Article

Fast Prediction Method of Combustion Chamber Parameters Based on Artificial Neural Network

Chenhuzhe Shao 1,2, Yue Liu 1,2,*, Zhedian Zhang 1,2, Fulin Lei 1,2 and Jinglun Fu 1,3

1 University of Chinese Academy of Sciences, Beijing 100190, China; shaochenhuzhe@iet.cn (C.S.)
2 Key Laboratory of Advanced Energy and Power, Institute of Engineering Thermophysics, Chinese Academy of Sciences, Beijing 100045, China
3 Advanced Gas Turbine Laboratory, Institute of Engineering Thermophysics, Chinese Academy of Sciences, Beijing 100045, China
* Correspondence: liuyue16@iet.cn

Abstract: Gas turbines are widely used in industry, and the combustion chamber, compressor, and turbine are known as their three important components. In the design process of the combustion chamber, computational fluid dynamics simulation takes up a lot of time. In order to accelerate the design speed of the combustion chamber, this article proposes a combustion chamber design method that combines an artificial neural network (ANN) and computational fluid dynamics (CFD). CFD results are used as raw data to establish a fast prediction model using ANN and eXtreme Gradient Boosting (XGBoost). The results show that the mean squared error (MSE) of the ANN is 0.0019, and the MSE of XGBoost is 0.0021, so the ANN’s prediction performance is slightly better. This fast prediction method combines CFD and the ANN, which can greatly shorten CFD calculation time, improve the efficiency of gas turbine combustion chamber design, and provide the possibility of achieving digital twins of gas turbine combustion chambers.

Keywords: gas turbine; combustion chamber; artificial neural network

1. Introduction

The gas turbine is a complex multidisciplinary coupling device that transforms the energy of fuel into useful work by using a continuous flow of gas as the working mass to drive the impeller to rotate at high speed. It is known as the jewel in the crown of the equipment manufacturing industry. Gas turbines are used in a wide range of applications, including ship power, aviation power, energy generation, and many other fields. Due to its low pollution, high efficiency, and reliability, the advanced gas turbine is the most competitive way to provide clean, environmentally friendly, high-quality, and efficient power generation and combined heat and power generation, and it has become the core of industrial technology in the 21st century. Since the world’s first gas turbine was built in Switzerland in 1939, there have been more than 21,000 gas turbines for power generation of more than 1 MW worldwide, with a total capacity of more than 1000 GW, and gas turbines account for 1/5 of the world’s total power generation [1].

The combustion chamber is the heart of the gas turbine, which needs to maintain a stable working state for a long time under the harsh conditions of high temperature and high pressure. However, the combustion chamber of an aerogine faces the problem of rapidly variable and dynamic characteristics [2]. Furthermore, the development process of the combustion chamber has the characteristics of great difficulty, a long cycle, and high cost. Plus, in order to meet the increasing requirements and technical indicators of the engine, the complexity of the engine system has increased significantly, and the combustion chamber design is also facing the challenge of leapfrog development [3].

The combustion chamber design system mainly consists of one- or two-dimensional thermal design and three-dimensional combustion simulation. Combustion is a complex
combination of chemical reaction and turbulent flow, and the combustion chamber needs to rely on a large number of tests for continuous iterative design. The main goal of gas turbine combustion chamber design is to achieve low emission, as well as transient and highly stable combustion under high-temperature and pressure conditions, while having the ability to burn multiple fuels, including low-calorific-value fuels such as blast furnace gas with hydrogen doping, especially pure hydrogen fuel with zero carbon emissions. Computational fluid dynamics (CFD) is an important simulation design tool, the difficulty of which lie in thermoacoustic oscillations and fluid-thermal coupling. Three-dimensional design simulation has been more widely used, and full four-dimensional non-constant design simulation is in the development and research stage. As numerical simulation tests use a number of assumptions and fixed boundary conditions, which cannot map all the characteristics of the physical object, a wise and prudent use of numerical simulation tests, combined with physical test verification, can effectively avoid the risk of misdirection brought about by numerical tests. At the same time, the increasing number of meshes in CFD simulation calculations makes it difficult to reduce the combustion chamber design cycle further. Intelligent algorithms such as machine learning can instead be used to mine big simulation data, not only to share some of the computational tasks to speed up the simulation calculation process but also to perform well in predicting non-constant phenomena in flow and heat transfer [4].

Machine learning methods can be divided into two aspects: traditional machine learning and deep learning. In the early stages of applying machine learning to scientific discoveries, scientists generally used traditional machine learning methods due to limitations in data volume and computational power. Based on the classic model, appropriate adjustments are made to the input, hyperparameters, structure, etc., of the model according to specific problems, in order to achieve the best results in classification and regression tasks [5,6]. Naive Bayes, logistic regression, K-nearest neighbors, and decision trees are relatively simple methods that have the characteristics of being intuitive and easy to implement [7–9].

Deep learning methods directly mine raw data and are adept at exploring the hidden structures and correlations of high-dimensional data. They can learn complex features and patterns that cannot be clearly extracted at the moment, break through the limitations of manual feature extraction, avoid deviations, and reduce the time required to calculate features. Compared to traditional machine learning methods, they have achieved better results. However, due to the complex structure and lack of interpretability of deep learning models, their application is to some extent limited [10,11]. Therefore, deep learning methods are suitable for scenarios with sufficient computing and data resources, complex problems, and poor performance in manually extracting features. Among them, common convolutional neural network structures include LeNet [12], AlexNet [13], ResNet [14], etc.; common recurrent neural network structures include LSTM [15] and GRU [16]; and common deep generation models include the variational autoencoder (VAE) [17] and generative adversarial network (GAN) [18].

The use of a neural network approach to build a fast prediction model for combustion chamber parameters can shorten the combustion chamber design cycle by omitting complex physical relationship calculations as it is fully data-driven. In addition, with sufficient data, it is possible to predict the state of a gas turbine combustion chamber entity using a fast prediction model of the combustion chamber. Therefore, the application of neural network methods to the field of combustion chamber design is very practical and has great potential.

Zhao Gang [19] of Zhengzhou Gas Power Generation Co., Ltd. used a single-hidden-layer artificial neural network to build a NOx emission prediction model and trained the model using gas turbine plant operating data to achieve prediction of four operating parameters, namely unit power, NOx concentration, combustion chamber vibration acceleration, and combustion chamber pressure pulsation. Warren G. Lamont [20] performed artificial neural network (ANN) modeling of a gas turbine combustion unit to predict gas turbine exhaust gas emissions and combustion chamber outlet temperature. Qian Wang [21] realized
the regression model of unsteady flow through proper orthogonal decomposition (POD) and a feedforward neural network (NN). Gianmarco Aversano [22] combined principal component analysis (PCA) with Kriging’s method to determine accurate low-order models. In this approach, PCA is used to identify and separate the invariants of the system, and then the Kriging method is used to find a response surface for these coefficients, thus generating a proxy model that allows parametric operations to be performed at a lower computational cost. Woojin Lee [23] used the proper orthogonal decomposition (POD) as well as the Kriging method and radial-basis-function neural network (RBFN) to perform a regression model, in which 500 MWe tangentially fired pulverized coal boilers were regressed and a reduced order model (ROM) was constructed. Wu Yifan et al. [24] combined the elementary cascade method and random forest method to achieve the prediction of compressor outlet temperature and pressure. Part of the feature data was collected using the primitive cascade method, and historical data were inputted into a model based on the random forest method to train and output the compressor outlet temperature and pressure. The average relative errors of the total pressure ratio and variable efficiency calculated by the model were −0.13% and 0.04%, respectively. The maximum relative errors were 2.11% and 1.90%, respectively. Gu X [25] and others used performance deviation models and extreme value function theory to monitor turbine exhaust temperature, mainly including backpropagation neural network (BPNN) models and performance deviation models. By training the BPNN model to output the performance deviation between the predicted and measured values of the turbine exhaust temperature, the normal or abnormal state can be determined. This method has an accuracy of up to 99.85%. The RBF neural network prediction model for combustion chamber emissions established by Sun Jihao [26] can accurately and quickly predict combustion chamber outlet emissions under different structural parameters. The maximum error in predicting NOx is 12.28%, and the average error is 4.58%. The maximum error in predicting CO is 2.75%, and the average error is 0.97%. Fentaye A D [27] combined the artificial neural network (ANN) and SVM to achieve multi-component fouling and erosion gas path fault diagnosis for compressors, power turbines, and gas generators. After inputting data into ANN for sample classification, the SVM model is used for diagnosis. Compared with multilayer perception (MLP), the classification accuracy of this method is more than 10% higher, and the classification accuracy for three component faults is 99.4%. Montazeri Gh M et al. [28] combined a growth neural network (GNN) and residual compensation limit learning machine (RCELM) to learn the fault characteristic map (FCM) of components to diagnose fouling and erosion faults in compressors, gas generators, and power turbines. The user inputs known measurement parameters into an extreme learning machine (ELM) to train and output health status bias. After modeling and compensating for the residual, the robust signal is input into the GNN network for detection, isolation, and recognition. The accuracies of fault detection and isolation are 99.97% and 97.74%, respectively.

At present, most scholars apply machine learning methods to two aspects of the gas turbine field, namely gas turbine outlet parameter prediction and gas turbine fault detection. There are relatively few cases where machine learning methods are applied to predict the internal flow field of gas turbines. Therefore, based on existing research, this article studies the application of ANN in predicting the internal flow field of gas turbine combustion chambers, and it proposes a rapid design method combining ANN and CFD for gas turbine combustion chambers.

2. ANN and XGBoost

2.1. ANN

Figure 1 shows a schematic diagram of a single-hidden-layer ANN.
Figure 1. Single-hidden-layer artificial neural network.

The input layer has n neurons from $x_1$ to $x_n$, representing each of the n input variables in the dataset. The output layer has m neurons from $y_1$ to $y_m$, representing each of the m output variables in the dataset. The hidden layer has p neurons from $a_1$ to $a_p$, and passes the data into the output layer after activation by the activation function $tanh(x)$.

\[
tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}
\]  

(1)

From the input layer to the hidden layer, the $i$-th neuron input to the hidden layer is $a_i$:

\[
a_i = \sum_{k=0}^{n} w_{ik} x_k
\]  

(2)

In the above equation, $w_{ik}$ is the weight of the $k$-th neuron in the input layer to the $i$-th neuron in the hidden layer.

The output of the $i$-th neuron of the hidden layer is $z_i$:

\[
z_i = g(a_i)
\]  

(3)

From the hidden layer to the output layer, the $j$-th neuron input to the output layer is $b_j$:

\[
b_j = \sum_{i=0}^{p} w_{ji} z_i
\]  

(4)

In the above equation, $w_{ji}$ is the weight of the $i$-th neuron in the hidden layer to the $j$-th neuron in the output layer.

The output layer activation function is the Leaky ReLU function $f(x)$:

\[
f(x) = \begin{cases} 
\lambda x, & x < 0 \\
 x, & x \geq 0 
\end{cases}
\]  

(5)

The output of the $j$-th neuron of the output layer is $y_j$:

\[
y_j = r(b_j)
\]  

(6)

According to the above algorithm, the neural network propagates forward from the input layer to the output layer, and the value of the output layer can be calculated from the original input data of the dataset and the weights between neurons. The value calculated...
by the output layer is compared with the value in the dataset, and the neural network node connection weights are continuously updated by using the error backpropagation method (BP algorithm), so as to continuously reduce the error between the model output and the dataset output and optimize the fitting effect. The final output model includes the structure of the neural network, the selection of the neuron activation function, and the connection weight between neurons.

2.2. XGBoost

XGBoost [29] is an efficient gradient boosting decision tree algorithm. It has been improved on the basis of the original GBDT (gradient boosting decision tree), greatly raising the model performance. As a forward addition model, its core is to adopt the integration idea of boosting, which integrates multiple weak learners into a strong learner through certain methods. Multiple trees are used to make joint decisions, and the results of each tree are the difference between the target value and the predicted results of all previous trees. By accumulating all the results, the final result is obtained, thereby improving the overall model performance.

XGBoost is composed of multiple CARTs (classification and regression trees), so it can handle problems such as classification and regression.

The prediction model for XGBoost is

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

(7)

In Equation (7), $K$ represents the number of decision trees, $f_k$ represents the $k$-th decision tree, and $\hat{y}_i$ represents the prediction result of $x_i$.

Initialization (there is no decision tree in the model, and its prediction result is 0):

$$\hat{y}_i^{(0)} = 0$$

(8)

Add the first decision tree to the model:

$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$$

(9)

Add the second decision tree to the model:

$$\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)$$

(10)

Add the $t$-th decision tree to the model:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$$

(11)

The loss function of XGBoost consists of two parts, namely the training error represented by $l$ and the regularization item represented by $\Omega$:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

(12)

Every time a decision tree is added to the model, the loss function changes. When the $t$-th decision tree is added, the previous $t-1$ decision trees have completed training, and the training error and regularization item of the previous $t-1$ decision trees have become constants. Therefore, the loss function can also be written as

$$Obj = \sum_{i=1}^{n} \left( y_i, \hat{y}_i^{(t-1)} + f_t(x_i) \right) + \Omega(f_t) + C$$

(13)
The regularization item can be rewritten as

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2$$  (14)

In the above equation, $T$ is the number of leaf nodes, and $w$ is the score of the leaf nodes. Substituting Equation (14) into Equation (13) yields

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 + C$$  (15)

Perform second-order Taylor expansion on Equation (15), and make the following assumptions:

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$  (16)

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$  (17)

$$G_j = \sum_{i \in I_j} g_i$$  (18)

$$H_j = \sum_{i \in I_j} h_i$$  (19)

The $I_j$ in Equations (18) and (19) represents the sample on the $j$-th leaf node. The loss function can be written as

$$Obj = \sum_{j=1}^{T} \left[ G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T$$  (20)

As shown in Equation (20), the loss function can be seen as a quadratic function about $w_j$, where there exists a $w_j$ that minimizes the loss function:

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$  (21)

3. Prediction of Combustion Chamber Parameters
3.1. CFD Calculation

The data used in this paper are from CFD simulations of a single-tube micro-hybrid combustion chamber. The geometry of the single-tube combustion chamber is well defined and the internal flow state is relatively simple, making it suitable for prediction of its internal parameters by artificial neural networks. The development of micro-mix combustion, a combustion technology proposed in recent years for hydrogen-rich fuel combustion, has resulted in smaller gas turbines and more efficient mixing. The principle is to reduce the mixing time by reducing the mixing scale through several simplified micro-nozzles instead of the traditional large-diameter nozzle structure [30].

The structure of the combustion chamber is shown in Figure 2. The main body length of the combustion chamber is 300 mm, the cross-section of the chamber is square, and the side length is 40 mm. The combustion chamber consists of four cylindrical nozzles with a diameter of 12 mm and a length of 64 mm.
The combustion chamber grid is shown in Figure 3.

Using Ansys Fluent 2022R1 for CFD calculation and Reynolds averaged Navier–Stokes (RANS) simulation for combustion chamber flow field, the control equation system cannot be closed due to the introduction of additional stress. Therefore, the Realizable $k - \varepsilon$ model widely used in the field of micro mixed combustion is used. The combustion model adopts Flamelet Generated Manifold (FGM). The continuity equation, momentum equation, and energy equation in the RANS simulation control equation are as follows:

$$\frac{\partial \rho \bar{u}_i}{\partial t} + \frac{\partial \rho u_i \bar{u}_j}{\partial x_j} = - \frac{\partial \rho}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \frac{\partial \rho u_i \bar{u}_j}{\partial x_j} \right)$$  \hspace{1cm} (22)

$$\frac{\partial \rho \bar{u}_i}{\partial t} + \frac{\partial \rho \bar{u}_i \pi_j}{\partial x_j} = - \frac{\partial \rho}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \frac{\partial (-\rho u'_i u'_j)}{\partial x_j} \right)$$  \hspace{1cm} (23)

$$\frac{\partial}{\partial t} \left( \rho \bar{E} \right) + \frac{\partial}{\partial x_j} \left( u_j (\rho \bar{E} + \bar{p}) \right) = \frac{\partial}{\partial x_j} \left( k \frac{\partial \bar{T}}{\partial x_j} \rho_c \bar{u}_i \bar{u}_j \right) + S$$  \hspace{1cm} (24)

In the above equations, $\pi_i$ represents Reynolds average velocity, $\sigma_{ij}$ represents stress tensor, $-\rho u'_i u'_j$ represents Reynolds stress, $k$ represents turbulent kinetic energy, and $u'_i$ represents pulsating velocity.

The mole fractions of fuel and oxidant, as well as the boundary conditions for CFD solution, are shown in Tables 1 and 2:

**Table 1.** Mole fraction of fuel and oxidant.

<table>
<thead>
<tr>
<th>Species</th>
<th>Fuel</th>
<th>Oxid</th>
</tr>
</thead>
<tbody>
<tr>
<td>O$_2$</td>
<td>0.21</td>
<td>0</td>
</tr>
<tr>
<td>CH$_4$</td>
<td>0.9623</td>
<td>0</td>
</tr>
<tr>
<td>CO$_2$</td>
<td>0.0047</td>
<td>0</td>
</tr>
<tr>
<td>C$_2$H$_6$</td>
<td>0.0233</td>
<td>0</td>
</tr>
<tr>
<td>N$_2$</td>
<td>0.0097</td>
<td>0.79</td>
</tr>
</tbody>
</table>
Table 2. Boundary conditions in Ansys Fluent 2022R1.

<table>
<thead>
<tr>
<th>Boundary Condition</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet Mass Flow Rate [kg/s]</td>
<td>0.07286</td>
</tr>
<tr>
<td>Inlet Turbulent Intensity [%]</td>
<td>20</td>
</tr>
<tr>
<td>Inlet Hydraulic Diameter [m]</td>
<td>0.012</td>
</tr>
<tr>
<td>Outlet Backflow Turbulent Intensity [%]</td>
<td>5</td>
</tr>
<tr>
<td>Outlet Backflow Turbulent Viscosity Ratio</td>
<td>10</td>
</tr>
<tr>
<td>Wall Motion</td>
<td>Stationary Wall</td>
</tr>
<tr>
<td>Wall Roughness Model</td>
<td>No Slip</td>
</tr>
<tr>
<td>Ignition Temperature [K]</td>
<td>2000</td>
</tr>
</tbody>
</table>

3.2. Dataset Preprocessing

The raw dataset contains 42 physical parameters on 941,774 grid points in the combustion chamber, containing pressure, density, x-axis direction velocity, temperature, oxygen molar fraction, nitrogen molar fraction, and nitrogen oxide molar fraction. In order to simplify the calculation, the 3D coordinates of the 941,774 grid points of the combustion chamber are chosen as the input $X$ to the artificial neural network in this paper:

$$X = \begin{bmatrix}
  c_{x1}^{1} & \cdots & c_{x941,774}^{1} \\
  c_{x1}^{2} & \cdots & c_{x941,774}^{2} \\
  c_{x1}^{3} & \cdots & c_{x941,774}^{3}
\end{bmatrix}$$

(25)

In the above equation, $c_{x1}$ to $c_{x941,774}$ are the x-axis coordinates on all grid points; $c_{y1}$ to $c_{y941,774}$ are the y-axis coordinates on all grid points; $c_{z1}$ to $c_{z941,774}$ are the z-axis coordinates on all grid points.

The five parameters corresponding to pressure, density, velocity in the x-axis direction, temperature, and molar fraction of oxygen are chosen as the outputs of the artificial neural network $Y$:

$$Y = \begin{bmatrix}
  p^{1} & \cdots & p^{941,774} \\
  d^{1} & \cdots & d^{941,774} \\
  c_{x1}^{1} & \cdots & c_{x941,774}^{1} \\
  c_{y1}^{2} & \cdots & c_{y941,774}^{2} \\
  c_{z1}^{3} & \cdots & c_{z941,774}^{3}
\end{bmatrix}$$

(26)

In the above equation, $p^{1}$ to $p^{941,774}$ are the pressure values on all grid points; $d^{1}$ to $d^{941,774}$ are the density values on all grid points; $v_{x1}$ to $v_{x941,774}$ are the velocity values in the x-axis direction on all grid points; $t^{1}$ to $t^{941,774}$ are the temperature values on all grid points; and $o_{21}$ to $o_{2941,774}$ are the oxygen molar fractions on all grid points.

Data preprocessing consists of two main aspects: outlier processing and normalization.

The dataset is scanned for outliers using the box line diagram method. As shown in Figure 4, data that are not determined to be outliers are not processed and data that are determined to be outliers are replaced with an upper limit if it is greater than the upper limit of the box plot method and a lower limit if it is less than the lower limit of the box plot method. The first quartile (Q1), also known as the ‘lower quartile’ or ‘lower quartile’, is equal to the 25th percentile of all values in the sample in descending order. The second quartile (Q2), also known as the ‘median’, is equal to the 50th percentile of all values in the sample from the smallest to the largest. The third quartile (Q3), also known as the ‘greater quartile’ or ‘upper quartile’, is equal to the 75th percentile of all values in the sample from the smallest to the largest. Data less than the lower quartile $Q1 – 1.5 \times IQR$ or greater than the upper quartile $Q3 + 1.5 \times IQR$ are considered outliers and are replaced by the lower and upper quartiles, respectively, with the remaining normal values being left untreated.
The lower and upper quartiles, respectively, with the remaining normal values being left untreated.

Figure 4. Outlier preprocessing flow chart.

The input and output data have different scale units with large differences in order of magnitude, which is neither conducive to the convergence of the neural network nor to the evaluation of errors in the subsequent process. The dataset is linearly normalized by scaling the input data to $[-1, 1]$ and the output data to $[0, 1]$, as shown in Equations (27) and (28):

$$x' = \frac{x - \text{mean}(x)}{\text{max}(x) - \text{min}(x)}$$

$$y' = \frac{y - \text{min}(y)}{\text{max}(y) - \text{min}(y)}$$

The training set is used to update the weights to fit the network and to continuously improve the accuracy. The validation and test sets are used to check how well the model matches the data not involved in the training, i.e., to check the generalization ability of the model trained from the training set. Due to the large amount of raw data, the partition ratio of the training set, validation set, and testing set is 98:1:1.

3.3. Training of ANN

Firstly, the network structure of the neural network is determined, which mainly includes the number of hidden layers, the number of hidden-layer neurons, the training algorithm, and the activation function of each layer. Theoretically, the structure of multiple hidden layers has a better fitting effect on multivariate non-linear mapping relationships. The number of neurons in the hidden layer is similar to the number of layers in the hidden layer. Increasing the number of neurons can better fit the non-linear relationship, but at the same time increases the possibility of overfitting and time cost, so the number of nodes should be reduced as much as possible while considering the number of input and output features and ensuring the accuracy of the fit. The optimization algorithm is an algorithm that adjusts the connection weights of the neural network nodes according to the error, and the Adam algorithm is chosen. By adjusting its initial learning rate, it can ensure that the update step size is limited to an approximate range, and at the same time can achieve automatic adjustment of the learning rate, which is very suitable for the application scenarios of large-scale data in this paper. The tanh (1) is chosen for the hidden layer, which
has a strong learning capability and is a good fit for multivariate non-linear problems with output values in (0, 1). The Leaky ReLU (5) is chosen for the output layer, which can further accelerate the convergence speed.

The mean square error (MSE) was chosen as the loss function, and the ANN model with different numbers of hidden layers, different numbers of neurons, and different initial learning rates was trained 20 times, each time with 2000 epochs. The lowest value of the MSE was taken for comparison to select the artificial neural network model with the best prediction effect. The code for the ANN was written in Python3.9 and PyCharm Community Edition 2023.2.1.

3.4. Training of XGBoost

To simplify the model, only manual adjustments were made to the hyperparameters of learning rate and number of decision trees. The range of learning rate values was 0.01 to 0.2, and the adjustment range for the number of decision trees was 100 to 500. In theory, a small learning rate and a small number of decision trees can lead to poor model accuracy, while large ones carry a greater risk of overfitting. During the hyperparameter adjustment process, if the MSE of the test set no longer decreases as the learning rate and number of decision trees increase, it indicates that the optimal hyperparameter combination has been found.

4. Results Analysis

4.1. Number of ANN Hidden-Layer Neurons and Learning Rate

Theoretically, as the number of hidden layers and neurons increases, artificial neural networks become more capable of extracting the non-linear mapping relationship between input and output. However, in practice, too many layers of hidden layers or the number of neurons can also bring about overfitting problems. In order to obtain the optimal artificial neural network structure, this section explores the effect of different numbers of hidden layers and different numbers of neurons in each hidden layer on the error. As shown in Figure 5, the MSE of the validation set tends to decrease and then increase as the number of neurons increases when only a single hidden layer is used. When 30 neurons are included, the MSE is the smallest, at 0.0108.

![Figure 5. Relationship between number of neurons in single hidden layer and MSE.](image)

As shown in Figure 6, with the use of double hidden layers, the MSE of the validation set is significantly larger when the first hidden layer has 5 and 10 neurons. When the first hidden layer has 15, 20, 25, and 30 neurons, the MSE is relatively well distributed between 0.01 and 0.002. In particular, when the first hidden layer contains 20 neurons and the second hidden layer contains 10 neurons, the mean squared error is the smallest at 0.0019.
As shown in Figure 6, with the use of double hidden layers, the MSE of the validation set is significantly larger when the first hidden layer has 5 and 10 neurons. When the first hidden layer has 15, 20, 25, and 30 neurons, the MSE is relatively well distributed between 0.01 and 0.002. In particular, when the first hidden layer contains 20 neurons and the second hidden layer contains 10 neurons, the mean squared error is the smallest at 0.0019.

Figure 6. Relationship between the number of double-hidden-layer neurons and MSE.

As shown in Figure 7, 5*10 in the legend represents that the first hidden layer contains 5 neurons and the second hidden layer contains 10 neurons. In the case of using three hidden layers, when the numbers of neurons in the first hidden layer and the second hidden layer are both 5, and when the first hidden layer and the second hidden layer contain 5 and 10 neurons, respectively, the MSE of the validation set is greater than 0.025, which is significantly larger. When the first hidden layer contains 10 neurons, the second hidden layer contains 5 neurons, and the third hidden layer contains 15 neurons, the MSE is the smallest, reaching 0.0035.

Figure 7. Relationship between number of neurons in three hidden layers and MSE.
In summary, the minimum MSE in learning 2000 epochs is 0.0108 for the single-hidden-layer structure and 0.0019 for the double-hidden-layer structure. The minimum MSE for the triple-hidden-layer structure is 0.0035. Therefore, the optimal structure is the double-hidden-layer structure, where the first hidden layer contains 20 neurons and the second hidden layer contains 10 neurons.

The learning rate affects the speed of convergence of the error. The larger the learning rate, the faster the error decreases, but it may oscillate back and forth around the minimum or even fail to converge. In order to arrive at the most appropriate initial learning rate for the optimal ANN structure explored in the previous section, this paper goes on to compare the effect of different initial learning rates on the error.

As shown in Figure 8, when the initial learning rate is 0.05, 0.01, or 0.005, the mean squared error of the validation set drops to less than 0.01 at 500 epochs of training, and the convergence rate is relatively fast. When the initial learning rate is 0.01, the validation set MSE is the smallest, and the error value is 0.0019 when trained to 2000 epochs.

4.2. XGBoost

The hyperparameters of learning rate and number of decision trees have a significant impact on the error of XGBoost regression. The higher the learning rate, the higher the model accuracy but the greater the tendency for overfitting; the smaller the learning rate, the lower the overfitting tendency of the model, but the lower the accuracy. The more decision trees there are, the more complex the model becomes and the more prone it is to overfitting. As shown in Figure 9, as the learning rate and number of decision trees continue to increase, the MSE of XGBoost on the test set shows a decreasing trend. When the learning rate is 0.2, XGBoost regression has the best effect, but when the number of decision trees exceeds 200, the downward trend of MSE is no longer significant and ultimately reaches 0.0021.
4.3. Visualization

Predictions of the combustion chamber parameters were made using artificial neural networks based on the studies in the previous two sections. Fourteen groups of CFD raw data and predicted values in the test set were randomly intercepted and compared, as shown in Figures 10–14. Overall, the artificial neural network predictions for the five parameters of pressure, density, x-axis direction velocity, temperature, and oxygen molar fraction are relatively close to the CFD raw data, indicating that the prediction of the combustion chamber parameters using artificial neural networks is feasible.

Figure 9. MSE decline process with different initial learning rates and numbers of decision trees.

Figure 10. Comparison of raw pressure data and predicted pressure.
Figure 11. Comparison of raw density data and predicted value.

Figure 12. Comparison of raw data and predicted value of velocity in x-axis direction.

As shown in Figure 10, except for how the predicted value of the 5th group of pressure is slightly higher than the original data, the other 13 groups of data are very close. In Figure 11, the density prediction value of group 6 is slightly higher than the original data. In Figure 12, the predicted value of the 10th group of x-axis speed is slightly higher than the original data. As shown in Figure 13, the 14 groups of predicted temperature values are close to the original data. In Figure 14, the predicted oxygen mole fraction values of groups 5 and 6 are slightly less than those in the original data.

In order to explore the reasons for the deviation between the above individual predicted values and the original data, the training epoch of the neural network was increased from 2000 to 5000, but there was no change in the experimental results, indicating that the training epoch of the neural network was independent of the deviation. Further conjecture is that there are individual outliers in the original data, and the impact of this aspect needs to be further studied.
Figure 13. Comparison of raw temperature data and predicted value.

Figure 14. Comparison of raw data and predicted values of oxygen mole fraction.

For a more intuitive comparison of the prediction accuracies of artificial neural networks, as shown in Figures 15–19, we plotted the original data and predicted values of the test set into point cloud maps. From the results, it can be seen that there are outliers in the point clouds of the raw data of each parameter that differ significantly from the values of nearby grid points, while there are basically no outliers in the predicted point clouds. Overall, the CFD raw data are highly consistent with the ANN predicted values, indicating that outliers in the raw data have a small impact on the ANN predicted results, and the ANN predicted results are in line with expectations.
As shown in Figure 10, except for how the predicted value of the 5th group of pressure is slightly higher than the original data, the other 13 groups of data are very close. In Figure 11, the density prediction value of group 6 is slightly higher than the original data. In Figure 12, the predicted value of the 10th group of x-axis speed is slightly higher than the original data. As shown in Figure 13, the 14 groups of predicted temperature values are close to the original data. In Figure 14, the predicted oxygen mole fraction values of groups 5 and 6 are slightly less than those in the original data.

In order to explore the reasons for the deviation between the above individual predicted values and the original data, the training epoch of the neural network was increased from 2000 to 5000, but there was no change in the experimental results, indicating that the training epoch of the neural network was independent of the deviation. Further conjecture is that there are individual outliers in the original data, and the impact of this aspect needs to be further studied.

For a more intuitive comparison of the prediction accuracies of artificial neural networks, as shown in Figures 15–19, we plotted the original data and predicted values of the test set into point cloud maps. From the results, it can be seen that there are outliers in the point clouds of the raw data of each parameter that differ significantly from the values of nearby grid points, while there are basically no outliers in the predicted point clouds. Overall, the CFD raw data are highly consistent with the ANN predicted values, indicating that outliers in the raw data have a small impact on the ANN predicted results, and the ANN predicted results are in line with expectations.

![Figure 15](image15.png)  
**Figure 15.** Pressure point cloud: (a) original data; (b) predicted values.

![Figure 16](image16.png)  
**Figure 16.** Density point cloud: (a) original data; (b) predicted values.
level or even the entire machine level. The combustion chamber control system to achieve true digital twins at the component level can be connected in series with the fast prediction model. The research can be extended to the prediction of multiple operating conditions in the combustion chamber, and the fast prediction model can be connected in series with the training duration of the ANN is only 6 min, which is much shorter than the calculation in the predicted value are significantly less than the CFD calculation results. Moreover, MSE of the ANN predicted value is 0.0019, which is better than XGBoost, and the outliers effects of the number of hidden layers and neurons, as well as the number of decision trees, on two fast prediction models based on the ANN. This article investigates the effects of the number of hidden layers and neurons, as well as the number of decision trees, on two fast prediction models based on the ANN. Research has shown that compared with the CFD calculation results, the fast prediction method can greatly accelerate the calculation speed while ensuring accuracy.

5. Conclusions
This article applies the CFD calculation results to the training of the ANN and XGBoost for a single-tube micro mixed combustion chamber model. By inputting three-dimensional positional coordinates, we predict the distribution of five parameters in the combustion chamber: pressure, density, x-axis velocity, temperature, and oxygen mole fraction. This
The article investigates the effects of the number of hidden layers and neurons, as well as the number of decision trees, on two fast prediction models based on the ANN and XGBoost. Research has shown that compared with the CFD calculation results, the MSE of the ANN predicted value is 0.0019, which is better than XGBoost, and the outliers in the predicted value are significantly less than the CFD calculation results. Moreover, the training duration of the ANN is only 6 min, which is much shorter than the calculation duration of CFD, which often takes several hours. Therefore, the fast prediction method of combustion chamber parameters based on artificial neural networks can assist in CFD calculation during the combustion chamber design process, greatly accelerating the calculation speed while ensuring accuracy.

The current research is a preliminary exploration of digital twins in gas turbine components. The research can be extended to the prediction of multiple operating conditions in the combustion chamber, and the fast prediction model can be connected in series with the combustion chamber control system to achieve true digital twins at the component level or even the entire machine level.

Author Contributions: Conceptualization, C.S. and Y.L.; methodology, C.S., Y.L. and Z.Z.; software, C.S. and F.L.; validation, C.S.; formal analysis, C.S.; investigation, J.F.; resources, Y.L. and F.L.; data curation, C.S. and F.L.; writing—original draft preparation, C.S.; writing—review and editing, C.S. and Y.L.; visualization, C.S.; supervision, Y.L. and J.F.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study and the code are available on request from the corresponding author.

Acknowledgments: The authors wish to acknowledge the energy and power research center of the Institute of Engineering Thermophysics, Chinese Academy of Sciences for its assistance in the CFD calculation of combustion chamber data.

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

References
9. Sendek, A.; Yang, Q.; Cubuk, E.; Duerloo, K.-A.N.; Cui, Y.; Reed, E.J. Holistic computational structure screening of more than 12000 candidates for solid lithium-ion conductor materials. Energy Environ. Sci. 2017, 10, 306–320. [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.