more information loss in nonlinear problems, while manifold learning (such as LLE [8], t-SNE [9], etc.) starts from the relationship between samples, which leads to a high effectiveness of dimension reduction but also a high computational complexity. So the traditional method cannot calculate the dimension reduction at the second level.

For Few-shot learning problems in engineering projects, Kriging algorithm is often used. Although this method does not rely on the amount of data, it still depends on the distribution of data. If the distribution of data is poor, the prediction accuracy of Kriging may be greatly affected [10]. In addition, statistical learning methods such as K-nearest neighbor algorithm (KNN) [11] and Support Vector Regression (SVR) [12] can also adapt to Few-shot learning problems. But these algorithms often use off-line learning, and there is no good online incremental learning method for new sample data generated in the future.

In view of the issues above, the main contributions of the study in this paper are as follows:

(1) An improved auto-encoder is proposed to implement feature extraction, which not only has high computational efficiency, but also can adapt to small sample problems.

(2) Online sequential extreme learning machine with Boosting strategy(Boosting-OSKELM) is proposed to adapt to online learning. This method possesses fast speed of learning and model iteration, which can meet online learning requirements.

2 Methodology

2.1 Representation Reduction of Sample Data

2.1.1 Dimension Reduction Principle of Auto-encoder

In recent years, neural networks have made remarkable achievements in many fields, such as genetics [13], graph classification [14], medical diagnosis [15], fault diagnosis [16] and so on. Based on the general approximation theorem [17], it can fit any function in theory. Its general structure is mainly divided into three parts: input layer, hidden layer and output layer. The input layer and output layer correspond to the input data and prediction results respectively, and the middle hidden layer introduces a nonlinear activation function, which enables the neural network to learn more hidden feature information. Therefore, the neural network is good at multi-level representation and data prediction of nonlinear systems.

In the field of data dimension reduction, the common

model of neural network is auto-encoder (AE), which is a neural network aiming at restoring the input data to the greatest extent [18]. figure 2 is the structure of a single-layer auto-encoder (SAE), which is fully connected among layers. Assuming that the sample i is $\mathbf{x}(i) \in R^d$, the AE will map this sample to a new feature space as $\mathbf{z}(i) \in R^m$; Then, the AE will reconstruct the new sample representation into the original one, and the reconstructed sample is defined as $\mathbf{x}'(i) \in R^d$.

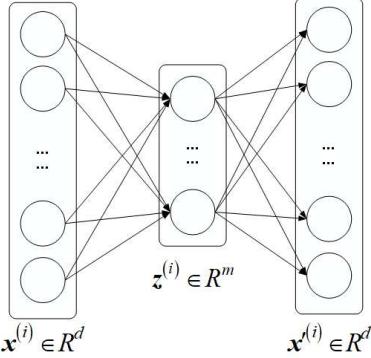


Figure 2 Single-layer Auto-encoder

As can be seen from the above figure, the AE can be divided into two parts: encoder $f : R^d \rightarrow R^m$ and decoder $g : R^m \rightarrow R^d$. The learning goal of the AE is to minimize the reconstruction loss shown in Eq.(1).

$$L = \sum_{i=1}^N \|x^{(i)} - g(f(x^{(i)}))\|^2 \quad (1)$$

where $\|\cdot\|$ is the vector-2 norm, and the mapping relationship f from $\mathbf{x}(i)$ to $\mathbf{z}(i)$ is represented in Eq.(2), where $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ respectively represent the connection weight and bias of the encoder part. Similarly, there is a mapping relationship g from $\mathbf{z}(i)$ to $\mathbf{x}'(i)$, which is represented by Eq.(3).

$$\mathbf{z}^{(i)} = f(\mathbf{W}^{(1)} \mathbf{x}^{(i)} + \mathbf{b}^{(1)}) \quad (2)$$

$$\mathbf{x}'^{(i)} = g(\mathbf{W}^{(2)} \mathbf{z}^{(i)} + \mathbf{b}^{(2)}) \quad (3)$$

If the dimension number of the feature space m is less than the dimension of the original space d , the AE can be regarded as a feature extraction method. The following is a sample representation reduction method. If necessary, in order to prevent over-fitting, the binding weight can be

considered, that is $\mathbf{W}^{(1)} = \mathbf{W}^{(2)}$. Then the regularization is used, as shown in Eq.(4), where $\|\cdot\|_F$ represents the F norm of the matrix, λ is the coefficient of regularization.

$$L = \sum_{i=1}^N \|x^{(i)} - g(f(x^{(i)}))\|^2 + \lambda \|W\|_F^2 \quad (4)$$

In some scenarios, the number of hidden layers can be increased to form a deep auto-encoder (DAE). Theoretically, deeper layers of neural network mean more neurons, and stronger learning performance. However, in practice, blindly stacking layers may lead to gradient disappearance or gradient explosion, resulting in non-convergence of neural training. Therefore, the construction of neural network requires skills and experience.

2.1.2 Fine-tuning Trick

In the early neural network training, if the random initialization strategy is adopted for the model parameters, the training of deep network will be difficult. In this regard, when training deep belief network, Hinton et al. [19] proposed a greedy pre-training method, to pre-train the restricted Boltzmann machine of each layer, and finally add the output layer. This method is called fine-tuning, which is a common and important deep learning training skill. At present, it has been widely used in many fields of artificial intelligence, especially in the field of transfer learning. The basic idea of fine-tuning is that, based on network pre-training, the network is modified through the traditional global learning algorithm to make the model converge to a better local optimum.

This paper uses the fine-tuning trick based on supervised learning. The basic process is shown in figure 3. Firstly, the decoder output of AE is combined with the input of multi-layer perceptron (MLP), namely the output of encoder is used as the input of MLP. Then the network structure is used to train the regression problem on the dataset, which is called pre-training. The reason for choosing MLP is that the network weight parameters of AE can be updated through gradient back propagation. After pre-training, the AE structure is separated and connection weights in the pre-training stage are retained. On this basis, the weight of AE is retrained, which is called tuning. After the fine-tuning is completed, the decoder part of AE is removed, while the encoder is retained, and the representation reduction model of the sample data is obtained.

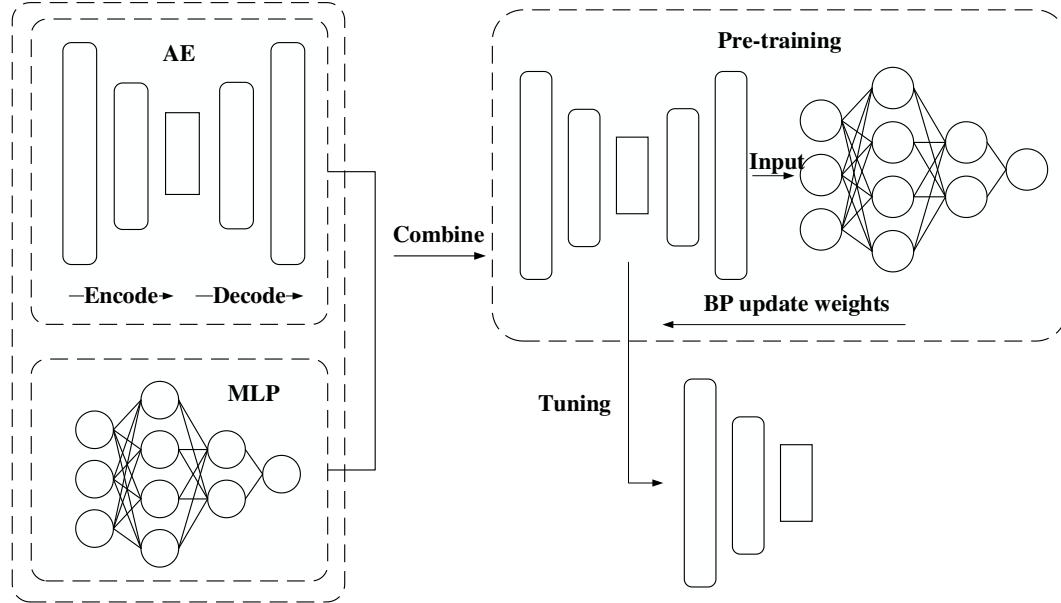


Figure 3 The basic process of fine-tuning

2.2 Online Learning Model

In theory, the network structure of encoder and MLP can be directly used to predict the assembly accuracy. However, due to the lack of available training samples, the generalization performance of the trained neural network may be poor. In addition, from the perspective of model online correction, although neural network can correct the model through gradient descent method, it heavily relies on batch data. If it is corrected by only a small number of samples, the prediction performance of the model may be sharply reduced due to the problem of "data poison" [20].

In order to solve the problem of model online learning, scholars have proposed a large number of online learning algorithms. Among them, online sequential extreme learning machine (OSELM) is favored by many scholars because of its fast training speed and good generalization performance, which can achieve online prediction of sample data efficiently [21]. It has the following advantages:

- (1) The learning speed of OSELM is very fast, which avoids the disadvantage of slow back-gradient update of traditional BP neural network.
- (2) It is easy for OSELM to obtain the global optimal solution. The optimization model, namely the least square method, is used to solve the network weight.
- (3) OSELM has few parameters, which avoids the great influence of learning rate parameters on the performance of BP neural network;
- (4) It can adapt to small sample data and has strong online learning ability.

OSELM derives from extreme learning machine (ELM) [22]. Based on this, the incremental learning formula of new samples is achieved. ELM is a feedforward neural network, and its basic structure is shown in figure 4.

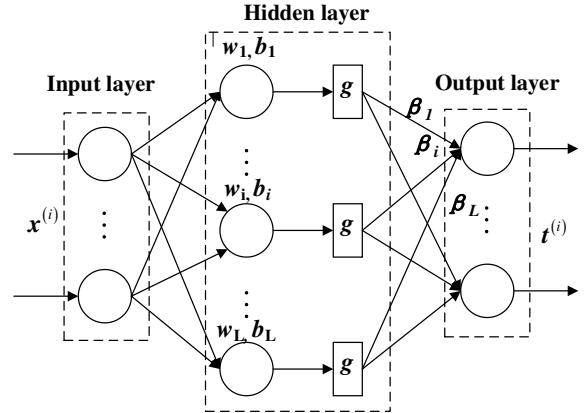


Figure 4 Single-layer ELM

Different from the current popular deep neural network, it does not update the network weight through back-gradient propagation, but solves the weight value through Moore-Penrose generalized inverse. Define dataset with n samples (X, T) , where $X = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}]^T$, $\mathbf{x}^{(n)} \in R^d$, and $T = [\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \dots, \mathbf{t}^{(N)}]^T$, $\mathbf{t}^{(n)} \in R^m$. If the number of neurons in the hidden layer is L and its activation function is g , the final output of ELM on the data set is shown in Eq. (5), $w_i \in R^d, b_i \in R$.

$$f_L(X) = \sum_{i=1}^L \beta_i g_i(\mathbf{w}_i X + b_i) \quad (5)$$

$$= \mathbf{Y}, n = 1, 2, \dots, N$$

Define $\mathbf{h}_i = \mathbf{g}_i(\mathbf{w}_i X + b_i)$ and convert it to matrix representation to obtain the output shown in Eq.(6).

$$f_L(X) = \mathbf{H}\boldsymbol{\beta} \quad (6)$$

where $\mathbf{H} = [\mathbf{h}_1(X), \mathbf{K}, \mathbf{h}_L(X)]_{N \times L}$, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \mathbf{K}, \boldsymbol{\beta}_L]^T$, $\boldsymbol{\beta}_l \in R^m$.

ELM training is mainly divided into two stages. The first stage is random mapping, where ELM randomly initializes the weight \mathbf{w}_i and bias b_i from the input layer to the hidden layer. The second stage is linear parameter solving. According to the weight and bias in the first stage, combined with the optimization problems shown in Eq.(6) and Eq.(7), then $\boldsymbol{\beta}$ can be solved.

$$\min \| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \|^2 \quad (7)$$

In order to enhance the stability of \mathbf{H} (nonsingular matrix), the regularization coefficient C and identity matrix \mathbf{I} are introduced. And because matrix \mathbf{H} is often row full rank, so the optimal value of $\boldsymbol{\beta}$ is shown in Eq.(8), where \mathbf{H}^+ represents the Moore-Penrose generalized inverse matrix of \mathbf{H} .

$$\boldsymbol{\beta}^* = \mathbf{H}^+ \mathbf{T} = \mathbf{H}^T \left(\mathbf{H} \mathbf{H}^T + \frac{\mathbf{I}}{C} \right)^{-1} \mathbf{T} \quad (8)$$

Although the learning speed of ELM is very fast, its prediction performance still lags behind the popular deep neural network. In order to enhance the nonlinear fitting ability of ELM, kernel function is introduced to form kernel extreme learning machine (KELM) [23].

Kernel function $K(x^{(i)}, x^{(j)}) = \mathbf{h}(x^{(i)}) \cdot \mathbf{h}(x^{(j)})$ is a common method to solve nonlinear problems. It maps the data in the original feature space to the new high-dimensional one. Learning is implicit in the new feature space, and there is no need to explicitly define the kernel mapping function in feature space. The kernel matrix $\mathbf{Q} = \mathbf{H} \mathbf{H}^T$ is defined according to Mercer condition [24], as shown in Eq.(9).

$$\mathbf{Q} = \mathbf{H} \mathbf{H}^T = \begin{bmatrix} K(x^{(1)}, x^{(1)}) & \dots & K(x^{(1)}, x^{(N)}) \\ \dots & \dots & \dots \\ K(x^{(N)}, x^{(1)}) & \dots & K(x^{(N)}, x^{(N)}) \end{bmatrix}_{N \times N} \quad (9)$$

Common kernel functions are as follows:

(1) Polynomial kernel function

$$K(x^{(i)}, x^{(j)}) = (a \cdot x^{(i)} \cdot x^{(j)} + b)^p \quad (10)$$

(2) Gaussian kernel function

$$K(x^{(i)}, x^{(j)}) = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}\right) \quad (11)$$

(3) Linear kernel function (i.e. no kernel)

$$K(x^{(i)}, x^{(j)}) = x^{(i)} \cdot x^{(j)} \quad (12)$$

where a, b, p, σ are constants.

It can be clearly seen that the introduction of kernel function makes ELM no longer affected by random weights \mathbf{w} and \mathbf{b} , and the prediction of new sample x can be calculated directly according to Eq.(13).

$$f(x) = h(x) \boldsymbol{\beta}^* = \left[K(x, x^{(1)}), \dots, K(x, x^{(N)}) \right] \left(\mathbf{H} \mathbf{H}^T + \frac{\mathbf{I}}{C} \right)^{-1} \mathbf{T} \quad (13)$$

When the kernel function is introduced into OSELM, an online sequential kernel extreme learning machine (OSKELM) is formed. For the dataset $\{(x^{(i)}, t^{(i)})\}_{i=1}^t$, which is set up to time t , the prediction result is $f(x) = \mathbf{h}(x) \boldsymbol{\beta}_t$, where $\boldsymbol{\beta}_t$ is $\boldsymbol{\beta}$ at time t .

Define $\mathbf{k}_t(x)$ and $\boldsymbol{\theta}_t$, as is shown in Eq.(14) and Eq.(15), then Eq.(16) can be derived.

$$\mathbf{k}_t(x) = \left[K(x, x^{(1)}), K(x, x^{(2)}), \dots, K(x, x^{(t)}) \right] \quad (14)$$

$$\boldsymbol{\theta}_t = \left(\mathbf{H}_t \mathbf{H}_t^T + \frac{\mathbf{I}}{C} \right)^{-1} \mathbf{T}_t \quad (15)$$

$$f(x) = \left[K(x, x^{(1)}), \dots, K(x, x^{(t)}) \right] \left(\mathbf{H}_t \mathbf{H}_t^T + \frac{\mathbf{I}}{C} \right)^{-1} \mathbf{T}_t = \mathbf{k}_t(x) \boldsymbol{\theta}_t \quad (16)$$

Moreover, define $\mathbf{A}_t = \mathbf{H}_t \mathbf{H}_t^T + C^{-1} \mathbf{I}$, then $\boldsymbol{\theta}_t = \mathbf{A}_t^{-1} \mathbf{T}_t$. For new samples $(x^{(t+1)}, t^{(t+1)})$ in time $t+1$, matrix \mathbf{A}_{t+1} as follows.

$$\mathbf{A}_{t+1} = \begin{bmatrix} \mathbf{A}_t & \mathbf{k}_t^T(x^{(t+1)}) \\ \mathbf{k}_t(x^{(t+1)}) & v_t \end{bmatrix} \quad (17)$$

where

$$v_t = C^{-1} + K(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t+1)}) \quad (19)$$

$$\mathbf{k}_t^T(\mathbf{x}^{(t+1)}) = [K(\mathbf{x}^{(t+1)}, \mathbf{x}^{(1)}), \dots, K(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)})]^T \quad (18)$$

The inverse matrix of the matrix \mathbf{A}_{t+1} is obtained according to the inverse formula of the block matrix, as shown in Eq. (20).

$$\mathbf{A}_{t+1}^{-1} = \begin{bmatrix} \mathbf{A}_t^{-1} + \mathbf{A}_t^{-1} \mathbf{k}_t^T(\mathbf{x}^{(t+1)}) \rho_t^{-1} \mathbf{k}_t^T(\mathbf{x}^{(t+1)}) \mathbf{A}_t^{-1} & -\mathbf{A}_t^{-1} \mathbf{k}_t^T(\mathbf{x}^{(t+1)}) \rho_t^{-1} \\ -\rho_t^{-1} \mathbf{k}_t^T(\mathbf{x}^{(t+1)}) \mathbf{A}_t^{-1} & \rho_t^{-1} \end{bmatrix} \quad (20)$$

where $\rho_t = v_t - \mathbf{k}_t^T(\mathbf{x}^{(t+1)}) \mathbf{A}_t^{-1} \mathbf{k}_t^T(\mathbf{x}^{(t+1)})$. Thus, the iterative formula of kernel function coefficient vector is obtained, as shown in Eq.(21). In this way, the new samples can be predicted according to Eq.(16).

$$\boldsymbol{\theta}_{t+1} = \mathbf{A}_{t+1}^{-1} \mathbf{T}_{t+1} \quad (21)$$

To sum up, the online training of OSKELM does not need to organize the old and new data together for retraining, but to absorb the new sample information by updating the matrix \mathbf{A}_{t+1} . After updating, the old dataset information will not be needed, which greatly reduces the computational complexity and improves the efficiency.

2.3 Boosting-OSKELM

The goal of supervised learning algorithm is to train stable models that perform well in all aspects. However, most of the time, the performance of supervised learning algorithm can only have decent performance in specific fields, which is also called weak learner. According to ensemble learning theory, weak learners and strong learners are actually equivalent, as several weak learners can obtain the same prediction performance as strong learners through a special

combination method. Among them, boosting is a common ensemble learning skill. Its basic idea is to correct the wrong prediction of other weak learners through weak learners.

Based on the idea of boosting, in order to squeeze the performance of the learner as much as possible, this paper ensembles multiple limit learners with different kernels, and proposes the Boosting-OSKELM algorithm, which is shown in figure 5. Firstly, polynomial kernel OSKELM is used on the original dataset for training. Then, calculate the residual between the fitting result and the real result on the training set, replacing the label of the original data with the residual, and then use Gaussian kernel OSKELM to learn and predict the new residual label. Finally, calculate the residual again, namely the ‘residual’ of the residual, and so on.

The pseudo code of Boosting-OSKELM is shown in Table 1. In practice, the training of the model does not have to follow the lifting order of Polynomial kernel → Gaussian kernel → Linear kernel, but can be adjusted appropriately, or even reuse a kernel, such as Gaussian kernel → Gaussian kernel → Gaussian kernel.

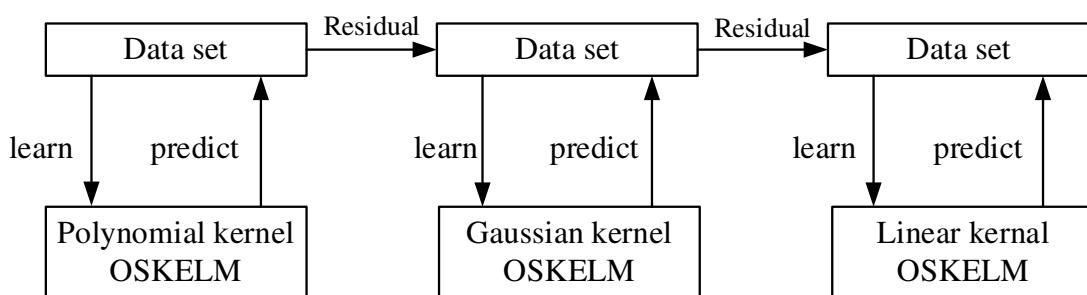


Figure 5 Improvement of OSKELM based on Boosting

Table 1 Pseudo code of Boosting-OSKELM

Algorithm: Boosting-OSKELM

Input: dataset: $D = \{\mathbf{x}, \mathbf{y}\}^N$, kernels = $\{k_1, k_2, k_3\}$;

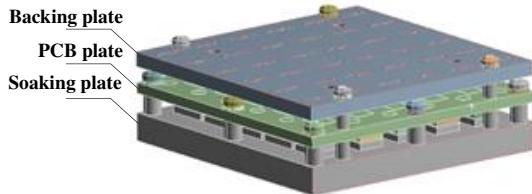
Output: Ensemble Learning E^* ;

1. Init $k = k_1$;
2. $\hat{\mathbf{y}} = OSKELM(k, \mathbf{x}, \mathbf{y})$
3. **For** $k = \{k_2, k_3\}$:
4. Calculate residual: $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$;
5. Fit residual: $\hat{\mathbf{r}} = OSKELM(k, \mathbf{x}, \mathbf{r})$;
6. Calculate: $\hat{\mathbf{y}} = \mathbf{y} + \hat{\mathbf{r}}$
7. **End**
8. get final Ensemble Learning E^* ;

3 Case Study

3.1 Data Description

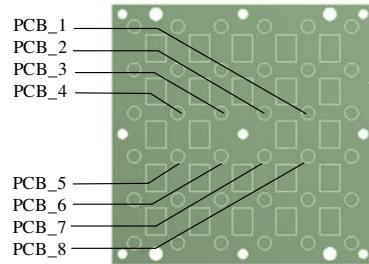
The simplified array antenna subarray unit is shown in figure 6. It is a stacking structure of three-layer flexible plates, including soaking plate, PCB plate and backing plate from bottom to top. The surface of the soaking plate has circular bosses with different heights, which are used to insert different panels. There will be slightly raised bosses on the surface of the PCB for welding the connector, and the other end of the connector is assembled with the insertion pins at the bottom of the backing plate. The connection between the panels is fixed by screws, while different screw preloads make different degrees of deformation between the panels, resulting in greater stress on the welding position of the connector, which seriously affects the assembly quality of the array antenna.

**Figure 6** Simplified solid model of subarray element

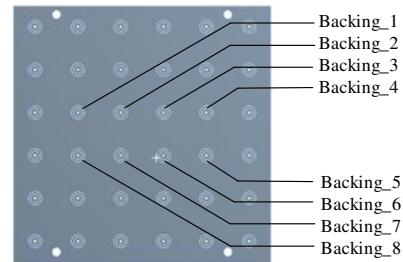
Because the backing plate will be arched to the middle under the action of preload, the middle area features large relative displacement, namely the main source of assembly error. In order to measure the assembly accuracy conveniently, this paper selects the relative displacement in X and Y direction of the position shown in figure 7 as the prediction target.

In this case, the input data of the model is the preload of

each screw (13 screws in total), and the output is the relative displacement of each position shown in figure 7. This paper uses numerical simulation in ANSYS simulation, and 40 corresponding data are obtained as shown in Table 2 and Table 3(the data has been normalized).



(a) positions in PCB plate



(b) positions in backing plate

Figure 7 Selected assembly accuracy prediction position**Table 2** Preload data of 40 screws.

	Screw 1	Screw 2	...	Screw 13
Sample 1	0.7237	0.8845	...	0.904
Sample 2	0.7784	0.4471	...	0.8941
...
Sample39	0.3464	0.7638	...	0.4953
Sample40	0.6692	0.7862	...	0.7256

Table 3 Displacement data of 40 screws.

	Δ_{1x}	Δ_{1y}	...	Δ_{8x}	Δ_{8y}
Sample1	0.6809	0.6776	...	0.6797	0.6812
Sample2	0.2165	0.2235	...	0.2179	0.217
...
Sample39	0.4779	0.4790	...	0.4766	0.4018
Sample40	0.1661	0.1617	...	0.1678	0.8455

3.2 Auto-encoder Training

The training of neural network is different from traditional machine learning. It involves a large number of hyper-parameters, including the network structure, learning rate and the selection of learners. If using traditional grid search, it will occupy a large amount of computing resources. Therefore, when training the AE, this paper gives empirical fixed values for some hyper-parameters.

(1) As to network architecture, based on the number of input features, the AE architecture is determined as 13-8-13.

(2) As to activation functions, the commonly used nonlinear activation functions are ReLU function, Sigmoid function and Tanh function. Compared with the other two functions, Tanh function has a relatively wide output range, which is more conducive to distinguishing the reduced representation between samples.

(3) As to optimizer, the adaptive moment estimation (Adam) proposed by Kingma et al. [25] is used, which retains the advantages of SGD (stochastic gradient descend) and introduces the momentum, so that the convergence speed of the neural network is accelerated and the learning rate can gradually decline with the number of iterations, which helps find a better local optimal solution.

(4) Due to the small sample size in the pre-training stage, regularization should be considered in order to prevent over-fitting. The regularization form shown in Eq.(4) is adopted, taking $\lambda = 20$.

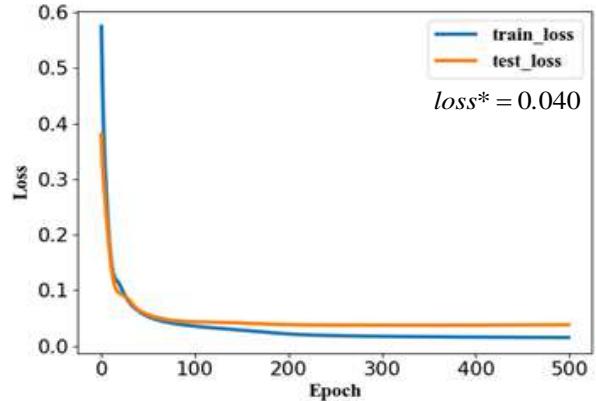
To sum up, the empirical values for parameters are shown in Table 4.

Table 4 Empirical values of AE parameters.

Name	Value
Activation function	Tanh
Learning machine	Adam
Learning rate	0.01
Regularization coefficient	20
Structure	13-8-13

The TensorFlow 2 AI framework based on Python is

used in this paper. Under the given parameters, the loss function changes with epoch, as shown in figure 8. It can be seen that after 500 epochs, the minimum loss on the test set is 0.040.

**Figure 8** Iteration of AE loss

In the stage of fine-tuning, the MLP network architecture is set to 13-16-32-16. As to the selection of activation function, the neural network regression problem generally chooses ReLU activation function, which can alleviate the problem of gradient disappearance or explosion in deeper neural networks. Other parameters of the network are consistent with AE.

In order to make the whole network develop towards optimizing the AE network, the loss function can be modified, which is shown in Eq.(22).

$$L = \eta_1 L_1 + \eta_2 L_2 \quad (22)$$

where L_1 represents the reconstruction loss of AE and L_2 represents the predicted loss of MLP. η_1 and η_2 are balance coefficients, used to balance the weight between reconstruction and prediction losses. The purpose of fine-tuning is to obtain a more refined low-dimensional representation of the sample, so the reconstruction loss should be fully considered. This paper takes $\eta_1 = 0.4$ and $\eta_2 = 0.6$.

Since the pre-training has adjusted the network weight to a reasonable range, the learning rate can be appropriately reduced at the training stage. In this paper, the initial learning rate is adjusted from 0.01 to 0.002, and the change of loss with epoch is shown in figure 9. Due to the role of pre-training, AE shows low loss at the beginning of training. With the number of iteration increases, the loss firstly decreases and then begins to rise, which means that the model shows over fitting. After tuning, the optimal loss of AE is 0.037, which is less than

the loss in figure 8 (0.004). It can be seen that fine-tuning is effective.

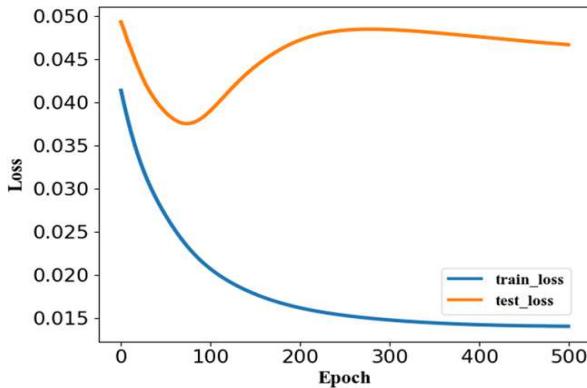


Figure 9 AE loss at tuning stage

Then the encoder of AE is extracted after training, with the weight retained. This part is used to reduce the representation of screw preloads and the results are shown in Table 5., where HD means hidden dimension.

Table 5 Representation reduction of screw preloads

	HD 1	HD 2	...	HD 8
Sample1	-0.6693	1.0457	...	-0.4189
Sample2	-0.6443	0.7543	...	-0.5363
...
Sample39	-0.4134	0.9636	...	-0.3634
Sample40	-0.7332	0.9879	...	-0.4876

3.3 Online Prediction of Assembly Error

After representation reduction of the dataset, the number of features is reduced from 13 to 8, and then the data is passed to KELM as input. For KELM, the kernel function needs to be determined first. Among three kernel functions, Polynomial kernel function and Gaussian kernel function need the corresponding constant coefficient values, and these values to consider are shown in Table 6.

Table 6 Optional hyper-parametric value of kernel function

Name	Value
a	1, 2, 3
b	1, 2, 3
p	1, 2, 3
σ	10, 50, 100

Although three kinds of kernel functions are used here, it is easy to cause the problem of combinatorial explosion when considering hyper-parameters of kernel function. In order to show the process and effectiveness of the method, this paper only tests 9 kinds of lifting order, and selects the optimal lifting order from them.

Based on the optional constant values provided in Table 5 and 9 lifting sequences, the training set was firstly trained and then the prediction performance was tested on test set. In order to further verify the effectiveness of representation reduction by AE, the experiment also compared the prediction result with the one without representation reduction. The final results are shown in Table 7, where L, G and P in the table represent Linear kernel, Gaussian kernel and Polynomial kernel respectively.

Table 7 Optimal results under 9 lifting sequences.

Lifting sequence	a	b	p	σ	MSE (Before reduction)	MSE (After reduction)
L-G-P	1	1	1	10	0.130062476	0.154568212
L-P-G	3	3	1	10	0.105822771	0.131452813
G-L-P	1	1	1	10	0.130104604	0.130115361
G-P-L	3	3	1	100	0.105821111	0.122455686
P-G-L	3	3	1	100	0.105821115	0.115254136
P-L-G	3	3	1	10	0.105823524	0.123654616
G-G-G	-	-	-	50	0.111207806	0.115623874
L-L-L	-	-	-	-	0.233036629	0.201251368
P-P-P	3	3	1	-	0.105893658	0.113456438

It can be seen from Table 7 that the performance of the nine lifting sequences is similar. Taking G-P-L as the lifting sequence can bring the smallest prediction error, and the corresponding optimal parameters are $a = 3$, $b = 3$, $p = 1$, $\sigma = 100$. In addition, by comparing the prediction results before and after representation reduction, it can be found that the prediction accuracy after reduction is slightly higher, which shows that the representation reduction of sample data is effective.

The above experimental data is only the prediction result of KELM under the boosting strategy. It is an offline data prediction to determine parameters of Boosting-OSKELM. In order to verify the online prediction performance of Boosting-OSKELM, a simple sequential addition principle is adopted, as shown in figure 10. The new sample points generated at time $t+1$ can be directly added to the training set, and the incremental learning is carried out according to Eq.(20). Finally, the error between the prediction and ground-truth will be obtained.

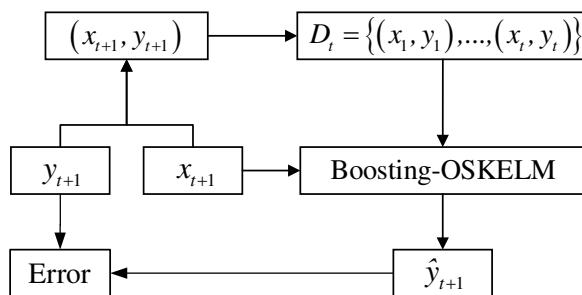


Figure 10 Sequential addition principle

For further verifying the online learning ability of Boosting-OSKELM, the artificial neural network (ANN) is selected for comparison. The results are shown in figure 11. According to this figure, with the increase of samples, the prediction error shows a downward trend. From the perspective of online prediction process, due to the randomness of sample distribution, there are some fluctuations in the online prediction process, but the fluctuations of Boosting-OSKELM is smaller than ANN, which shows that Boosting-OSKELM has stronger online learning adaptability.

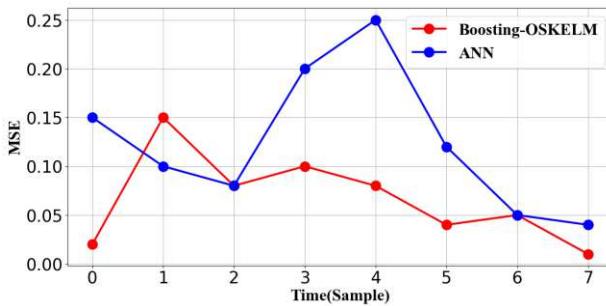


Figure 11 Online prediction performance over time

4 Conclusions

Due to the production mode, the sample data of array antenna presents the characteristics of high dimension and small samples, which brings difficulties to the prediction of its assembly accuracy. Therefore, this paper proposes a data representation reduction method based on AE with fine-tuning trick and an online prediction method based on Boosting-OSKELM.

(1) The representation reduction by AE can not only remove the redundant information in the original data, but also meet the real-time requirements in digital twin.

(2) With the help of multiple kernel functions and ensemble learning, Boosting-OSKELM can better adapt to online nonlinear learning problems. Compared with

traditional ANN, its generalization performance is relatively stable.

5 Declaration

Acknowledgements

This work is supported by the National Key Research and Development Program of China (grant No. 2020YFB1710300).

Funding

Not applicable.

Availability of data and materials

The raw datasets supporting the conclusions of this article are confidential.

Authors' contributions

The authors' contributions are as follows: Miao Wang wrote the manuscript and made data analysis; Yi-Fei Tong was in charge of the whole trial and sampling; Tong Zhou assisted with sampling.

Competing interests

The authors declare no competing financial interests.

Consent for publication

Not applicable

Ethics approval and consent to participate

Not applicable

References

- [1] F Guo, Z Liu, W Hu, et al. Gain prediction and compensation for subarray antenna with assembling errors based on improved XGBoost and transfer learning. *IET Microwaves, Antennas & Propagation*, 2020, 14(6): 551-558.
- [2] X F Zhang, Z J Zhang, Y Sun, et al. The Method Used for Process Precision Prediction of Micro-Assembly Based on Principle of Maximum Entropy. *Advanced Materials Research*, 2012, 591: 409-413.
- [3] X Mu, Y Wang, B Yuan, et al. A New assembly precision prediction method of aeroengine high-pressure rotor system considering manufacturing error and deformation of parts. *Journal of Manufacturing Systems*, 2021, 61: 112-124.
- [4] J Hong, J K Guo, Z G Liu, et al. Assembly accuracy prediction and adjustment process modeling of precision machine tool based on state space model. *Chinese Journal of Mechanical Engineering*, 2013, 49(06): 114-121. (in Chinese)
- [5] X Sun, J Bao, J Li, et al. A digital twin-driven approach for the assembly-commissioning of high precision products. *Robotics and Computer-Integrated Manufacturing*, 2020, 61: 101839.
- [6] B Vachharajani, D Pandya. Dimension reduction techniques:

- Current status and perspectives. *Materials Today: Proceedings*, 2022.
- [7] A J Bell, T J Sejnowski. The “independent components” of natural scenes are edge filters. *Vision Research*, 1997, 37(23): 3327-3338.
 - [8] S T Roweis, L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 290(5500): 2323-2326.
 - [9] D M L Van, G Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008, 9(11).
 - [10] Z H Han. Research progress of Kriging model and surrogate optimization algorithm. *Journal of Aeronautics*, 2016, 37(11):3197-3225. (in Chinese)
 - [11] B V Dasaray. Nearest Neighbor Pattern Classification Techniques. *Los Alamitos*. 1991.
 - [12] W S Noble. What is a support vector machine? *Nature Biotechnology*, 2006, 24(12): 1565-1567.
 - [13] M W Libbrecht, W S Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 2015, 16(6): 321-332.
 - [14] A Krizhevsky, I Sutskever, G E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012(25): 1097-1105.
 - [15] K Kourou, T P Exarchos, K P Exarchos, et al. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, 2015(13): 8-17.
 - [16] Y Li, Y X Huang, L J Zhao, et al. Tool wear evaluation under multiple working conditions based on t-distribution neighborhood embedding and XGBoost. *Chinese Journal of Mechanical Engineering*, 2020, 56(01):132-140. (in Chinese)
 - [17] K Hornik, M Stinchcombe, H White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, 2(5): 359-366.
 - [18] D E Rumelhart, G E Hinton, R J Williams. Learning internal representations by error propagation. *California Univ San Diego La Jolla Inst for Cognitive Science*, 1985.
 - [19] G E Hinton, S Osindero, Y W Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006, 18(7): 1527-1554.
 - [20] X Chen, C Liu, B Li, et al. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv Preprint ArXiv:1712.05526*, 2017.
 - [21] W Guo, J J Yu, K M Tang, T Xu. A Survey of Online Excessive Learning Algorithms for Dynamic Data Flow Analysis. *Computer Science*, 2019, 46(04):1-7. (in Chinese)
 - [22] G B Huang, Q Y Zhu, C K Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 2006, 70(1-3): 489-501.
 - [23] J L Dai, M H Wu, X Liu, R F Li. A Survey of Online State Prediction Methods for Kernel Extreme Learning Machines. *Journal of Ordnance Equipment Engineering*, 2021, 42(06):12-19. (in Chinese)
 - [24] J Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trinsdiction Royal Soc*, 1909(209): 4-415.
 - [25] D P Kingma, J Ba. Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*, 2014.

Tel: 17327990057; E-mail: 1063168093@qq.com

Yi-Fei Tong, born in 1981, received the PhD degree in Mechanical Engineering Science and now is a professor at *Nanjing University of Science & Technology, China*. His main research interests include advanced manufacturing system management & control and IoT.
E-mail: tyf51129@aliyun.com

Tong Zhou, born in 1997, is currently a master candidate at *School of Mechanical Engineering, Nanjing University of Science & Technology, China*.
E-mail: 1159187336@qq.com

Biographical notes

Miao Wang, born in 1999, is currently a master candidate at *School of Mechanical Engineering, Nanjing University of Science & Technology, China*.