A New Kind of Anti-bending and Anti-deformation Cathode Frame for Electrostatic Precipitators

Jingsong Zeng [1,a], Yuhan Zeng [2,b], Ziwei Wang [2,c]

[1] Department of Electrical Engineering,Anhui Technical College of Mechanical and Electrical Engineering,Wuhu, China.

[2] School of continuing education, Anhui Technical College of Mechanical and Electrical Engineering Wuhu, China.

[3]Nanjing KFC Anhui Branch, Wuhu, China.

[a] ahjdzjs@126.com, [b] zengyh2023@163.com, [c] 2387669489@qq.com

Abstract. The aeroengine boasts a complex structure and operates in a demanding environment characterized by prolonged exposure to extreme conditions such as high temperature and pressure. This type of system is susceptible to nonlinear faults. To enhance the accuracy of aeroengine predictions, this paper proposes a fault diagnosis model based on stacking-based ensemble learning. Initially, the KNN algorithm (K-Nearest Neighbor) and the AdaBoost algorithm (Adaptive Boosting) are optimized by hyperparameters to further augment the accuracy of a single model. Subsequently, the XGBoost model is employed as a meta-learner to fuse the prediction outcomes of the optimized KNN and AdaBoost models. The stacking ensemble learning technique is then applied, followed by the output of the prediction results. Through confirmatory experiments, the accuracy rate following stacking integrated learning improved to 0.9591, while the standard deviation was further minimized to 0.0074. The findings demonstrate that the model is remarkably precise and robust, and can be implemented in aeroengine fault diagnosis.

Keywords: Ensemble learning; Stacking; KNN; AdaBoost; XGBoost; Aircraft engine.

1. Introduction

In the 21st century, aeroengines will maintain their significant position in advancing human scientific and technological progress and social development[1]. However, owing to the intricate precision of aeroengines and their operating environment which is characterized by extreme conditions such as high temperature, high pressure and heavy load for extended periods, engine failure is inevitable, leading to considerable personnel and economic losses. Between 1949 and 1999, 29 significant flight safety accidents occurred, with aircraft failure accounting for a high proportion of 68.44% as the primary cause of these accidents[2]. Between 1995 and 2015, 69 general aviation accidents transpired, with engine failure being the most common unsafe state, accounting for 8 of these accidents[3]. As a result, aeroengine fault research has always been a major obstacle both domestically and internationally[4].

Currently, aeroengine fault research methods can be broadly classified into the following three categories: fault diagnosis based on the construction of aeroengine mathematical and physical models, fault diagnosis based on sensor signal processing, and fault diagnosis based on data drive[5]. Based on the method of constructing a mathematical model, Zhang Peng[6] and Lambert[7] explored the fault detection method that employs the Kalman filter directly to the nonlinear model of the engine. Borguet et al.[8] investigated the addition of model deviation to the original Kalman filter as an additional random measurement error, thus resulting in a more precise improved Kalman filter. Nevertheless, the aeroengine itself is an extremely complex and nonlinear model, and the method of constructing mathematical models presents a challenge where high accuracy of the model corresponds to high accuracy of diagnosis, and vice versa. Presently, diagnosing aeroengine faults using pure mathematical models proves to be difficult.

The signal processing-based method effectively overcomes the difficulty of modeling high-precision mathematical models. Fault diagnosis through signal processing can be categorized into single parameter and multi-parameter fusion diagnosis methods. In single parameter diagnosis, Wanyucheng et al.[9] utilized the metal elements' content parameter in the oil sample as a characteristic parameter to analyze the wear fault of aeroengine using extension set theory. Zhaohongli et al.[10] studied aeroengine performance degradation using the engine exhaust temperature margin as the characteristic parameter. However, the above single parameter diagnosis research exhibits a high false alarm rate and lacks high robustness. Thus, research on multi-parameter fusion was conducted. Hujinhai et al.[11] proposed installing multiple vibration sensors to form a network and applying the D-S evidence decision fusion method for final diagnosis. Han et al.[12] designed the gain scheduling of a PI controller based on the dual scheduling of speed and health parameters, comprehensively considering the engine fault problem using the dual parameters. Although the above multi-parameter research applies the method of parameter fusion, the aero-engine's complex mechanical coupling structure and time-varying speed and torque result in variable working conditions, increasing the difficulty of sensor acquisition and signal processing, which affects accuracy negatively.

The employment of data-driven fault diagnosis techniques has gradually gained prominence in contemporary research. In the domain of shallow neural networks, Xuqihua et al.[13] and Wei xunkai et al.[14] have directly implemented SVM for the purpose of fault diagnosis of aeroengine. Further advances in the application of support vector machine in aeroengine fault detection have been made by Wujunfeng et al.[15], Zhuyongxin et al.[16] and Islam et al.[17], leading to a marked improvement in the accuracy of prediction. Sun Xiaoyu[18], on the other hand, utilized three machine learning models for prediction and verified that the random forest algorithm possesses superior learning capabilities for higher dimensional data when compared to the support vector machine algorithm. Zedda et al.[19] have introduced sparse Bayesian algorithm in engine gas path fault diagnosis, achieving commendable results. However, a large proportion of the research findings rely on a solitary machine learning model, and consequently suffer from issues of low accuracy and limited robustness.

Given the aforementioned issues, this paper puts forth a model grounded in stacking ensemble learning, which serves to lessen feature dimensionality and forestall overfitting. Subsequently, the hyperparameter-optimized base learning machines are employed for individual prediction of engine parameters, followed by the integration of stacking for comprehensive learning to achieve engine parameter prediction. This approach, through hyperparameter optimization, amalgamates the benefits of both models, resulting in not only a significant improvement in the accuracy of fault diagnosis, but also a robustness that is both formidable and reliable. Experiments conducted indicate the feasibility and effectiveness of this method.

2. Model part

2.1 Aeroengine fault diagnosis model based on stacking ensemble learning

Given that this task pertains to a classification problem, the customary KNN and AdaBoost algorithms in machine learning are utilized to compute and optimize the hyperparameters. In this paper, hyperparameter tuning grid search is employed, whereby every possibility is tried via a circular traversal of all the candidate parameter selections, with the best parameter ultimately serving as the final outcome. In light of the fact that the accuracy of a single model is not particularly high, an integrated learning model grounded in stacking is put forth in order to enhance the performance of aeroengine prediction.

As illustrated in Fig. 1, the fault diagnosis model of stacking integrated learning, proposed in this paper, comprises two layers. The first layer is grounded in the base learner, specifically the KNN and AdaBoost algorithms following hyperparameter optimization, which in turn deploy the base learner for prediction. The second layer is built around the XGBoost meta learner, leveraging the prediction outcomes of the base model as the input for the XGBoost meta model, ultimately yielding the fault prediction outcomes of the aeroengine.

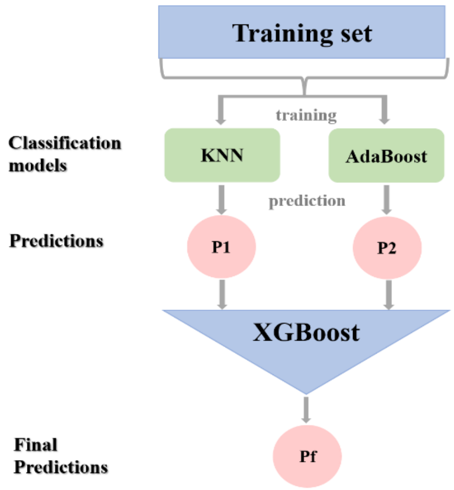

Fig.1 stacking ensemble learning model

2.2 KNN algorithm

2.2.1 Principle of KNN algorithm

K-nearest neighbor, initially introduced by Cover T et al.[20] in 1967, is regarded as one of the most traditional classification algorithms in data mining classification technology. The fundamental principle of the algorithm involves computing the distance between the unknown sample and the designated known sample through employment of the nearest sample instance as a reference. Subsequently, the K known samples that are closest to the unknown sample are selected, and the unknown sample is classified into a category that has a large proportion, in accordance with the majority-voting approach.

2.2.2 The key of KNN algorithm

1.To determine the number of nearest sample instances, one must strike a balance between overfitting and underfitting. If the K value is too large, underfitting is likely to occur, whereas if it is too small, overfitting is probable. This paper employs the KNN algorithm in scikit-learn, where the ***n_neighbors*** parameter is used to determine the K value. Thereafter, hyperparameter tuning will be conducted to ascertain the optimal K value.

2.A distance function is required to compute the distance between two samples. For example, to calculate the distance between two points A(X1, Y1) and B (X2, Y2) in a two-dimensional space, one may use the Euclidean distance or Manhattan distance formulas as given in Equation (1) and Equation (2), respectively:

$$EuclideanDistance(d) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

$$ManhattanDistance(d) = |x_2 - x_1| + |y_2 - y_1| \quad (2)$$

Euclidean distance is the most commonly used method for measuring distance, owing to its simplicity, ease of computation, and versatility. Consequently, this paper employs the Euclidean distance as the distance measurement method.

2.3 AdaBoost algorithm

2.3.1Basic principle of AdaBoost algorithm

In 1995, Freund and Schiprare[21] introduced an extended boosting algorithm called AdaBoost. The name stands for adaptive boosting, as the algorithm's adaptability is exemplified by elevating the weight of misclassified samples while reducing the weight of correctly classified ones to train the subsequent basic classifier. Each iteration supplements a new weak classifier until the minimum error rate is achieved, thereby determining the final strong classifier.

2.3.2 AdaBoost algorithm flow

1.First, the weight distribution of the initial training data. Each training example is given the same weight ω_f at the beginning，as shown in equation (3). The initial weight distribution $D_1(i)$ of the training sample set is shown in equation (4):

$$\omega_f = \frac{1}{N} \quad (3)$$

$$D_1(i) = (\omega_1, \omega_2, \dots, \omega_N) = (\frac{1}{N}, \dots, \frac{1}{N}) \quad (4)$$

2.Then iterate t=1,…,T

(1) Select a weak classifier h with the lowest error at presentas the t-th basic classifier H_t, and calculate the weak classifier h_t: $\{-1,1\}$, The error of the weak classifier D_t on is equation (5),e_t is the$H_t(x)$ misclassification sum of weight of sample. It's the sum of the weights of the misclassified samples.

$$\mathrm{e_t} = \mathrm{P(H_t(x_i) \neq y_i)} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{\omega_{ti} I(H_t(x_i) \neq y_i)} \quad (5)$$

(2) Calculate the weight of this weak classifier α in the final classifier as equation (6):

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right) \quad (6)$$

(3) Update the weight distribution D_{t+1} of the training sample as equation (7), where Z_t is the normalized constant as equation (8):

$$D_{t+1} = \frac{D_t(i)\exp(-\alpha_t y_i H_t(x_i))}{Z_t} \quad (7)$$

$$Z_t = 2\sqrt{e_t(1-e_t)} \quad (8)$$

3.Each weak classifier is combined according to the weight of the weak classifier α_t, as shown in equation (9):

$$f(x) = \sum_{t=1}^{T} \alpha_t H_t(x) \quad (9)$$

Then, a strong classifier is obtained through the function of the $sign(f(x))$, as shown in equation (10):

$$H_{final} = sign(f(x)) = sign(\sum_{t=1}^{T} \alpha_t H_t(x) \quad (10)$$

Among them, because the weight update depends on the weak classifier α and error rate e the weight update formula expressed by error rate e can be obtained:

When the sample is misclassified, it is equation (11):

$$D_{t+1}(i) = \frac{D_t(i)}{2e_t} \quad (11)$$

When the samples are paired, it is equation (12):

$$D_{t+1}(i) = \frac{D_t(i)}{2(1-e_t)} \quad (12)$$

2.4 XGBoost algorithm

XGBoost, short for eXtreme Gradient Boosting, is a machine learning project that was developed by Chen et al.[22] . It is an implementation of the boosting algorithm, which is aimed at boosting the speed and efficiency of the algorithm to an extreme level. The core algorithm follows these steps:

(1) The algorithm grows a tree by continuously splitting features, and adds one tree at a time to fit the residual of the previous prediction. The predicted value is represented by equation (13):

$$\hat{y} = \emptyset(x_i) = \sum_{k=1}^{K} f_k(x_i) \quad (13)$$

In the equation,$f(x)$ is the node weight function, It can be expressed as $f(x) = \omega_{q(x)}$, ω is the weight of the leaves.

(2) The objective function of XGBoost is a combination of the loss function and the regularization term, and can be expressed as equation (14):

$$L(\emptyset) = \sum_{i=1}^{i} l(\hat{y}_i - y_i) + \sum_{k=1}^{k} \Omega(f_k) \quad (14)$$

In the equation, $\hat{y}_i$ is the output of the entire cumulative model, the regularization term is $\sum_{k=1}^{k} \Omega(f_k)$, which is a function of the complexity of the tree, and its reified expression is $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$, T is the number of leaves, w is the fraction of a leaf node.

3. Experimental part

3.1 Experimental equipment and environment

The experimental environment of this model is shown in Table 1 below. The system environment is windows10 and the hardware environment is AMD Ryzen5 5600x CPU@3.70GHz, RAM 16G, the software environment is Python3.9.7 in the Jupiter notebook environment, and the machine learning algorithm uses the scikit-learn integrated library and XGBoost algorithm.

Table 1 experimental equipment and environment

System	Windows 10	AMD Ryzen 5 5600X CPU@3.70GHz, RAM 16G
Software	Python 3.9.7	Jupyter notebook
Packages	Data Wrangling	NumPy 1.20.3 & pandas 1.3.4
	Visualization	Matplotlib 3.4.3 & Scikit-plot 0.3.7
	ML algorithm	scikit-learn 0.24.2 & XGBoost 1.7.4

3.2 Data set introduction

3.2.1 Source and content

The information utilized in this project originates from the turbofan engine degradation simulation data set (TEDS), which is included in the ***PCoE DataSets*** provided by ***NASA***[23]. Please refer to Table 2 for a detailed file description of the original dataset.

Table 2 file description of original data set

	Data Set	Train trajectories	Test trajectories	Fault Models
1	FD001	100	100	HPC Degradation
2	FD002	260	259	HPC Degradation
3	FD003	100	100	Two(HPC Degradation, Fan Degradation)
4	FD004	248	249	Two(HPC Degradation, Fan Degradation)

The dataset simulates four distinct scenarios, each involving different combinations of operating conditions and fault modes. It records multiple sensor channels to describe the evolution of faults. Each sample comprises 26 columns of data separated by spaces. The "id" list specifies the unique ID number of the engine, while the "cycle" column indicates the operational sequence from 1 to the cycle when the fault occurs. Columns {st1, st2, st3} represent the engine's operational settings, and columns "S1" through "S21" represent the measured values from sensor 1 to sensor 21.

3.2.2 Exploratory data analysis

Prior to processing the turbofan engine dataset, this paper conducted exploratory data analysis (EDA) on both the training and test datasets. Figure 2 illustrates the variation of engines 1 through 10 on sensors 18 to 21 through a scatter diagram. It is discernible that the records of sensors s18 and s19 form a straight line, indicating that the data from these two sensors remain relatively constant.

Furthermore, the dataset contains several low-impact parameters that are similar in nature and require subsequent adjustments.

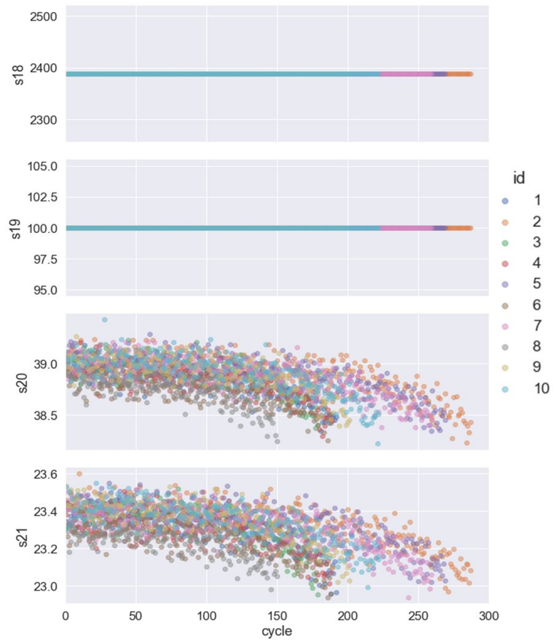

Fig. 2 scatter diagram of operation records from engine ID 1 to 10 on sensor s{18,19,20,21}

3.3 Dataset Feature Engineering

3.3.1 Dataset feature selection

As the ***standard deviation*** can signify the discreteness of a dataset, equation (15) is utilized to calculate the standard deviation:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} \tag{15}$$

The ***coefficient of variation*** can be computed through equation (16) to eliminate the influence of different units and averages on comparing the degree of variation among multiple variables:

$$C.V = \frac{S}{\bar{x}} \times 100\% \tag{16}$$

While the original features in the dataset encompass observations from settings 1 to 3 and from sensor 1 to sensor 21, some sensors' records demonstrate minimal change, as depicted in Fig. 3 and Fig. 4. Specifically, {st3, s1, s2, s5, s6, s8, s10, s13, s16, s18, s19} show little variation, and some are even close to 0. Hence, these sensor features are removed as part of the dataset's dimensionality reduction.

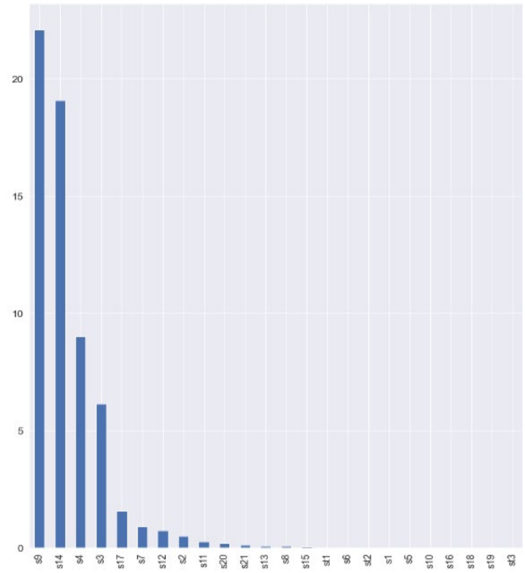

Fig. 3 Features Standard Deviation

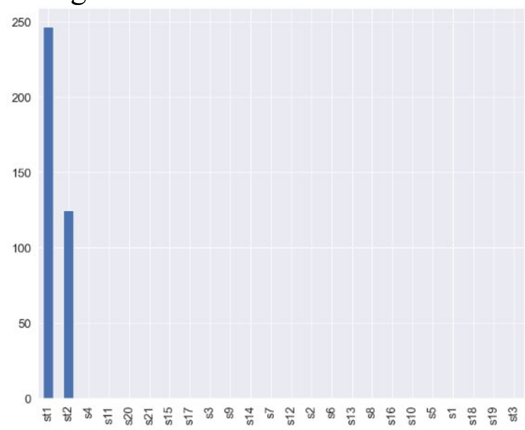

Fig. 4 Features Absolute value of Coefficient of Variation

As indicated in Table 3, after reducing the dataset's dimensionality, several artificial variables must be appended from the original data to prevent the model from overfitting. Thus, the moving average and rolling standard deviation are integrated into the last 10 operating cycles of each setting and sensor. Consequently, the dataset that incorporates these new features is referred to as the "plus dataset" for modeling experiments. Equation (17) outlines the moving average:

$$SMA_t = \frac{P_1+P_2+\cdots+P_n}{n} \tag{17}$$

Table 3 statistical summary of engine operation cycle in training data set and test data set

	Mean	Std	Min	50%	Max
Training	206.31	46.34	128	199	362
Test	75.52	41.76	7	86	145

3.3.2 Preparation for model establishment

To construct the prediction model, it is vital to establish the classification label for the binary classification problem. This paper employs binary labels 0 and 1, based on RUL (Remaining Useful Life). If RUL is greater than 30, the label is 0; otherwise, it is 1. This label serves as the basis for interpreting the accuracy of subsequent training.

3.4 Experimental setup

3.4.1 Experimental contents

In this experiment, the "plus data set" obtained after data processing is used. The experiment is divided into two stages. In the first stage, the classification calculation of single models, namely K-Nearest Neighbor (KNN) and AdaBoost, is performed separately. The optimal parameters of each model are determined through cross-validation and grid search. The best accuracy of each single model is obtained after superparameter optimization. In the second stage, stacking ensemble learning is used to further improve the accuracy and robustness of the aeroengine prediction. KNN and AdaBoost are used as base learners, and XGBoost is used as the meta learner.

In this experiment, hyperparameter optimization is carried out using grid search and cross-validation (GridSearchCV). Grid search is an exhaustive search method, where parameters are adjusted within a specified range according to certain steps to find the best parameters with the highest accuracy on the validation set. K-fold cross-validation is used, where the data set is randomly divided into k subsets with the same number of data, and each time k-1 subsets are used as the training set and the remaining subset as the test set. The validation is repeated k times, and the average accuracy is used as the evaluation metric for the model. Figure 5 shows a schematic diagram of k-fold cross-validation segmentation. In this paper, we use the method of balancing data sets, where k = 5.

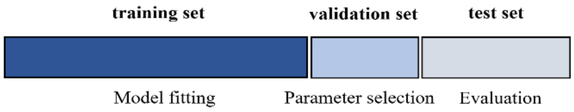

Fig.5 Schematic diagram of cross validation segmentation

3.4.2 Experimental results and analysis

This study employs two metrics to assess the efficacy of the model, specifically, accuracy (ACC) and standard deviation (STD), as per equations (18) and (15). A model with superior performance is characterized by higher ACC and lower STD values. Moreover, the present paper provides a more lucid representation of the model's predictive capacity via the confusion matrix.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{18}$$

In the single-model experiment, following the grid search hyperparameter optimization, the KNN algorithm's optimal parameters are:

{'n_neighbors': 61, 'weights': 'uniform'}

While the AdaBoost algorithm's optimal parameters are:

{'learning_rate': 4.3, 'n_estimators': 119}

Figure 6 and Figure 7 depict the confusion matrices of the KNN and AdaBoost models, respectively, after hyperparameter optimization.

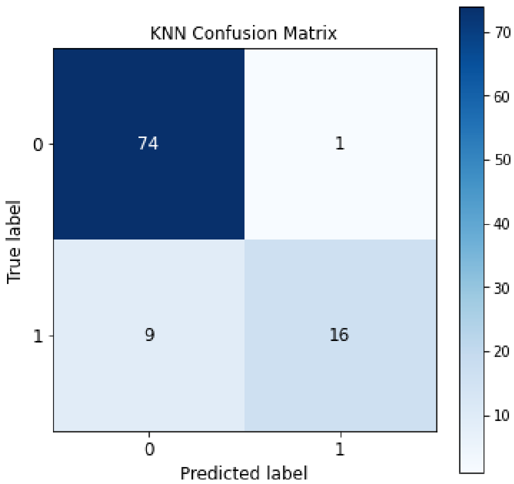

Fig.6 confusion matrix of KNN model after superparametric optimization

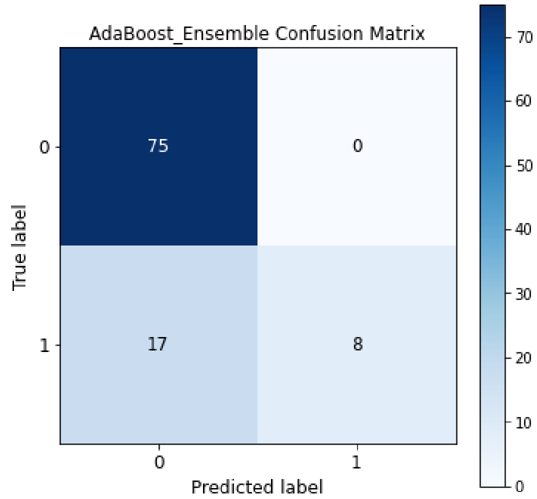

Fig.7 confusion matrix of AdaBoost model after superparametric optimization

Table 4 demonstrates that stacking integration learning has significantly enhanced the ACC, improving accuracy to 0.9591, while concurrently reducing STD to 0.0074, thereby indicating further improvement in robustness.

Table 4 test results of different learning models

	KNN	AdaBoost	Stacking-based
ACC	0.9501	0.9190	0.9591
STD	0.0084	0.0084	0.0074

4. Conclusion

This paper presents an integrated learning model based on stacking, incorporating hyperparameter optimization using grid search and cross-validation to enhance the accuracy of the base learner involved in the fusion. The resulting model employs XGBoost as the meta-learner for fault detection in aeroengines, offering a high-accuracy and robust detection method without mathematical modeling. The main conclusions are as follows:

(1) In data processing, the parameters {st3, s1, s2, s5, s6, s8, s10, s13, s16, s18, s19} are removed based on standard deviation and variation coefficient to reduce dimensionality. Moving average and rolling standard deviation are subsequently added in the last 10 operating cycles of the sensor to prevent overfitting.

(2) By employing grid search and cross-validation, the optimal parameters for a single model are obtained, leading to improved prediction performance.

(3) To address the shortcomings of low accuracy and high discreteness of a single model, stacking-based ensemble learning is used to integrate hyperparameter-optimized models of KNN and AdaBoost, further enhancing prediction performance. The resulting prediction accuracy of 0.9591 is significantly higher than the maximum value of 0.9501 for a single model. The prediction standard deviation of 0.0074 is also lower than 0.0084 of a single model, establishing the reliability of the ensemble learning model's prediction results.

References

[1] Liu Daxiang, Jin Jie. Development trend and Prospect of world aero power technology in the 21st century [j]. China Engineering Science, 2004, (09): 1-8

[2] Luo fan Research on the formation mechanism and early warning system of aviation disasters [d]. Wuhan University of technology, 2004

[3] Xueyu Jingyang Research on the causes of general aviation flight accidents in China [d]. China University of mining and Technology (Beijing), 2019.doi:10.27624/d.cnki.gzkbu.2019.000057

[4] General editorial board of Aeroengine Design Manual. Aeroengine Design Manual. Volume 19, rotor dynamics and whole machine vibration ［ m ］. Beijing: Aviation Industry Press, 2000

[5] Wang Yue. Aeroengine fault diagnosis method [j]. China Science and technology information, 2023, no.692 (03): 19-21

[6] Zhang Peng Research on Aeroengine Fault Diagnosis Technology Based on Kalman filter [d]. Nanjing University of Aeronautics and Astronautics, 2009

[7] Lambert h. A simulation study of turbofan engine depth estimation using Kalman filtering techniques[r] one thousand nine hundred and ninety-one

[8] Borguet s, dewallef P, Le ´ onard O. a way to deal with model plant mismatch for a reliable diagnosis in transient operation[c]//turbo expo: power for land, sea, and air 2006, 42371: 593-602

[9] Wan Yucheng, sun Qian, yuanwenjun Aeroengine wear fault diagnosis method based on extension correlation function [c]//northeast University, China, IEEE Industrial Electronics (ie) chapter, Singapore, China University of mining and technology.procedures of 2010 Chinese control and decision conference [publisher unknown], 2010:4

[10] Zhaohongli, Zhang Meng. Research on aero engine performance degradation based on stochastic Wiener process [j]. propulsion technology, 2021,42 (03): 488-494.doi:10.13675/j.cnki.tjjs.190632

[11] Hujinhai, Yu Zhiguo, zhaixusheng, et al. Research on aeroengine rotor fault decision fusion diagnosis based on Improved D-S evidence theory [j]. Acta Aeronautica Sinica, 2014,35 (02): 436-443

[12] Han Z, Wang W. design of fault tolerant control algorithm for aeroengine performance degradation[c]//2021 33rd Chinese control and Decision Conference (CCDC) IEEE, 2021: 1296-1301

[13] Xuqihua, Shi Jun. aeroengine fault diagnosis based on support vector machine [j]. Journal of Aeronautical dynamics, 2005 (02): 298-302.doi:10.13224/j.cnki.jasp.2005.02.025

[14] Wei xunqai, Lu Bo, Wang Cheng, Lu Jianming, Li Yinghong. Application of support vector machine in aeroengine fault diagnosis [j]. Journal of Aeronautical dynamics, 2004 (06): 844-848.doi:10.13224/j.cnki.jasp.2004.06.020

[15] Wujunfeng. Civil aviation engine fault diagnosis based on twin support vector machines [j]. electronic measurement technology, 2020,43 (22): 132-136.doi:10.1965 1 / j.cnki.emt.2005062

[16] Zhuyongxin Research on fault diagnosis of aero engine oil system based on improved support vector machine [d]. Civil Aviation University of China, 2020.doi:10.27627/d.cnki.gzmhy.2020.000196

[17] Islam m m m m, prosvirin a e, Kim J M. data driven diagnostic scheme for rolling element bearings using a new health index and variations of least square support vector machines[j] Mechanical systems and signal processing, 2021, 160: 107853

[18] Sun Xiaoyu Research on prediction of aeroengine gas path performance parameters based on data driven [d]. Dalian University of technology, 2019.doi:10.26991/d.cnki.gdllu.2019.001504

[19] Zedda m, Singh R. fault diagnosis of a turbofan engine using neural Networks-A quantitative approach[c]//34th aiaa/asme/sae/asee joint propulsion conference and exhibit 1998: 3602

[20] Cover t, Hart P. nearest neighbor pattern classification[j] IEEE Transactions on information theory, 1967, 13 (1): 21-27

[21] Yoav Freund, Robert E schapire A decision theoretical generalization of on line learning and an application to boosting[j] Journal of computer and system sciences, 1997, 55 (1)

[22] Chen T, he T. Higgs boson discovery with boosted trees[c]// jmlr: workshop and conference proceedings, 2015, 42: 69 – 80

[23] Saxena a, Goebel K, Simon D, et al. damage propagation modeling for aircraft engine run to failure simulation[c]//2008 International Conference on diagnostics and health management IEEE, 2008: 1-9