Engineered Science
DOI: https://dx.doi.org/10.30919/es1696

# Evaluating the Performance of Generative Adversarial Network, Synthetic Minority Oversampling Technique, and Adaptive Synthetic Sampling in Marine Diesel Engine Fault Diagnosis using Vibration Data

Zhijun Chen,[1, 2] Ziyu Xiao,[2] Zhongjun Wang,[1, 2,*] Pengcheng Wang,[2] Jie Wang,[3] Yijie Qu,[3,*] Xiong Bao[3] and Xiaofeng Guo[4]

## Abstract

Machine learning-driven fault diagnosis of marine diesel engines is a major research focus, challenged by scarce, insufficient fault data. This study explores vibration data under five conditions: normal operation, single-cylinder ignition failure, piston ring wear, nozzle blockage, and exhaust valve leakage. It employs GAN (Generative Adversarial Network), SMOTE (Synthetic Minority Oversampling Technique), and ADASYN (Adaptive Synthetic Sampling) to augment data, addressing imbalance. Principal Component Analysis (PCA) with Cosine Similarity (CS) and Pearson Correlation Coefficient (PCC) qualitatively and quantitatively analyzed augmentation effects. Classification performance was evaluated using Support Vector Machine (SVM), Decision Tree, and Random Forest via accuracy, precision, recall, and F1-score. Experiments showed GAN, by enhancing sample diversity and realism, outperformed SMOTE and ADASYN, increasing average CS by 0.04-0.08 and PCC by 0.05-0.09 across faults. Compared to the original dataset, GAN-enhanced data improved the three classifiers' accuracy by ~3.3%, 7.6%, and 5.8%, with the most significant gains in other metrics, supporting real-world marine diesel fault diagnosis.

*Keywords:* Fault diagnosis; Marine diesel engine; Data augmentation; GAN; SMOTE; ADASYN.

Received: 24 July 2025; Revised: 25 August 2025; Accepted: 26 August 2025.
Article type: Research article.

## 1. Introduction

Marine diesel engines are the core power for ship navigation, and monitoring their operating status is a hot topic in current research. With the increasing demand for ship intelligence, various machine learning algorithms have been applied to the fault diagnosis of diesel engines. Researchers build intelligent models for fault diagnosis to reduce dependence on manual experience.[1-6] However, the imbalance in training samples remains a key constraint on the performance of machine learning models. This imbalance manifests in two primary forms: first, the data disparity between samples of different fault types, and second, the uneven distribution between fault samples and normal samples. The reason for the imbalance phenomenon is mainly that the diesel engine usually operates in a normal state, with few faults occurring, while the cost of simulating fault experiments in the laboratory to collect fault data is high. Using unbalanced data sets to train machine learning models easily leads to model overfitting to the characteristics and noise of the majority class samples, causing overfitting and making the model more inclined to predict the more frequent categories. The small number of minority class samples also leads to insufficient sample diversity to portray a complete sample distribution.[7-10] These problems will reduce

[1] *Key Laboratory of High Performance Ship Technology, Wuhan University of Technology, Ministry of Education, Wuhan, 430063, China*
[2] *School of Naval Architecture, Ocean and Energy Power Engineering, Wuhan University of Technology, Wuhan, 430063, China*
[3] *School of Automobile and Traffic Engineering, Hubei Provincial Engineering Research Center of Advanced Chassis Technology for New Energy Vehicles, Wuhan Scientific and Technological Achievements Transformation Pilot Platform (Base) of Automotive Intelligent Sensor, Wuhan University of Science and Technology, Wuhan, 430065, China*
[4] *Université Paris Cité, CNRS, LIED UMR 8236, Paris, F-75006, France*
* *E-mail*: zhiw153624@163.com (Z. Wang); 1638805926@qq.com (Y. Qu)

the generalization ability of the model. To reduce the impact of unbalanced data sets, data augmentation of minority class samples is an effective method.

Generative adversarial network (GAN) is a popular data generation data augmentation method that has emerged with the development of the machine learning field in recent years, proposed by Ian Goodfellow *et al*. in 2014.[11] GAN updates the parameters of the generator and discriminator by setting a confrontation between the two neural networks, gradually making the generated fake samples approach real samples during iterations. Jinhai Liu *et al*. used GAN to develop a coarse fault data refiner for wind turbine fault detection, making coarse fault data more similar to real fault data, and achieved good results by using the generated data to train AI-based fault detection models.[12] Yun Gao *et al*. combined the finite element method (FEM) and GAN to obtain synthetic rotor-bearing fault samples as training samples, using different AI models for classification and achieving high classification accuracy.[13] Pengfei Liang *et al*. combined wavelet transform (WT), GAN, and convolutional neural networks (CNN) to propose a WT-GAN-CNN fault detection method for rotating machinery. Experimental results proved the method's effectiveness, showing that the algorithm's test accuracy could still be higher than other intelligent fault detection methods in the literature even under strong environmental noise interference.[14] Luo *et al*. proposed a fault diagnosis method based on two-stage generative adversarial network (2S GAN), which provides an effective solution for fault diagnosis in data imbalance scenarios by combining the time series GAN and auxiliary classifier GAN features and introducing continuous wavelet transform with ResNet18 classification model.[15] Zhenglin Dai *et al*. proposed a Con-GAN algorithm based on GAN, which can realize the real continuation of existing signals and integrate the continuation signals with the original signals. Validation on multiple data sets proved that Con-GAN could generate new signals with better validity and diversity.[16] Bingxi Zhao *et al*. introduced an auxiliary classifier to promote the training process of GAN and designed an online sample filter to ensure the accuracy of generated samples.[17] Cui Dongjin *et al*. trained and compared the prediction performance of three GAN models for pollutant diffusion in two scenarios, and explored the influence of two optimization methods (training stability and training data optimization) on the model performance. The results show that the clustered dataset can reflect the real Shenzhen residential blocks characteristics.[18]

Synthetic minority over-sampling technique (SMOTE) is an interpolation-based random oversampling method proposed by Nitesh V. Chawla *et al*. in 2002.[19] Rao Shaowei *et al*. used SMOTE to solve the imbalance problem of the transformer fault diagnosis data set, combining artificial neural networks (ANN) as classifiers and studying high-accuracy ANN feature combinations and activation functions.[20] Wei Chang *et al*. validated the improvement effect of SMOTE and Gaussian SMOTE on fault classification models through real circulating water system sensor data from power plants. Experiments showed that both SMOTE variant algorithms could effectively improve fault model performance and achieve efficient power generation.[21] Yuqiang Fan *et al*. used principal component analysis (PCA) and SMOTE to oversample the fault data of centrifugal chillers, performing fault diagnosis through SVM, finding that the diagnostic performance of the chiller improved, and by changing the sampling ratio from 100% to 400%, they found that the 100% oversampling ratio achieved the highest diagnostic accuracy of 96.7%.[22] Liu *et al*. proposed a feature-level SMOTE method for the problem of overlapping synthetic samples with normal samples due to traditional data augmentation, mapping the data to a feature space with stronger inter-class separability via deep Siamese multi-head self-attention network (DSMHSA), and then using the SMOTE method to generate synthetic faulty samples to balance the dataset. [23] Zihao Li *et al*. used ADASYN to oversample fault data of power transformers, making the data set more balanced, using an improved deep coupled dense convolutional neural network (CDCN) to judge the fault status of power transformers, achieving high fault diagnosis accuracy.[24] Hailong Cao *et al*. compared the effectiveness of four algorithms (weighted cross-entropy loss, random oversampling, random undersampling, and ADASYN) in improving the sensitivity of the ANN model predicting harmful substances in groundwater. The experiment showed that all four algorithms produced more accurate predictions, with ADASYN performing the best.[25] Wenzhe Yin *et al*. used ADASYN to synthesize multi-channel vibration data to expand unbalanced samples for fault diagnosis technology of nuclear power plant rotating machinery, using ensemble empirical mode decomposition (EEMD) and continuous wavelet transform (CWT) to convert vibration data into time-frequency images, constructing a deep residual neural network to achieve fault diagnosis. Experiments proved that this method could achieve good diagnostic performance under different degrees of unbalanced samples.[26] Xu Hongsheng *et al*. applied SMOTE to address the imbalance issue in the dataset. The results indicated that the performance of the ensemble model was superior to that of individual models, with an accuracy score of 99.58%.[27]

Although the above-mentioned research has achieved certain results, there are still limitations. Some methods are insufficient in the recognition of complex fault modes and have difficulty accurately distinguishing multiple concurrent fault types. Some models have poor adaptability to changes in working conditions. When the working conditions frequently change during the actual operation of ships, the diagnostic accuracy drops significantly. Some methods rely on a large amount of data under specific working conditions, resulting in high data acquisition costs and limited universality. In order to face and solve the problem of non-equilibrium data of marine diesel engine fault samples in realistic scenarios, this paper adopts GAN, SMOTE, and ADASYN algorithms to perform data augmentation on non-equilibrium samples of the dataset,

and then analyzes and discusses the impact of each data augmentation method on the fault diagnostic results through SVM, decision tree, and random forest classifiers, and takes the accuracy, precision, recall, and F1 score as the classification model evaluation indexes. Meanwhile, the analysis qualitatively and quantitatively analyzes the diversity and authenticity of the data generated by GAN, SMOTE, and ADASYN by combining Principal Component Analysis (PCA) with Cosine Similarity (CS) and Pearson's Correlation Coefficient (PCC) value calculations, and the experiments on Z6170ZICZ-1 diesel engine show that GAN is more helpful for the fault diagnosis model compared to the other two methods. It provides technical reference and theoretical basis for solving the data augmentation of marine diesel engine fault data, and at the same time, it has an important application value for improving the operation reliability and intelligent operation and maintenance level of ship power system.

## 2. Basic methods and theory

### 2.1 Data generation and augmentation algorithm GAN

GAN require the setup of two neural networks, serving as the generator ($G$) and the discriminator ($D$). The generator $G(z;\theta_G)$ randomly selects data from noise that follows a prior distribution (such as noise signals following a standard normal distribution) as input signal $z$, and outputs fake samples $G(z)$, which have the same length as the real samples. The discriminator $D(x;\theta_D)$ is a binary classification neural network that returns an estimate $D(x)$ based on the input sample $x$ to determine whether $x$ is a real sample or a fake sample. If Sigmoid is used as the classification function of the last layer of the discriminator $D$, then the range of $D(x)$ is [0, 1], with higher output values indicating a higher probability that $x$ is a real sample, and lower values indicating a higher probability that $x$ is a fake sample. The structure of GAN is shown in Fig. 1.

The loss functions for the discriminator $D$ and the generator $G$ can be expressed as Eqs. (1) and (2):

$$D_{loss} = -\log D\,(x_{real}) - log(1 - D(x_{fake})) \quad (1)$$

$$G_{loss} = -\log D\,(x_{fake}) \quad (2)$$

where $x_{real}$ represents the real data input, and $x_{fake} = G(z)$ represents the fake data input.

The parameter update equation for the discriminator $D$ and the generator $G$ can be expressed as Eqs. (3) and (4):

$$\theta_D = \theta_{D-1} + \widehat{\alpha_D} \times \nabla_{\theta_D} \frac{1}{N} \sum_{i=1}^{N} (log\, D\,(x_{real\_i}) + log(1 - D(x_{fake\_i}))) \quad (3)$$

$$\theta_G = \theta_{G-1} + \widehat{\alpha_G} \times \nabla_{\theta_G} \frac{1}{N} \sum_{i=1}^{N} log\, D\,(x_{fake\_i}) \quad (4)$$

where $\widehat{\alpha_D}$ and $\widehat{\alpha_G}$ represent the gradient update steps of the discriminator $D$ and the generator $G$ after being corrected by the optimizer.

From Eqs. (3) and (4), it can be seen that through iteration, the discriminator $D$ gives larger outputs for real samples and smaller outputs for fake samples. This means that the discriminator can more accurately classify the input data $x$ as real or fake. The generator $G$, through iteration, generates fake samples that are closer to real samples, causing the discriminator $D$ to mistakenly classify fake samples as real samples.

The iterative process of the GAN's discriminator $D$ and generator $G$ is as follows:

Step 1: Input randomly sampled noise data $z$;

Step 2: Input the noise through the generator $G(z;\theta_G)$, and output fake samples $G(z)$;

Step 3: Input the fake samples $G(z)$ and real samples $x_{real}$ into the discriminator $D(x;\theta_D)$;

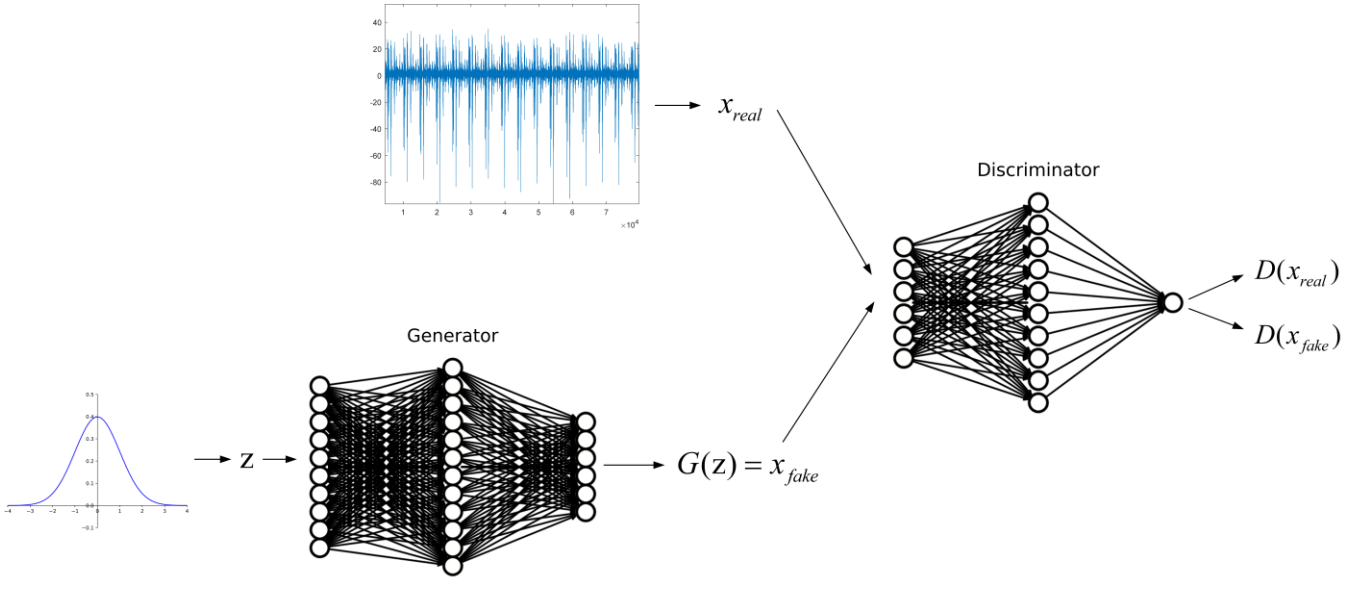



**Fig. 1:** GAN structure diagram.

Step 4: Calculate the generator's loss function according to Eq. (2), backpropagate the gradient according to the chain rule, and update parameters according to Eq. (4) to iterate the generator;

Step 5: Calculate the discriminator's loss function according to Eq. (1), backpropagate the gradient according to the chain rule, and update parameters according to Eq. (3) to iterate the discriminator;

Step 6: When the number of iterations $K$ reaches the set iteration threshold, stop the iteration and output the fake samples $G(z)$.

The iterative flow chart is shown in Fig. 2. When the discriminator $D$ can no longer distinguish between real samples and generated fake samples, it can be considered that the training of the generator $G$ is complete. At this point, theoretically, Eq. (5) should be satisfied:

$$D(x_{real}) = D(x_{fake}) = 0.5 \tag{5}$$

In the backpropagation process of the GAN network, the Adam gradient optimizer is employed to optimize gradients and learning rates. This optimizer replaces raw gradients with momentum, while computing exponential moving averages and bias corrections for both momentum and learning rates. This makes the gradient update process smoother and convergence faster.[28] The hyperparameters for the exponential moving average of momentum and learning rate in Adam are $\beta_1$ and $\beta_2$.

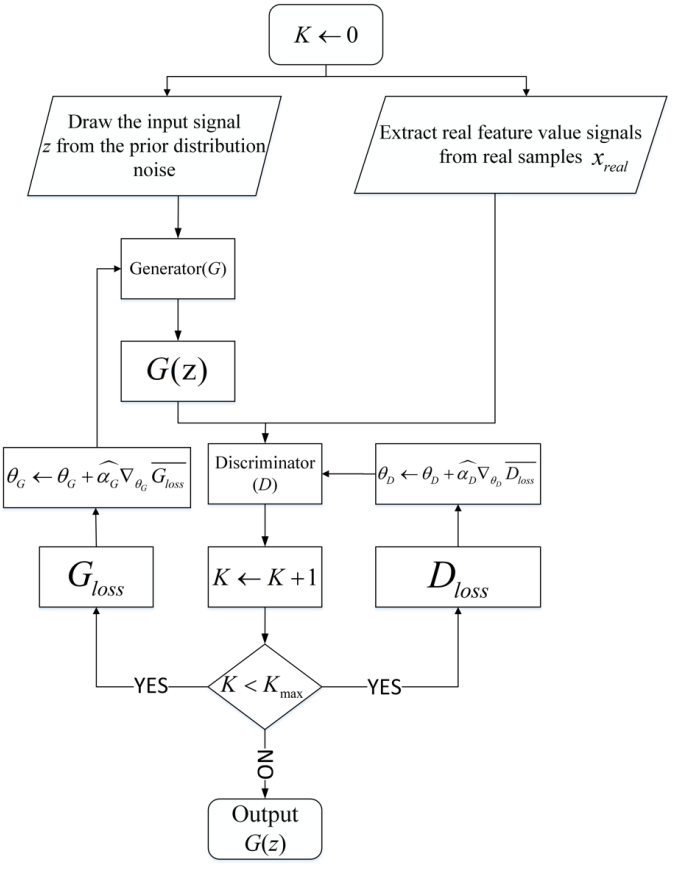



**Fig. 2:** GAN iterative flow chart.

## 2.2 Oversampling-based data augmentation algorithm

SMOTE (Synthetic Minority Over-sampling Technique) is an interpolation-based random oversampling method. The algorithm process is as follows:

Step 1: Let the number of minority class samples be $m_s$, and the number of majority class samples be $m_l$. Determine the sampling ratio as Eq. (6):

$$N = \left\lceil \frac{m_l}{m_s} \right\rceil \tag{6}$$

and set the number of nearest neighbor samples to be selected, $k$, where $k \geq N$;

Step 2: Calculate the Euclidean distance between each minority class sample and other minority class samples in the feature space to obtain its $k$ nearest minority class neighbors;

Step 3: For each minority class sample $x_i$, first randomly select $N$ samples from its $k$ nearest minority class neighbors, which are denoted as $x_{i\_N}$. Subsequently, based on the interpolation method, new sample points $\widetilde{x_{i\_N}}$ are randomly generated along the line segment connecting $x_i$ and each sample in $x_{i\_N}$. The calculation formula is as Eq. (7):

$$\widetilde{x_{i\_N}} = x_i + rand(0,1) \times (x_{i\_N} - x_i) \tag{7}$$

Step 4: Randomly select samples from the generated sample set that meet the required quantity $(m_l - m_s)$ and add them to the original sample set.

The algorithm principle is illustrated in Fig. 3:

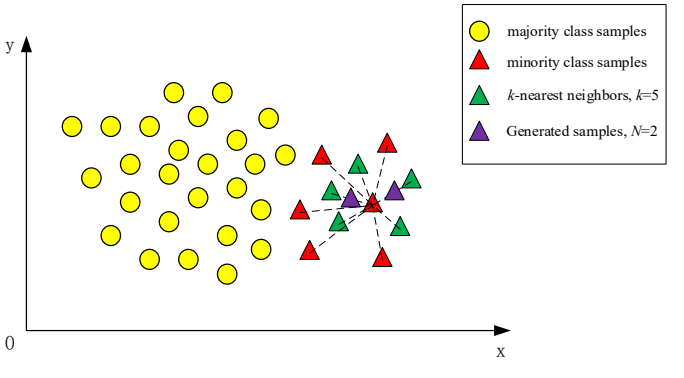



**Fig. 3:** SMOTE algorithm schematic diagram.

ADASYN first determines the total number of minority class samples that need to be generated. Then, it adaptively determines the quantity of minority class samples to be generated around each minority class sample, based on the number of majority class samples in the latter's local neighborhood. The more majority class samples around a minority class sample, the higher the classification difficulty and learning difficulty. ADASYN assigns greater weight to such samples and generates more minority class samples around them. The ADASYN algorithm process is as follows:

Step 1: Determine the total number of minority class samples that need to be generated. Let the number of minority class samples be $m_f$, and the number of majority class samples be $m_t$, where $m_f \leq m_t$. Set the desired level of balance as $\beta$,

$\beta \in [0,1]$. When $\beta = 1$, complete balance between minority and majority class samples is achieved. The total number G of the minority class samples to be generated is calculated through Eq. (8):

$$G = (m_t - m_f) \times \beta \tag{8}$$

Step 2: Calculate the weight of each minority class sample. For each minority class sample $x_i$, use Euclidean distance to calculate $k$-nearest neighbors. The number of most class samples is denoted as $\Delta_i$, and the weight $r_i$ is calculated by Eq. (9):

$$r_i = \frac{\Delta_i}{k}, i = 1,2,\dots,m_f \tag{9}$$

where $r_i \in [0,1]$. Normalize $r_i$ as Eq. (10):

$$\hat{r_i} = \frac{r_i}{\sum_{i=1}^{m_f} r_i} \tag{10}$$

where $\sum_{i=1}^{m_f} \hat{r_i} = 1$.

Step 3: The number of samples to be generated for each minority class sample $g_i$ is calculated by Eq. (11):

$$g_i = G \times \hat{r_i} \tag{11}$$

Step 4: Randomly select $g_i$ minority class samples from the $k$-nearest neighbor samples of $x_i$, denoted as $x_{i\_g_i}$. Using the interpolation method, randomly generate new sample points $\widetilde{x_{i\_g_i}}$ on the line between $x_i$ and $x_{i\_g_i}$. Calculated by Eq. (12):

$$\widetilde{x_{i\_g_i}} = x_i + rand(0,1) \times (x_{i\_g_i} - x_i) \tag{12}$$

The algorithm principle is illustrated in Fig. 4:

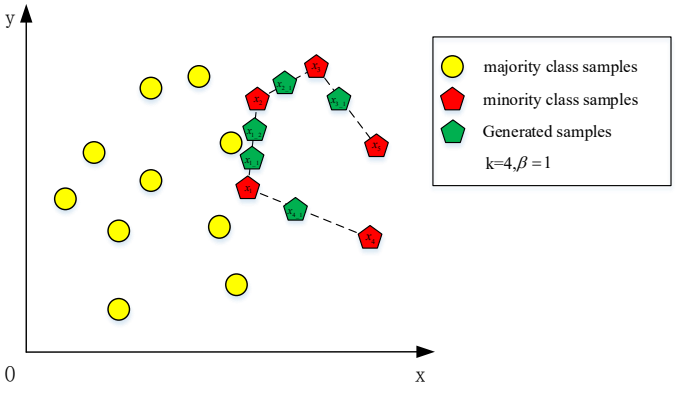



**Fig. 4:** ADASYN algorithm schematic diagram.

### 2.3 Classification methods and evaluation metrics

To more accurately compare the diversity and authenticity of the data generated by different data augmentation methods, this study employs various classifiers such as SVM,[29] decision tree,[30] and random forest.[31]

SVM identify an optimal hyperplane in the feature space to separate samples belonging to different categories. Renowned for its effectiveness in small-sample classification tasks and its ability to yield highly interpretable classification outcomes, SVM has emerged as a widely adopted classification method in machine learning. Decision trees, based on information entropy, quickly and accurately match classification labels by dividing features into multiple subsets. Decision trees perform well on large data sets, but for data with inconsistent sample sizes across categories, the results of information gain tend to favor features with more values, which may lead to overfitting. Random forest is an ensemble classifier based on decision trees. It builds multiple decision trees by randomly selecting features and samples, and determines the final classification result by voting among the multiple decision trees. This approach improves classification accuracy and stability while reducing the risk of overfitting.

Evaluating the performance of a classification model is a fundamental problem in fault diagnosis, and this paper introduces statistical performance evaluation metrics to evaluate the model performance. Accuracy is the ratio of the number of correctly predicted samples to the total number of predicted books, which is usually used to measure the overall classification performance of the model, but it is not precise enough for a certain category. Therefore, in addition to Accuracy, this paper also introduces Precision, Recall and F1-Score indicators, Precision is also known as checking accuracy rate, which is the proportion of correctly predicted positive samples among all predicted positive samples, Recall is the proportion of correctly predicted positive samples among all actual positive samples, and the higher Recall is, the more positive samples are detected by the model monitor. The higher the Recall, the more positive samples are detected by the model monitor, and the fewer faults are misdiagnosed. F1-Score combines Precision and Recall, and can be regarded as the reconciled mean of the two. All of these evaluation metrics are calculated based on the confusion matrix, which is a standard format for model accuracy evaluation, and is used as a visual tool to analyze the true labels and predictions of a classification problem, taking the binary classification problem as an example, as defined in Table 1.

**Table 1:** Confusion matrix definition.

| Predicted \ Actual | Actually Positive | Actually Negative |
|---|---|---|
| Predicted Positive | TP (True Positive) | FP (False Positive) |
| Predicted Negative | FN (False Negative) | TN (True Negative) |

where TP——True Positive; FN——False Negative; FP——False Positive; TN——True Negative.

Other performance indicators are calculated using the above parameters by Eqs. (13-16):

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (13)$$

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

$$Recall = \frac{TP}{TP+FN} \quad (15)$$

$$F1-Score = \frac{2\times(Recall\times Precision)}{Recall+Precision} \quad (16)$$

In addition, the quality of the generated samples is crucial for data augmentation. For image tasks, only judgment can be made by observing the similarity through visual assessment, while for time series thought data, it is difficult to directly observe the similarity. Therefore, this paper evaluates the quality of generated samples by comparing the similarity of moment distributions of vibration signals, and selects two commonly used data relevance evaluation metrics, which are Cosine Similarity (CS) and Pearson Correlation Coefficient (PCC). Where CS evaluates the similarity of the sample data distribution by calculating the cosine value of the vector pinch angle, the closer the value is to 1, the more similar the generated sample is to the real sample. Similarly, PCC calculates the linear correlation between variables, and the closer the value is to 1, the higher the similarity, it is calculated by Eqs. (17) and (18).

$$CS = cos(\theta) = \frac{AB}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\times\sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (17)$$

$$PCC = \rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (18)$$

where $A_i$ and $B_i$ are the components of vectors $A$ and $B$, $X$ and $Y$ denote the two vectors, $cov(*)$ denotes the covariance, $\sigma$ denotes the standard deviation.

## 3. Experiments and data preprocessing

### 3.1 Experimental design and data sources

The research focus is on the Z6170ZICZ-1 Marine diesel engine, as shown in the supplementary materials. Fault experiments were conducted by simulating fault conditions to obtain cylinder head vibration fault signals under four fault conditions: single-cylinder misfire, piston ring wear, nozzle blockage, and exhaust valve leakage. The engine operating conditions were set to a rated speed of 1000 r/min with 0% load.

Single-cylinder misfire is often caused by abnormal increases in oil pressure in the injection system, leading to high-pressure oil pipe joint rupture. Misfire under high load can cause severe damage to engine components such as the crankshaft. Piston rings are commonly used to seal between the combustion chamber and the oil sump. After piston ring wear, high-pressure gas from the combustion chamber leaks, causing a drop in engine power. Nozzle blockage can prolong the fuel injection cycle, disrupt the injection pattern, and result in incomplete fuel combustion, leading to decreased engine performance and deteriorated emissions. Exhaust valve leakage is mainly caused by the impact and friction between the exhaust valve and the valve seat when the valve is closed. Exhaust valve leakage can cause uneven work among cylinders, resulting in decreased engine power [32]. The fault types and corresponding experimental design are shown in Table 2.

### 3.2 Data preprocessing

The experimental diesel engine works in a two-turn cycle, and the timing diagram of each simulated fault state and normal state acquired signal is shown in Fig. 6. In order to fully utilize the limited experimental data, this study first samples the original vibration signal by sliding window overlap sampling, that is, through a fault length window sliding on the time series data, each time sliding a certain step length, when the sliding step length is smaller than the window length will form the overlap, and finally the continuous signal will be segmented into a number of overlapping sub-sequences. In this chapter, the data length is set to 1024 each time, the window overlap rate is 0.5, and the sliding window overlap sampling schematic is shown in Fig. 5. The number of samples of each health state collected in the fault simulation experiments is shown in Table 3, in which the sample imbalance ratio is 1:7. As can be seen in Figs. 6(a-e), the signals collected in different health states have different vibration characteristics, which also provides a basis for the realization of fault diagnosis.

The eigenvalues of fault signals carry substantial information regarding the faults present. To address the limitations of relying solely on time-domain or frequency-domain features in capturing the intricate characteristics of fault signals, this study conducts a time-frequency analysis of

**Table 2:** Fault type and experimental design.

| Fault content | Experimental design |
|---|---|
| Single-cylinder misfire | Remove the high-pressure oil pipe of the first cylinder injector |
| Piston ring wear | Wear the inner ring of the first gas ring by 0.4 mm and the ring gap by 2 mm |
| Nozzle blockage | Wear the inner ring of the first gas ring by 0.4 mm and the ring gap by 2mm |
| Exhaust valve leakage | Cut two 1 mm × 6 mm slots in the exhaust valve |

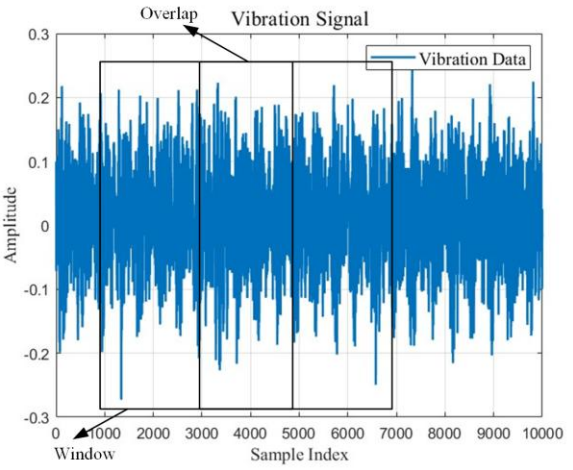



**Fig. 5:** Sliding window overlap sampling.

**Table 3:** Number of samples by category.

| Sample label | Sample quantity |
|---|---|
| Single-cylinder misfire | 16 |
| Piston ring wear | 16 |
| Nozzle blockage | 16 |
| Exhaust valve leakage | 16 |
| Normal | 112 |

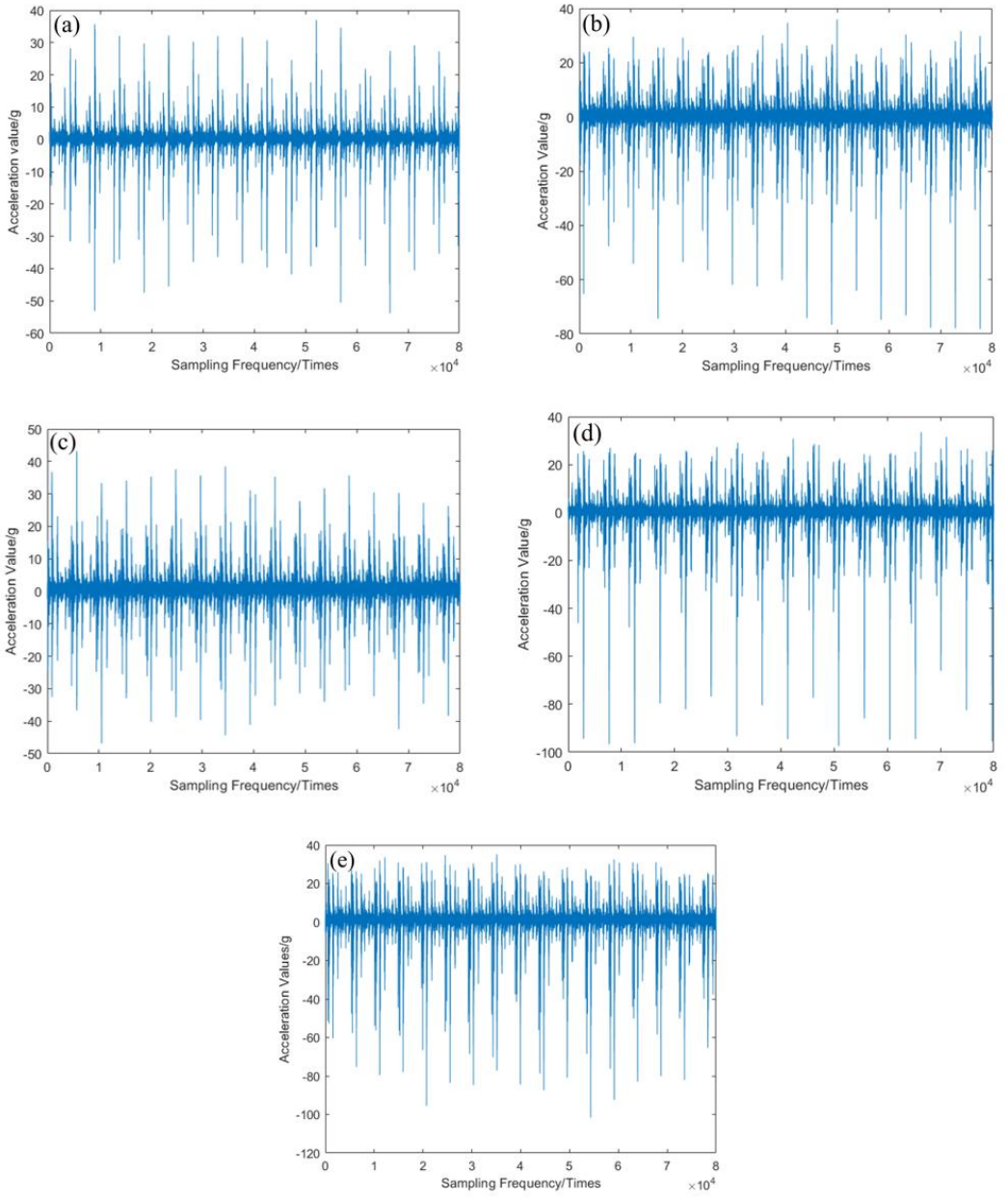



**Fig. 6:** Time series diagram of 4 diesel engine faults and normal conditions, (a) single-cylinder misfire, (b) piston ring wear, (c) nozzle blockage, (d) exhaust valve leakage, (c) normal.

the acquired fault data. We analyze the signals using a comprehensive set of statistical parameters, including Mean, Root Mean Square (RMS), Impulse Factor, Kurtosis Coefficient, Peak Factor, Skewness, Spectral Centroid, Spectral Bandwidth, and Spectral Skewness, among twelve other parameters. By integrating these diverse time and frequency domain features, we create a fusion feature parameter vector that serves as the initial set of features for the signal analysis, thereby overcoming the shortcomings inherent in single-domain analyses. Among them, the time-domain feature parameters such as the mean value can react to the vibration signal offset, the RMS value can measure the vibration signal in a period of time the size of the energy, the impulse factor, the craggy coefficient and the peak factor of the impact class of faults have a high sensitivity, and can effectively react to the signal in the sudden anomaly. As for the frequency domain parameters, the spectral center of mass reflects the concentration trend of the signal energy on the frequency axis, the spectral bandwidth reflects the distribution range of the signal frequency components, and the spectral skewness further describes the asymmetry of the spectral distribution. These time-domain and frequency-domain feature parameters complement each other to characterize the fault signal from different perspectives, making up for the inherent defects of a single analysis domain. Based on this, the subsequent data augmentation and fault classification are carried out based on these eigenvalues. The formula for calculating the time and frequency domain eigenvalues is shown in Table 4, where $x_i$ denotes the time-domain vibration signal value obtained from sampling.

After completing the feature extraction step, each sample consists of twelve-dimensional time and frequency domain

**Table 4:** Time domain characteristic index.

| Time-domain Parameter | Calculation Formula | Frequency-domain Parameter | Calculation Formula |
|---|---|---|---|
| Mean ($\bar{x}$) | $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ | Spectral Centroid (SC) | $SC = \frac{\sum_{k=1}^{N} f_k \lvert X_k\rvert^2}{\sum_{k=1}^{N} \lvert X_k\rvert^2}$ |
| Root Mean Square (RMS) | $RMS = \sqrt{(\frac{1}{N}\sum_{i=1}^{N} x_i{}^2)}$ | Spectral Bandwidth (SB) | $SB = \sqrt{\frac{\sum_{k=1}^{N}(f_k - SC)^2 \lvert X_k\rvert^2}{\sum_{k=1}^{N} \lvert X_k\rvert^2}}$ |
| Impulse Factor (I) | $I = \frac{max(\lvert x_i\rvert)}{\frac{1}{N}\sum_{i=1}^{N} x_i}$ | Spectral Skewness (SS) | $SS = \frac{\sum_{k=1}^{N}(f_k - SC)^3 \lvert X_k\rvert^2}{SB^3 \sum_{k=1}^{N} \lvert X_k\rvert^2}$ |
| Kurtosis Coefficient (Ku) | $Ku = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left[\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right]^2}$ | Spectral Kurtosis (SK) | $SK = \frac{E[\lvert X(f,t)\rvert^4]}{(E[\lvert X(f,t)\rvert^2])^2} - 2$ |
| Peak Factor (C) | $C = \frac{max(\lvert x_i\rvert)}{X_{rms}}$ | Spectral Entropy (SE) | $SE = -\sum_{k=1}^{N} p_k \log_2 P_k, P_k = \frac{\lvert X_k\rvert^2}{\sum_{i=1}^{N} \lvert X_i\rvert^2}$ |
| Skewness (Sk) | $Sk = \frac{\frac{1}{N}\sum_{i=1}^{N} x_i^3}{\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right)^{3/2}}$ | Band Energy Ratio (BER) | $BER = \frac{\sum_{k\in Band} \lvert X_k\rvert^2}{\sum_{k=1}^{N} \lvert X_k\rvert^2}$ |

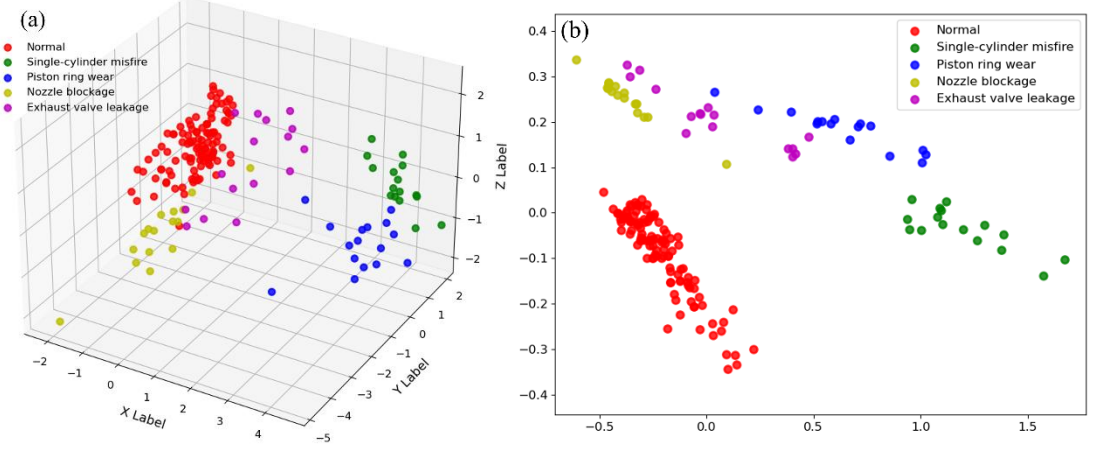



**Fig. 7:** Distribution of all kinds of samples after PCA dimension reduction, (a) 3D spatial distribution diagram and (b) 2D spatial distribution diagram.

component distribution. In order to deeply analyze the distribution patterns of different samples in the high-feature vectors, which cover the key information of the samples in terms of time series variation and frequency dimensional feature space, this study adopts the classic linear dimensionality reduction algorithm of Principal Component Analysis (PCA),[33,34] which maps the original high-dimensional features to the low-dimensional subspace composed of principal components through orthogonal transformations. The distribution of the downscaled samples in the two-dimensional and three-dimensional feature space is shown in Fig. 7, in which different categories of samples are distinguished by differentiated colors, and the intrinsic structure of the sample data can be effectively identified by observing the degree of aggregation and spatial distribution characteristics of the point set.

## 4. Data augmentation results comparison and analysis

### 4.1 Division of testing and training samples

In order to avoid the subjective bias of dataset division and ensure the balanced distribution of each category, the principle of random allocation was used to divide all the samples into the training set and the test set, and the samples are shown in Table 5, and the spatial distribution of the samples in the training set is visualized as shown in Fig. 8. At the same time, in order to avoid the chance of the experimental results, each classification model was carried out 10 times, and the average of the results obtained from ten experiments was taken to analyze and evaluate the experimental results.

### 4.2 Comparison and analysis of augmentation results

In this study, the generalization ability of models trained on the original training set, GAN-enhanced training set, SMOTE-enhanced training set, and ADASYN-enhanced training set was analyzed based on support vector machine, decision tree, and random forest classification algorithms using accuracy, precision, recall, and F1-score as evaluation metrics, and the diversity and realism of data generated by different data augmentation algorithms were compared and Validation of the diversity and realism of data generated by different data augmentation algorithms.

The structures of generator and discriminator in GAN are shown in Tables 6 and 7, respectively, and the hyperparameters are listed in Table 8.The experimental computer operating system in this paper is 64-bit Windows 10 Professional, the CPU model is Intel(R) Core(TM) i5-12400F, the RAM is 16GB, the GPU is NVIDIA GeForce RTX 4060 with 8GB of video memory, the integrated compiler software is Pycharm Community Edition 2023.2.1, the CUDA version is 12.0, the deep learning framework is Pytorch 2.1, and Python 3.10.

**Table 5:** The number of training set and test set samples.

| Sample Label | Total Number of Samples | Number of Training Set Samples | Number of Testing Set Samples |
|---|---|---|---|
| Single-cylinder misfire | 16 | 10 | 6 |
| Piston ring wear | 16 | 10 | 6 |
| Nozzle blockage | 16 | 10 | 6 |
| Exhaust valve leakage | 16 | 10 | 6 |
| Normal | 112 | 75 | 37 |

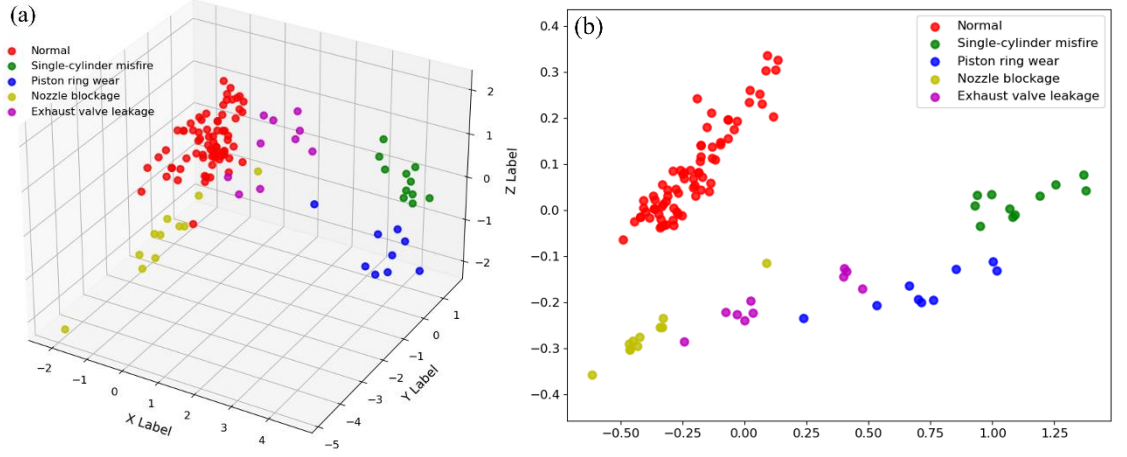



**Fig. 8:** The sample space distribution map of the original training set, (a) 3D spatial distribution diagram and (b) 2D spatial distribution diagram.

**Table 6:** Structure parameters of generator ($G$).

| Network Layer | Parameters |
|---|---|
| Input Layer | 10 |
| Hidden Layer | Linear (128) |
| Activation Function Layer | ReLU |
| Output Layer | 6 |

**Table 7:** Structure parameters of discriminator ($D$).

| Network Layer | Parameters |
|---|---|
| Input Layer | 6 |
| Hidden Layer | Linear (128) |
| Activation Function Layer | ReLU |
| Hidden Layer | Linear (1) |
| Activation Function Layer | Sigmoid |
| Output Layer | 1 |

**Table 8:** GAN hyperparameter setting.

| Hyperparameters | Values |
|---|---|
| Number of Iterations | 20000 |
| Learning Rate of G | 0.0001 |
| Learning Rate of D | 0.0001 |
| Adam Parameter $\beta_1$ | 0.9 |
| Adam Parameter $\beta_2$ | 0.999 |

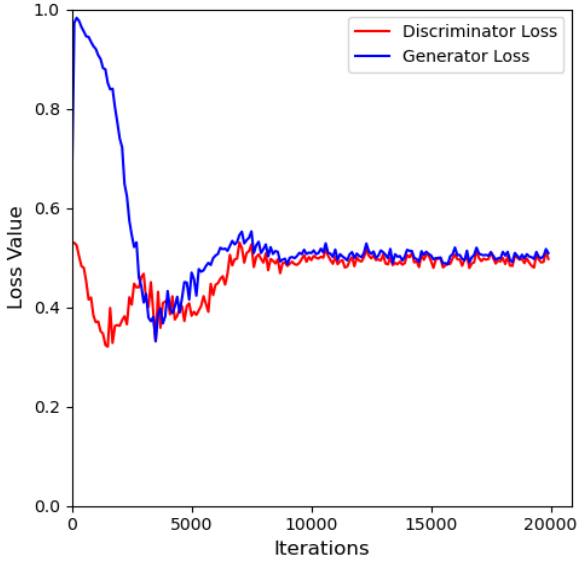



**Fig. 9:** Generator and discriminator loss curves.

In the iterative process of GAN, the generator and discriminator loss values change as shown in Fig. 9. As illustrated in Fig. 10, during the initial phase of GAN training, the curves for both the generator loss and discriminator loss exhibit significant fluctuations. This indicates that the generator is in the process of learning the distribution of real samples. At this point, the discriminator demonstrates strong performance, while the generator's ability to produce realistic outputs is still developing. Consequently, the discriminator's loss value is lower, whereas the generator's loss value remains higher. When the number of iterations reaches 10000 times, the generator loss and discriminator loss enter a relatively stable state. In the iterative process, with the increase in the number of iterations, the discriminator and generator antagonistic game, the discriminator loss rises, the generator loss declines, and ultimately realize the balance of the game, and the stability value of the two is also stabilized at about 0.5.

After GAN augmentation, the number of various types of samples in the original training set is successfully expanded to 75. Subsequently, the data were downscaled and visualized by principal component analysis (PCA) to obtain the state space distribution of the samples, as shown in Figs. 10(a) and (b). As seen in Figs. 10(a) and (b), the distribution of the fault samples after GAN augmentation shows a significant expansion compared to the original training set that was previously enhanced. This indicates that the GAN successfully increases the diversity of fault samples in the process of data augmentation. Further observation of the PCA graph details reveals that the fault samples of the same class are distributed around the aggregation center, and the classification boundaries between different classes of fault samples are clear. This indicates that the fault samples generated by GAN are not random data without logic, but have some authenticity and rationality.

When using SMOTE to process the training set, the sampling rate is set to N=8, and the number of nearest neighbors is set to k=8. After balancing the minority and majority class samples in this way, the spatial distribution of samples in each class is obtained as shown in Figs. 10(c) and (d). As can be seen from Figs. 10(c) and (d), the range of sample distribution of the training set enhanced by SMOTE is indeed expanded compared with the original training set, which reflects that SMOTE plays a role in increasing the number of samples and enriching the sample distribution. However, the SMOTE-enhanced training set is relatively limited in the expansion of the boundary range of various types of sample distributions, which appears to be more conservative.

Specifically, the forged samples generated by SMOTE exhibit a uniform and dense distribution around the original sample. This distribution indicates that SMOTE achieves sample balance based on the feature space of the original samples, based on generating new samples in their neighboring regions. The samples generated in this way have high similarity in features with the original samples, which can alleviate the sample imbalance problem to a certain extent, but may not be as good as the GAN method in terms of expanding the diversity of the samples, which can generate samples with wider distributions and more diverse features.

When using ADASYN to enhance the training set, set the balance degree $\beta = 1$, the number of nearest neighbors $k = 8$, and the distribution of various types of samples after augmentation is shown in Figs. 10(e) and (f). As seen in Figs. 10(e) and (f), the distribution range of the training set

enhanced by ADASYN is closer to that of the training set enhanced by SMOTE, which means that the two show some similarity in the degree of sample space expansion.

However, the forgery samples generated by ADASYN have their unique distributional properties. Specifically, these forged samples are more densely distributed near the classification boundary, while they are relatively coefficient away from the classification boundary. This distribution characteristic is closely related to the working principle of ADASYN, which can adaptively generate new samples according to the distribution of the data. near the classification boundary, the model is more likely to misclassify the samples in this region because the distinction between the categories of the samples is more critical, so ADASYN generates more samples to help the model learn the differences between the categories better, and to enhance the model's classification ability in the boundary region. classification ability. In regions far from the classification boundary, the sample categories are relatively clear and less difficult for the model to classify, so the number of samples generated is relatively small. This targeted sample generation strategy helps to utilize the generated samples more efficiently and improve the model's performance in the key regions.

The above graphs of PCA results are intuitive but lack quantitative analysis and comparison. Therefore, in this paper, CS and PCC are used to assess the quality of generated

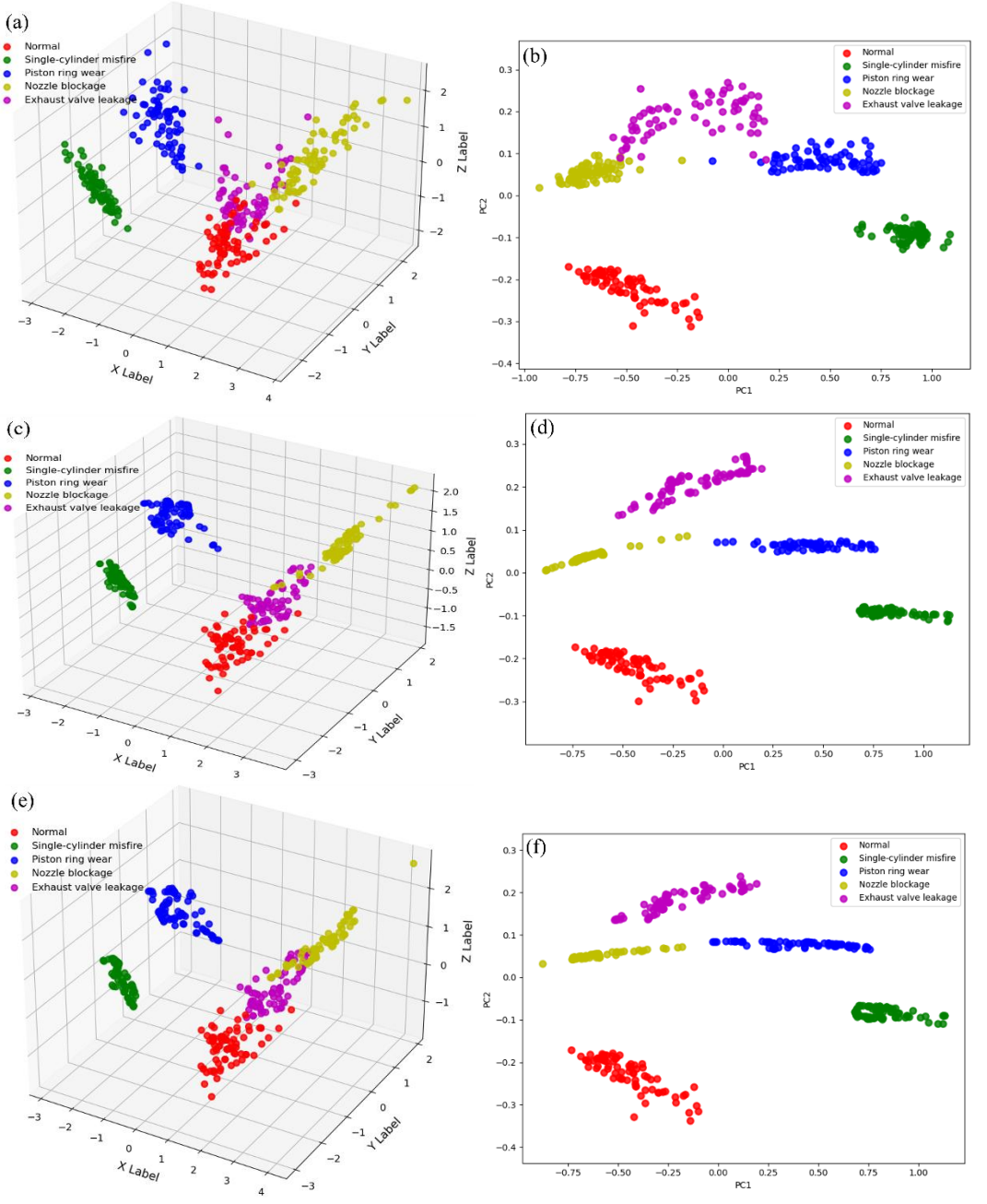


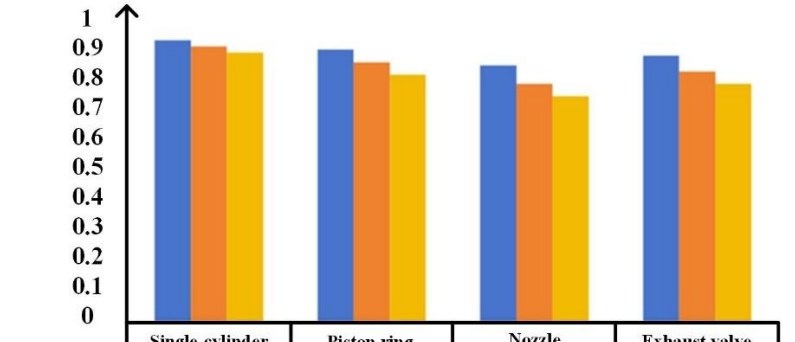

**Fig. 10:** The distribution of various types of samples after different augmentation methods. (a, b) The distribution of various types of samples after GAN Augmentation. (c,d) The distribution of various types of samples after SMOTE Augmentation. (e,f) The distribution of various types of samples after ADASYN Augmentation.

samples. The difference in data distribution between real samples and generated samples is evaluated and calculated, and the results are shown in Figs. 11 and 12, respectively. As can be seen in Figs. 11 and 12, the average values of GAN, ADASYN, and SMOTE for the CS of various types of faults are around 0.87, 0.83, and 0.79, respectively, and the average values of PCC are around 0.83, 0.78, and 0.74, respectively. It can be seen that the GAN method is better in both assessment metrics. In addition, in terms of objective criteria, values of PCC and CS greater than 0.5 indicate significant correlation, which shows that there is a highly similar data distribution between the GAN-generated samples and the real samples.

Subsequently, the data enhanced by each method were each used as a training set, and the original test set was used as a test sample to build the classification model based on Support Vector Machine (SVM), Decision Tree, and Random Forest classifiers on the GAN-enhanced training set, SMOTE-enhanced training set, and ADASYN-enhanced training set, respectively. Meanwhile, accuracy, precision, recall, and F1 score are used as model evaluation indexes. The experimental results of each classifier and test set are shown in Table 9, where the data are presented in the form of SVM/Decision Tree/Random Forest.

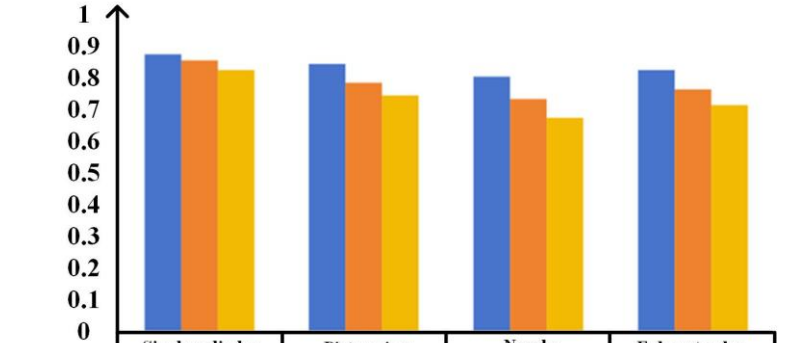
From the above results, it can be found that the training set processed by the three data augmentation algorithms can effectively help to improve the effect of the classifier, and all of them are better than the original training set in terms of classification accuracy. This indicates that for the unbalanced training set, the data augmentation operation can effectively improve the diversity level of the samples, which in turn provides the basis for the improvement of the model classification performance.

In an in-depth comparison of three data augmentation methods, GANs show the best performance. Specifically, GANs are able to significantly expand the boundaries of sample diversity while maintaining the authenticity characteristics of the samples. This dual advantage enables it to enhance the model training effect and thus the model classification performance, which is consistently demonstrated in the experiments with three different types of classifiers. Further analyzing its inner mechanism, GAN can deeply learn the potential distribution characteristics of the




| | Single-cylinder misfire | Piston ring wear | Nozzle blockage | Exhaust valve leakage |
|---|---|---|---|---|
| GAN | 0.91 | 0.88 | 0.83 | 0.86 |
| ADASYSN | 0.89 | 0.84 | 0.77 | 0.81 |
| SMOTE | 0.87 | 0.80 | 0.73 | 0.77 |

**Fig. 11:** Generated samples vs. real samples CS values.




| | Single-cylinder misfire | Piston ring wear | Nozzle blockage | Exhaust valve leakage |
|---|---|---|---|---|
| GAN | 0.87 | 0.84 | 0.80 | 0.82 |
| ADASYSN | 0.85 | 0.78 | 0.73 | 0.76 |
| SMOTE | 0.82 | 0.74 | 0.67 | 0.71 |

**Fig. 12:** Generated sample vs. real sample PCC values.

**Table 9:** SVM/Decision tree/Random forest experimental results.

| Dataset \ Evaluation indicators | Accuracy | Recall | Precision | F1-score |
|---|---|---|---|---|
| ADASYN-augmented | 0.933/0.921/0.925 | 0.865/0.855/0.913 | 0.874/0.865/0.876 | 0.858/0.860/0.897 |
| SMOTE-augmented | 0.925/0.900/0.880 | 0.833/0.798/0.850 | 0.857/0.843/0.800 | 0.826/0.774/0.824 |
| GAN-augmented | 0.950/0.943/0.932 | 0.900/0.876/0.920 | 0.938/0.923/0.910 | 0.897/0.864/0.915 |
| Original-augmented | 0.917/0.867/0.874 | 0.831/0.740/0.750 | 0.856/0.790/0.850 | 0.819/0.764/0.797 |

original data through the dynamic game process between the generator and the discriminator, so as to generate synthetic samples that not only conform to the intrinsic laws of the data, but also have a high degree of diversity, which provides rich and effective training samples for the model.

Based on the above experimental results, the data augmentation effect of the synthetic oversampling technique (SMOTE) is not as effective as the other two methods under the support vector machine (SVM) and random forest classifiers. In-depth exploration of the potential reasons can be found, SVM and random forest belong to the category of low-order models, when dealing with fault samples with complex nonlinear distribution, its feature space mapping ability is limited, and it is difficult to accurately capture the nonlinear decision boundaries between the samples, resulting in insufficient discriminative ability of the enhanced samples. At the same time, the SMOTE algorithm follows the principle of local linear interpolation to generate uniformly distributed synthetic samples in the neighborhood of the original training samples, a strategy that makes it pay insufficient attention to the key samples near the classification hyperplane, and fails to effectively improve the model's generalization ability in the boundary region, which in turn restricts the augmentation of classification performance.

The Adaptive Synthetic Sampling Algorithm (ADASYN) shows better augmentation effect compared to SMOTE, because it uses a density distribution estimation strategy to dynamically adjust the sampling density, and concentrates the generated fault samples near the classification boundary of the original training set. By enhancing the sample density in the boundary region, ADASYN can effectively alleviate the problem of sparse boundary samples and provide the classifier with more discriminative training samples. However, despite the breakthrough of ADASYN in optimizing the sample distribution, its augmentation effect is still inferior to that of Generative Adversarial Networks (GANs). This is mainly due to the fact that both SMOTE and ADASYN are essentially interpolation-based generative methods, and the distribution of their synthetic samples is strictly firstly bounded by the maximum distribution of the original training set, which makes it difficult to break through the limitations of the inherent data distribution. In contrast, GAN, through the adversarial training mechanism, is able to learn the high latitude potential distribution of the data, break through the original data boundary, and generate "truly unknown samples" outside the distribution boundary of the original training set, thus improving the diversity of samples.

## 5. Conclusion

In this paper, for the data imbalance problem in marine diesel engine fault diagnosis, the GAN algorithm based on data generation is systematically compared and analyzed with the SMOTE and ADASYN algorithms based on oversampling. When involving GAN networks, the gradient and learning rate of GAN are optimized by Adam gradient optimizer to improve the loss function convergence speed and model training efficiency. The evaluation model is constructed by combining Support Vector Machine (SVM), Decision Tree and Random Forest Classifier, and the diversity and authenticity of the augmented data are analyzed qualitatively and quantitatively by adopting Accuracy, Precision, Recall, F1 Score and Principal Component Analysis (PCA). The experimental results show that the GAN algorithm performs well on the unbalanced dataset of normal operating conditions, single cylinder misfire, piston ring wear, injector plugging and exhaust valve leakage collected from the cylinder head of Z6170ZICZ-1 marine diesel engine, which not only significantly improves the fault diagnosis performance of the three classifiers, but also effectively enhances the training set by breaking through the boundary of the distribution of the original data and generating new samples with real features diversity. In contrast, although SMOTE and ADASYN methods can improve the data imbalance situation to a certain extent, there are obvious limitations in improving the classification performance due to their interpolation-based sample generation mechanism, which is firstly based on the original data distribution range.

This study provides a theoretical basis and practical reference for the selection of data augmentation techniques in the field of marine diesel engine fault diagnosis, but this paper still suffers from limitations such as time-frequency domain feature dependence, fixed augmentation ratio assumption, and insufficient validation of external datasets. In the future, more in-depth research can be carried out on the exploration of multi-domain feature fusion data augmentation methods, dynamic augmentation strategy research, and cross-equipment

migration validation.

**Acknowledgments**
The authors appreciate the support from the School of Naval Architecture, Ocean, and Energy Power Engineering at Wuhan University of Technology. This research was funded by the High Technology Special Project of the Ministry of Industry and Information Technology (MIIT Joint Equipment Letter [2019] No. 120).

**Conflict of Interest**
There is no conflict of interest.

**Supporting Information**
Not applicable.

**CRediT Statement**
**Zhijun Chen**: Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition. **Ziyu Xiao**: Writing – Original draft, Methodology, Investigation, Formal analysis, Data curation. **Zhongjun Wang**: Writing – review & editing, Supervision, Software, Funding acquisition. **Pengcheng Wang**: Supervision, Resources. **Yijie Qu**: Supervision, Resources, Project administration. **Xiong Bao**: Supervision, Resources, Project administration. **Jie Wang**: Supervision, Resources.

**References**
[1] G. Zhong, H. Wang, K. Zhang, B. Jia, Fault diagnosis of Marine diesel engine based on deep belief network, *2019 Chinese Automation Congress (CAC)*, November 22-24, 2019, Hangzhou, China, IEEE, 2019, 3415-3419, doi: 10.1109/CAC48633.2019.8997060.
[2] X. M. Zhang, D. H. Chen, Study on fault diagnosis for turbocharging system of diesel engine based on support vector machine, *Applied Mechanics and Materials*, 2010, **42**, 371-374, doi: 10.4028/www.scientific.net/amm.42.371.
[3] C. Cai, C. Zhang, G. Liu, A novel fault diagnosis approach combining SVM with association rule mining for ship diesel engine, *2016 IEEE International Conference on Information and Automation (ICIA)*, August 1-3, 2016, Ningbo, China, IEEE, 2016, 130-135, doi: 10.1109/ICInfA.2016.7831809.
[4] K. Zhong, J. Li, J. Wang, M. Han, Fault detection for marine diesel engine using semi-supervised principal component analysis, *2019 9th International Conference on Information Science and Technology (ICIST)*, August 2-5, 2019, Hulunbuir, China, IEEE, 2019, 146-151, doi: 10.1109/ICIST.2019.8836805.
[5] Y. Wang, T. Cui, F. Zhang, T. Dong, S. Li, Fault diagnosis of diesel engine lubrication system based on PSO-SVM and centroid location algorithm, *2016 International Conference on Control, Automation and Information Sciences (ICCAIS)*, October 27-29, 2016, Ansan, Korea, IEEE, 2016, 221-226, doi: 10.1109/ICCAIS.2016.7822464.
[6] X. Wang, B. Han, Research on fault pattern analysis of marine diesel engine based on random forest algorithm, *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, August 8-10, 2017, Banff, AB, Canada, IEEE, 2017, 312-318, doi: 10.1109/ICTIS.2017.8047782.
[7] N. Rout, D. Mishra, M. K. Mallick, Handling imbalanced data: a survey, *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, Singapore: Springer Singapore, 2017, 431-443, doi: 10.1007/978-981-10-5272-9_39.
[8] L. Peng, B. Yang, Y. Chen, X. Zhou, An under-sampling imbalanced learning of data gravitation based classification, *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, August 13-15, 2016, Changsha, China, IEEE, 2016, 419-425, doi: 10.1109/FSKD.2016.7603210.
[9] X. Chao, G. Kou, Y. Peng, A. Fernández, An efficiency curve for evaluating imbalanced classifiers considering intrinsic data characteristics: Experimental analysis, *Information Sciences*, 2022, **608**, 1131-1156, doi: 10.1016/j.ins.2022.06.045.
[10] J. Xie, Z. Qiu, The effect of imbalanced data sets on LDA: a theoretical and empirical analysis, *Pattern Recognition*, 2007, **40**, 557-562, doi: 10.1016/j.patcog.2006.01.009.
[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, *Communications of the ACM*, 2020, **63**, 139-144, doi: 10.1145/3422622.
[12] J. Liu, F. Qu, X. Hong, H. Zhang, A small-sample wind turbine fault detection method with synthetic fault data using generative adversarial nets, *IEEE Transactions on Industrial Informatics*, 2019, **15**, 3877-3888, doi: 10.1109/TII.2018.2885365.
[13] Y. Gao, X. Liu, H. Huang, J. Xiang, A hybrid of FEM simulations and generative adversarial networks to classify faults in rotor-bearing systems, *ISA Transactions*, 2021, **108**, 356-366, doi: 10.1016/j.isatra.2020.08.012.
[14] P. Liang, C. Deng, J. Wu, Z. Yang, Intelligent fault diagnosis of rotating machinery *via* wavelet transform, generative adversarial nets and convolutional neural network, *Measurement*, 2020, **159**, 107768, doi: 10.1016/j.measurement.2020.107768.
[15] W. Luo, W. Yang, J. He, H. Huang, H. Chi, J. Wu, Y. Shen, Fault diagnosis method based on two-stage GAN for data imbalance, *IEEE Sensors Journal*, 2022, **22**, 21961-21973, doi: 10.1109/JSEN.2022.3211021.
[16] Z. Dai, L. Zhao, K. Wang, Y. Zhou, Generative adversarial network to alleviate information insufficiency in intelligent fault diagnosis by generating continuations of signals, *Applied Soft Computing*, 2023, **147**, 110784, doi: 10.1016/j.asoc.2023.110784.
[17] B. Zhao, Q. Yuan, Improved generative adversarial network for vibration-based fault diagnosis with imbalanced data, *Measurement*, 2021, **169**, 108522, doi: 10.1016/j.measurement.2020.108522.
[18] D. Cui, S. Liu, X. Xu, P. Lin, G. Hu, Rapid prediction of pollutant dispersion in residential blocks using generative adversarial networks, *Urban Climate*, 2025, **62**, 102533, doi: 10.1016/j.uclim.2025.102533.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, 2002, **16**, 321-357, doi: 10.1613/jair.953.
[20] S. Rao, G. Zou, S. Yang, S. A. Khan, Fault diagnosis of power transformers using ANN and SMOTE algorithm, *International Journal of Applied Electromagnetics and Mechanics*, 2022, **70**, 345-355, doi: 10.3233/jae-210227.
[21] T. Z. Wei Chang, S. C. Tan, K. S. Sim, C. P. Lim, P. Y. Goh, Application of over-sampling techniques and fuzzy ARTMAP to condition monitoring of a power generation system, *2023 IEEE 3rd International Conference in Power Engineering Applications (ICPEA)*, March 6-7, 2023, Putrajaya, Malaysia, IEEE, 2023, 233-238, doi: 10.1109/ICPEA56918.2023.10093214.
[22] Y. Fan, X. Cui, H. Han, H. Lu, Chiller fault diagnosis with field sensors using the technology of imbalanced data, *Applied Thermal Engineering*, 2019, **159**, 113933, doi: 10.1016/j.applthermaleng.2019.113933.
[23] D. Liu, S. Zhong, L. Lin, M. Zhao, X. Fu, X. Liu, Feature-level SMOTE: Augmenting fault samples in learnable feature space for imbalanced fault diagnosis of gas turbines, *Expert Systems with Applications*, 2024, **238**, 122023, doi: 10.1016/j.eswa.2023.122023.
[24] Z. Li, Y. He, Z. Xing, J. Duan, Transformer fault diagnosis based on improved deep coupled dense convolutional neural network, *Electric Power Systems Research*, 2022, **209**, 107969, doi: 10.1016/j.epsr.2022.107969.
[25] H. Cao, X. Xie, J. Shi, Y. Wang, Evaluating the validity of class balancing algorithms-based machine learning models for geogenic contaminated groundwaters prediction, *Journal of Hydrology*, 2022, **610**, 127933, doi: 10.1016/j.jhydrol.2022.127933.
[26] W. Yin, H. Xia, X. Huang, Z. Wang, A fault diagnosis method for nuclear power plants rotating machinery based on deep learning under imbalanced samples, *Annals of Nuclear Energy*, 2024, **199**, 110340, doi: 10.1016/j.anucene.2024.110340.
[27] H. Xu, A. Qadir, S. Sadiq, Malicious SMS detection using ensemble learning and SMOTE to improve mobile cybersecurity, *Computers & Security*, 2025, **154**, 104443, doi: 10.1016/j.cose.2025.104443.
[28] P. K. Diederik, Adam: A method for stochastic optimization, *International Conference on Learning Representations*, 2014, 1-13, doi: 10.48550/arXiv.1412.6980.
[29] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, 1995, **20**, 273-297, doi: 10.1007/BF00994018.
[30] J. R. Quinlan, C4. 5 programs for machine learning, Elsevier, 2014, doi: 10.1007/BF00993309.
[31] L. Breiman, Random forests, *Machine Learning*, 2001, **45**, 5-32, doi: 10.1023/A:1010933404324.
[32] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, *Information Processing & Management*, 2009, **45**, 427-437, doi: 10.1016/j.ipm.2009.03.002.
[33] H. M. Nahim, R. Younes, H. Shraim, M. Ouladsine, Oriented review to potential simulator for faults modeling in diesel engine, *Journal of Marine Science and Technology*, 2016, **21**, 533-551, doi: 10.1007/s00773-015-0358-6.
[34] M. Turk, A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, 1991, **3**, 71-86, doi: 10.1162/jocn.1991.3.1.71.