Health evaluation methodology of remote maintenance control system of natural gas pipeline based on ACWGAN-GP algorithm

Shibin Zhang*, Fei Song, Xiaojun Zhang, Lidong Jia, Wei Shi, Yueqiao Ai and Weichun Hei
National Petroleum Network Group North Pipeline Co., Ltd, 408 Xinkai Road, Langfang City, Hebei Province
*zhangsb@pipechina.com.cn

Abstract. The remote maintenance system of natural gas pipeline is of great significance to ensure the safe and stable operation of the pipeline network. The multi-classification method based on machine learning is more effective for the health evaluation of remote maintenance control system than the traditional evaluation method based on expert experience. In view of the severe imbalance in the number of samples of five health levels, a health evaluation methodology of remote maintenance control system based on Wasserstein distance sand auxiliary classification generative adversarial network (ACWGAN-GP) is proposed. Firstly, the model stability is improved by introducing Wasserstein distance and gradient penalty. The generator generates balanced data, while the discriminator trains with generated and actual data. In this way, several ACWGAN-GP sub-models are trained. Then, the health levels of the sub-model are directly obtained by using the discriminator to classify the samples. Finally, according to the hierarchical relationship of the system, a parallel-serial combined evaluation method is adopted. By this means, the health evaluation model of remote maintenance control system including ACWGAN-GP sub-models is constructed. The experimental results based on 13 sets of KEEL and UCI multi-class imbalanced datasets and actual sampling data show that the effectiveness and advancement of the proposed method improved significantly compared with the existing similar typical algorithms.

1. Introduction
The remote maintenance control system (RMCS) of natural gas pipeline improves the reliability of the operation of pipeline SCADA system and ensures the safe and stable operation of the pipeline network[1]. To ensure the efficient operation of RMCS, the health evaluation of RMCS becomes particularly important. Nowadays, the health evaluation of the system mainly adopts methods based on expert experience [2]. However, these methods lack objectivity when determining index weight and other vital factors. In recent years, machine learning methods have been used to evaluate target systems[3]. The health level of RMCS can be divided into five classes, and sample data for each group is a severe imbalance. Therefore, the health evaluation of RMCS can be regarded as a multi-classification problem of imbalanced data in machine learning.

Algorithm-level methods are often targeted at specific problems or specific data distributions, which are generally not universal[4]. Data-level methods are more general, including oversampling, undersampling, and hybrid sampling. When the data is highly imbalanced, the use of undersampling and hybrid sampling will lose the information of a number of majorities, oversampling thus is widely used.
A series of methods inspired by the Synthetic Minority Oversampling Technique (SMOTE) is proposed to solve the overgeneralization of SMOTE, while there is no mechanism to guarantee the authenticity of the sampled data.

Generative adversarial network (GAN) based on adversarial learning of the generator and the discriminator provides new ideas for solving the above problems. Douzas et al. used Conditional GAN (CGAN) to capture the overall distribution of data by adding additional conditional information to GAN and obtain a balanced dataset. Zheng et al. combined WGAN-GP with conditional information to alleviate model collapse caused by imbalanced data. Odena et al. proposed Auxiliary Classifier GAN (ACGAN), which can generate high-resolution images and distinguish image classes by improving the original GAN’s discriminator. However, the above GAN-based methods still have these problems: (1) The randomness of the generated data will also lead to the randomness of the results of the classifier. (2) Multi-classification problem is often transformed into binary classification, resulting in increased imbalance ratio, unclear classification boundaries, information loss from a single classifier and lower efficiency.

In this paper, we propose a health evaluation model of RMCS of natural gas pipeline based on Wasserstein distance and auxiliary classification generative adversarial network algorithm (ACWGAN-GP). A robust classifier is obtained in the training process by training the discriminator with actual data and multifarious balance-generated data. At the same time, ACWGAN-GP uses the discriminator to classify the test data directly, which ameliorates the problems when converting into binary classification. Finally, due to the introduction of Wasserstein distance and gradient penalty, the proposed method alleviates the problem that ACGAN is prone to mode collapse.

2. ACWGAN-GP algorithms

The structure of ACWGAN-GP is shown in Fig. 1. The generator generates samples based on noise and specified class labels. The discriminator distinguishes a sample whether it originated from the actual data distribution or the generative data distribution. During the training process, balance data generated by the generator is used to train the auxiliary classifier to avoid the decision boundary deviation. Meanwhile, generated data can increase the diversity of training data used to train the auxiliary classifier, which can improve the robustness of the classifier. Section 1.1 describes the design of the model loss function, and Section 1.2 introduces the model training process.

![Fig. 1 Structure of ACWGAN-GP.](image)

2.1. Loss function design

(1) Adversarial Loss

During the training process, we have to ensure the generator can generate data as authentic as possible, and also the discriminator can successfully distinguish the data sources. The JS divergence used in vanilla GAN is not suitable for measuring the distance between the generative data distribution and the actual data distribution. Instead, Wasserstein distance is adopted to measure the distance, theoretically solving training instability. However, the use of the Wasserstein distance requires Lipschitz continuity.
We introduce gradient penalty to the discriminator to satisfy this condition as well as to avoid gradient explosion or gradient disappearance. We adopt the following adversarial loss

\[
L_{adv} = \min_{\theta_D} \max_{\theta_G} L(D, G) = E_{x \sim p_{\text{data}}(x)} [D(x)] - E_{z \sim p_{\text{noise}}(z)} [D(G(z, c))] - \lambda E_{\tilde{x} \sim p_{\text{data}}(x)} [(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2],
\]

(1)

where \(x\) is sampled from the data distribution \(p_{\text{data}}(x)\), \(z\) is sampled from the noise distribution \(p_{\text{noise}}(z)\), \(\tilde{x}\) is sampled along line segment that joins the real data and generated data, \(\tilde{x} = a \ast x + (1 - a) \ast G(z, c)\), \(\lambda\) is the gradient penalty coefficient. The generator \(G\) generates data \(G(z, c)\) based on the input data \(z\) and labels \(c\), while \(D(x)\) and \(D(G(z, c))\) represents the probability that the sample is sampled from the data distribution. The larger the \(D(x)\), the more likely the discriminator considers \(x\) to be real. The generated data is increasingly consistent with the actual data distribution with alternating training and confrontation between the discriminator and the generator.

(2) Classification Loss

Another loss function of the discriminator is the loss of the auxiliary classifier. The discriminator realizes the simultaneous training of multi-class data by predicting the class of samples, making the training more stable, and realizing the direct classification using the auxiliary classifier\[^{12}\]. We adopt the training loss of the auxiliary classifier as

\[
L_{\text{class}} = E_{x \sim p_{\text{data}}(x)} [\log P(c = c_{\text{true}}|x)] + E_{z \sim p_{\text{noise}}(z)} [\log P(c = c_{\text{gen}}|G(z, c_{\text{true}}))],
\]

(2)

where \(c\) is the class obtained by the discriminator, \(c_{\text{true}}\) and \(c_{\text{gen}}\) are the categories of real data and generated data, respectively. \(\log P(\cdot)\) represents the probability that the discriminator \(D\) determines the class of data is \(c_{\text{true}}\) and \(c_{\text{gen}}\). The larger the \(L_{\text{class}}\), the more accurately the auxiliary classifier predicts the data class. Thus, the auxiliary classifier is trained by maximizing \(L_{\text{class}}\).

(3) Overall Loss

The overall loss of the proposed method is

\[
L = L_{adv} + L_{\text{class}}
\]

(3)

According to equation (1) and equation (2), the generator loss \(L_G\) and the discriminator loss \(L_D\) can be obtained. The discriminator can distinguish data sources and data classes as much as possible by minimizing \(L_D\), and the generator can generate data that is as consistent as possible with the actual distribution by minimizing \(L_G\). The loss functions of the generator and the discriminator are respectively

\[
L_D = -L_{adv} - L_{\text{class}}
\]

(4)

\[
L_G = -E_{z \sim p_{\text{noise}}(z)} [D(G(z, c))] - E_{z \sim p_{\text{noise}}(z)} [\log P(c = c_{\text{gen}}|G(z, c_{\text{true}}))]
\]

(5)

2.2. Training process

While training, the training procedure of one epoch consists of alternating \(n_{\text{critic}}\) optimizing steps for \(D\) and one optimizing step for \(G\). The model parameters are updated iteratively based on equation (4) and equation (5). The training process can be roughly divided into four steps.

Step 1: Sample noise samples \(z\) with specified class-balanced labels \(c\) from noise distribution \(p_{\text{noise}}(z)\) and class-balanced real samples \(x\) from real data distribution \(p_{\text{data}}(x)\). The generator \(G\) generates data \(\tilde{x}\) based on the input samples \(z\) and labels \(c\).

Step 2: The discriminator obtains the classes of the received samples and the probabilities that received samples are sampled from the actual data distribution. The discriminator parameters are updated based on the received sampled and equation (4). If the number of the discriminator training is less than \(n_{\text{critic}}\), keep training the discriminator.

Step 3: While training the generator, the parameters of the discriminator remain frozen. The parameters of the generator are updated based on equation (5) and the generated samples \(\tilde{x}\) that generated from the generator with the noise samples \(z\) and specified labels \(c\).

Step 4: After training, the discriminator trained by diverse balanced data is used as a classifier to classify the test data directly.

The detailed procedure of ACWGAN-GP is presented in Algorithm 1.
Algorithm 1 ACWGAN-GP

**Input:** number of trainings \(\text{epoch}\); batch size \(m\); \(n_{\text{critic}}\); real data distribution \(p_r(x)\); noise distribution \(p_z(x)\); initial discriminator parameters \(\omega_0\); initial generate parameters \(\theta_0\); gradient penalty coefficient \(\lambda\); Adam hyperparameters \(\alpha, \beta_1, \beta_2\)

**Output:** ACWGAN-GP classification model

1. for \(i = 1, \ldots, \text{epoch}\) do:
   2. for \(j = 1, \ldots, n_{\text{critic}}\) do:
   3. for \(n = 1, \ldots, m\) do:
   4. \(\bar{x} = G(z, c_g)\)
   5. \(\bar{x} = a \ast x + (1 - a) \ast \bar{x}\)
   6. \(c_r \leftarrow D_\omega(x), \; c_g \leftarrow D_\omega(\bar{x})\)
   7. \(L^D_\omega(n) = D_\omega(\bar{x}) - D_\omega(x) + \lambda(\|\nabla_x D_\omega(\bar{x})\|_2^{-1})^2 - \log P(c = c_r | x) - \log P(c = c_g | \bar{x})\)
   8. end for
   9. \(\omega \leftarrow \text{Adam}\left(\nabla_\omega \frac{1}{m} \sum_{i=1}^{m} L^D_\omega(n), \omega, \alpha, \beta_1, \beta_2\right)\)
   10. end for
   11. for \(n = 1, \ldots, m\) do:
   12. \(\bar{x} = G(z, c_g)\)
   13. \(c_g \leftarrow D_\omega(\bar{x})\)
   14. \(L^G_\omega(n) = -D_\omega(\bar{x}) - \log P(c = c_g | \bar{x})\)
   15. \(\theta \leftarrow \text{Adam}\left(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} L^G_\omega(n), \theta, \alpha, \beta_1, \beta_2\right)\)
   16. end for

3. Health evaluation model of remote maintenance control system of natural gas pipeline

3.1. Overall health evaluation model of remote maintenance control system of natural gas pipeline

The RMCS has numerous hardware devices, including servers, routers, switches, workstations, PLC, etc. Each kind of device realizes specific functions for a submodule separately and cooperates to ensure the overall health of RMCS. The health of the lowest layer submodule is affected by hardware, while the health of the superior layer submodule is affected by the health levels of lower layer submodules. The health levels of all submodules reflect the overall health level of the system. Therefore, this paper proposes a health evaluation model of RMCS based on the ACWGAN-GP algorithm. Considering the different kinds of resources occupied by different hardware systems, the health evaluation ACWGAN-GP sub-model is separately trained for each hardware submodule. With the health evaluation sub-models are obtained, the sub-models of the superior layer are obtained according to the hierarchical relationship of the system health level. Finally, the overall health evaluation ACWGAN-GP model of the system is obtained according to all health sub-models.

The health evaluation model of RMCS includes five health evaluation sub-models and one health evaluation overall model. Firstly, five health evaluation sub-models of server health, router health, switch health, workstation health, and PLC health are obtained through the ACWGAN-GP algorithm, respectively. The input features of each health evaluation sub-model are their resource occupations, while the output is health level. Then, the health levels of the five sub-modules are discretized as the input characteristics of ACWGAN-GP to gain the health evaluation overall model. Finally, with parallel training of sub-models and serial training of sub-models and overall model, the health evaluation model of RMCS constituted by sub-models and overall model is obtained. (See Fig. 2.)
3.2. Health evaluation sub-model of remote maintenance control system of natural gas pipeline based on ACWGAN-GP

The health evaluation model of RMCS includes six ACWGAN-GP models. The following content takes the server health evaluation sub-model as an example. The data which consists of the server resource occupations and server health levels, is collected as the dataset. The server resource occupations are taken as the input features of the model, and the server health levels are taken as the labels.

Step1: The resource usage of servers is used as the input features of the server evaluation health ACWGAN-GP sub-model, including eight input features: system uptime, CPU Single-core usage, memory usage, power state, fan status, fan speed, disk usage, and disk capacity.

Step2: Power state and fan state in the server resource usage feature are discretized to 0 and 1; The health level is divided into five groups: intact, regular, attention, abnormal, and fault, which is discretized into 0, 1, 2, 3, and 4 in server health evaluation ACWGAN-GP sub-model.

Step3: The original dataset is divided into training set and test set, of which 80% is training set, and the rest 20% is test set.

Step4: The balanced data is sampled from the original dataset of the five levels, respectively. The generated data is obtained by using the method for generating data described in Section 2.2. The discriminator is trained by actual data and generated data, and the discriminator parameters are updated according to equation (4). If the number of the discriminator training is less than $n_{\text{critic}}$, keep training the discriminator.

Step5: The discriminator parameters remain frozen. The generator parameters are updated with generated data according to equation (5).

The server health evaluation ACWGAN-GP sub-model is obtained through several iterations of steps4 and step5. The detailed procedure of training is shown in Fig. 3.
4. Experiments

4.1. Experimental design and performance evaluation metrics

The imbalance ratio $\gamma$ is the degree of imbalance in a dataset, $N_{\text{max}}$ and $N_{\text{min}}$ are the maximum and minimum sample sizes of a single class, respectively. The imbalance ratio is defined as follows:

$$\gamma = \frac{N_{\text{min}}}{N_{\text{max}}}$$  \hspace{1cm} (6)

The average accuracy $\tau_{\text{macroacc}}$ is adopted as the evaluation metric, assigning the same weight to different classes. The final classification accuracy is calculated by averaging the accuracy of every single class. Compared with the overall accuracy, it can more effectively evaluate the classification performance of the classifier on multi-class imbalanced datasets$^{[3]}$. $\tau_{\text{macroacc}}$ is defined as follows

$$\tau_{\text{macroacc}} = \frac{1}{N_d} \sum_{i=1}^{N_d} \rho_{i,\text{recall}}$$  \hspace{1cm} (7)

where $N_d$ is the number of classes in the dataset, and $\rho_{i,\text{recall}}$ is the proportion of the number of samples correctly classified in the $i$-th class to the total number of samples in this class.

All methods are performed five-fold cross-validation on each dataset ten times to get the average results. With the sample size, number of attributes (#Attr), imbalance ratio (#IR), number of classes (#Cl), and other dataset attributes considered, to compare the proposed method with other ones fully, we adopt the Friedman Test$^{[8]}$ and Wilcoxon Signed Rank Test$^{[3]}$ to further analyse the results.

4.2. Datasets

To verify the effectiveness of the proposed algorithm, 13 groups of representative multi-class imbalanced datasets selected from the authoritative machine learning datasets KEEL and UCI are used for comparative experiments. The descriptions of the datasets are shown in table 1.

| Dataset       | #Attr | Instances | #Cl | Class distribution          | #IR  |
|---------------|-------|-----------|-----|----------------------------|------|
| vehicle       | 18    | 846       | 4   | 218,217,212,199             | 1.10 |
| wine          | 13    | 178       | 3   | 71,59,48                   | 1.48 |
| led7digit     | 7     | 500       | 10  | 57,57,53,52,52,51,49,45,47,37 | 1.54 |
| newthyroid    | 5     | 215       | 3   | 150,35,30                  | 5.00 |
| dermatology   | 34    | 358       | 6   | 111,60,71,48,48,20         | 5.55 |
| balance       | 4     | 625       | 3   | 288,288,49                 | 5.88 |
| flare         | 11    | 1066      | 6   | 331,239,211,147,95,43      | 7.70 |
| glass         | 9     | 214       | 6   | 76,70,29,17,139            | 8.44 |
| cleveland     | 13    | 297       | 5   | 160,54,35,35,13            | 12.31|
| car           | 6     | 1728      | 4   | 1210,384,69,65             | 18.62|
| winequality-red| 11  | 1599      | 6   | 681,638,199,18,10          | 68.10|
| page-blocks   | 10    | 5472      | 5   | 4913,329,115,87,28         | 175.46|
| shuttle       | 9     | 58000     | 7   | 45586,8903,3267,171,49,13,10 | 4558.60|
The actual server operation data of remote maintenance control system of natural gas pipeline are used as the original dataset, with a total of 10,000 samples. The data of the servers contain eight input features, which are divided into five classes. The imbalance ratio reaches 21.38, which is a typical imbalanced multi-classification problem. Detailed descriptions of the dataset are described in table 2.

Table 2. Input attribute and output category description of server data of remote maintenance control system.

| Input attribute       | Category | Health level |
|-----------------------|----------|--------------|
| System uptime         | 0        | Intact       |
| CPU Single-core usage | 1        | Normal       |
| Memory usage          | 2        | Attention    |
| Power state           | 3        | Abnormal     |
| Fan state             | 4        | Fault        |
| Fan speed             | -        | -            |
| Disk usage            | -        | -            |
| Disk capacity         | -        | -            |

4.3. Parameter settings

Typical machine learning approaches SMOTE, Borderline-SMOTE, KmeansSMOTE, SOMO are used as control methods. Additionally, we include the typical ensemble learning method DES-MI and the performance of the classifiers when no oversampling is used. The implementation of traditional machine learning oversampling methods is based on Python library Scikit-Learn and Imbalanced-Learn with default settings employed.

As to generation approaches based on GAN, representative approaches GAN, CGAN, CWGAN-GP are used as control methods. With respect to the approaches based on GAN, most hyperparameters are the same, except for the loss function, output layer design, and ratio of training times of the generator and the discriminator. For the same hyperparameters, all four methods are five-layer neural networks, Dropout and Batch Normalization are applied to either the generator or the discriminator in each model. Both the generator and the discriminator used LeakyReLU with the parameter set to 0.2 as the activation function for the hidden layers and input layer. The numbers of hidden units of generator and discriminator are respectively set to 32-64-128 and 64-128-256. The dimension of noise space is set to range from 20 to 100, the batch size is set to 10, and the number of epochs is 20000. The four models are trained using Adam optimizer, the values of $\alpha$, $\beta_1$, and $\beta_2$ of the Adam optimizer are set to (0.001 for generator, 0.003 for discriminator), 0, and 0.9, respectively. From the perspective of different hyperparameters, the loss functions of GAN$^7$, CGAN$^9$, and CWGAN-GP$^{10}$ are consistent with those originally reported. The output layer of the discriminator in GAN and CGAN uses Sigmoid activation, but the activation function is applied to the output layer in neither CWGAN-GP nor ACWGAN-GP. The gradient penalty coefficient $\lambda$ in CWGAN-GP and ACWGAN-GP is set to 10. One discriminator is updated followed by the single generator in GAN and CGAN, while five discriminators are updated in CWGAN-GP and ACWGAN-GP. The generation approaches based on GAN are implemented in Python using Pytorch. Random Forest (RF) is used as classification algorithms to evaluate the above approaches, implemented based on the Python library Scikit-Learn.

4.4. Experimental results and discussion

4.4.1. Public Dataset.

The experimental results of the proposed ACWGAN-GP in this paper are compared with those of the other nine algorithms. (See Table3.) As observed in table 3, ACWGAN-GP performs better on the $P_{macroacc}$ metric, achieving the best performance on 9 of the 13 public datasets. Bold values indicate the best-performing approach for each public dataset. ACWGAN-GP also achieves the best average accuracy on 13 datasets, with a 5.52% improvement over the best of the other methods.
The Wilcoxon signed-rank test and Friedman test are used to compare the different methods. The results of the Wilcoxon signed-rank test are shown in table 4, where the value of $R^+$ represents the rank sum of the ACWGAN-GP method, and the value of $R^-$ represents the rank sum of the other nine methods. It can be shown in table 4 that assumptions of all methods except CWGAN-GP on $\tau_{macroacc}$ are rejected, indicating that ACWGAN-GP is significantly different from other methods except CWGAN-GP. $R^+$ is greater than $R^-$, showing that ACWGAN-GP has a better performance compared with the comparison method. Friedman test results are shown in Table 3. ACWGAN-GP has the lowest average ranking of 2.00 in the Friedman test. Experimental results show that the proposed method performs better than other comparison methods on imbalanced datasets.

### Table 3. Results and mean ranking score of ten approaches.

| Datasets   | RF | SMOTE +RF | Borderline SMOTE +RF | Kmeans-SMOTE +RF | SOMO +RF | DES-MI +RF | GAN +RF | CGAN +RF | CWGAN+RF | ACWGAN-GP |
|------------|----|-----------|----------------------|------------------|---------|------------|--------|---------|---------|-----------|
| balance    | 0.6053 | 0.5862 | 0.5907 | 0.5862 | 0.6053 | 0.5683 | 0.5925 | 0.5949 | 0.5972 | 0.9363 |
| car        | 0.9497 | 0.9614 | 0.9381 | 0.9613 | 0.9463 | 0.9551 | 0.9545 | 0.9544 | 0.9544 | 0.9634 |
| cleveland  | 0.3648 | 0.3257 | 0.3614 | 0.3377 | 0.3648 | 0.3481 | 0.3224 | 0.3649 | 0.3649 | 0.3688 |
| dermatology| 0.9750 | 0.9720 | 0.9779 | 0.9739 | 0.9663 | 0.9678 | 0.9811 | 0.9811 | 0.9783 | 0.9818 |
| flare      | 0.5899 | 0.6058 | 0.6121 | 0.6105 | 0.5899 | 0.6299 | 0.5938 | 0.5923 | 0.5950 | 0.6574 |
| glass      | 0.7216 | 0.8041 | 0.7968 | 0.7951 | 0.7151 | 0.7432 | 0.7570 | 0.7670 | 0.7697 | 0.7167 |
| led7digit  | 0.7192 | 0.7175 | 0.7170 | 0.7075 | 0.7034 | 0.7067 | 0.7254 | 0.7214 | 0.7196 | 0.7266 |
| page-blocks| 0.8542 | 0.8880 | 0.8832 | 0.8812 | 0.8391 | 0.8570 | 0.8411 | 0.8966 | 0.8897 | 0.9456 |
| shuttle    | 0.9284 | 0.9597 | **0.9880** | 0.9825 | 0.9284 | 0.9351 | 0.8803 | 0.9379 | 0.9379 | 0.9791 |
| vehicle    | 0.7481 | 0.7545 | 0.7538 | 0.7575 | 0.7481 | 0.7425 | 0.7551 | 0.7586 | 0.7599 | 0.8562 |
| wine       | 0.9694 | 0.9698 | 0.9694 | 0.9646 | 0.9694 | 0.9598 | 0.9646 | 0.9813 | 0.9709 | 0.9849 |
| wine_red   | 0.3351 | 0.4024 | 0.3820 | 0.3985 | 0.3351 | 0.3402 | 0.3976 | 0.4595 | 0.4333 | 0.4208 |
| newthyroid | 0.9454 | 0.9491 | 0.9165 | 0.9406 | 0.9404 | 0.9644 | 0.9476 | 0.9676 | 0.9676 | 0.9911 |

**$\tau_{macroacc}$**

| Mean ranking (p=6.1e^-7) |
|--------------------------|
| 0.70 | 5.12 | 5.85 | 5.77 | 8.04 | 7.42 | 6.38 | 3.77 | 3.65 | **2.00** |

Table 4. Results of the Wilcoxon Signed Rank Test when ACWGAN-GP is used as the control method. ($\alpha=0.05$)

| Comparison                  | $R^+$ | $R^-$ | p-value | Assuming($\alpha=0.05$) |
|-----------------------------|-------|-------|---------|-------------------------|
| ACWGAN-GP vs None           | 89    | 2     | 0.002366 | reject                  |
| ACWGAN-GP vs SMOTE          | 80    | 11    | 0.015906 | reject                  |
| ACWGAN-GP vs Borderline-SMOTE| 77    | 14    | 0.027708 | reject                  |
| ACWGAN-GP vs Kmeans-SMOTE   | 78    | 13    | 0.002313 | reject                  |
| ACWGAN-GP vs GAN            | 85    | 6     | 0.005772 | reject                  |
| ACWGAN-GP vs CGAN           | 74    | 17    | 0.046399 | reject                  |
| ACWGAN-GP vs CWGAN-GP       | 71    | 20    | 0.074735 | not reject              |
| ACWGAN-GP vs SOMO           | 85    | 6     | 0.005772 | reject                  |
| ACWGAN-GP vs DES-MI         | 91    | 0     | 0.001474 | reject                  |

4.4.2. Actual data of remote maintenance control system.

The average classification results obtained by five-fold cross validation repeated ten times of the actual server operation data of the remote maintenance control system of natural gas pipelines is shown in table 5. It can be seen from table 5 that ACWGAN-GP achieves the best performance on $\tau_{macroacc}$ and improves 2.43% compared with the best method among other methods.
5. Conclusion

To conduct health evaluation of the remote maintenance control system of natural gas pipeline, considering the class-imbalanced data of each health level of the system, we proposed a health evaluation model based on ACWGAN-GP algorithm in this paper. Performance of SMOTE, Borderline-SMOTE, KmeansSMOTE, SOMO, DES-MI, GAN, CGAN, CWGAN-GP on the public dataset verifies the effectiveness and advancement of the proposed method. Meanwhile, we obtained the health evaluation sub-models for different hardware devices and then realized the health evaluation overall model according to the hierarchical relationship of the system. Finally, the effectiveness and advancement of the proposed algorithm were verified by comparative experiments on the actual operation dataset of remote maintenance control system of natural gas pipeline.

References

[1] ZHAO G H (2020) Design and implementation of SCADA system for China–Russia Eastern Gas Pipeline Oil & Gas Storage and Transportation vol39 379-388

[2] ZHANG F M and LIU Z P (2017) Combined evaluation methods: a literature review Journal of Systems Engineering vol32 557-569

[3] TAN L T, LI J L, REN B, HE Y, GAO X, XU J H and et al. (2020) Electric Power Automation Equipment vol40 189-198

[4] LI Y X, CHAI Y, HU Y Q and YIN H P (2019) Review of imbalanced data classification methods Control and Decision vol34 673-688

[5] CHAWLA N V, BOWYER K W, HALL L O and Kegelmeyer. (2002) P W. SMOTE: synthetic minority over-sampling technique Journal of Artificial Intelligence Research vol16 321-357

[6] Soltanzadeh P and Hashemzadeh M. (2021) RCSMOTE: Range-Controlled Synthetic Minority Over-sampling Technique for handling the class imbalance problem Information Sciences vol542 92-111.

[7] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S and et al. (2014) Generative Adversarial Networks Advances in Neural Information Processing Systems vol3 2672-2680

[8] Douzas G and Bacao F (2018) Effective data generation for imbalanced learning using conditional generative adversarial networks Expert Systems with Applications vol91 464–471

[9] Mirza M and Osindero S (2014) Conditional Generative Adversarial Nets Computer Science 2672-2680

[10] ZHENG M, LI T, ZHU R, TANG Y H, TANG M J, LIN L L and et al. (2019) Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification Information Sciences vol512 1009-1023

[11] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V and Courville A (2017) Improved Training of Wasserstein GANs Advances in Neural Information Processing Systems 5767-5777

[12] Odena A, Olah C and Shlens J (2017) Conditional Image Synthesis With Auxiliary Classifier GANs The 34nd International Conference on Machine Learning 2642-2651

Table 5. Performance of each approach when applied to the actual dataset.

| Dataset       | RF   | SMOTE+RF | Borderline-SMOTE+RF | KmeansSMOTE+RF | SOMO+RF |
|---------------|------|----------|---------------------|----------------|---------|
| Actual data   | 0.9006 | 0.9232 | 0.9178              | 0.9225         | 0.9006  |

| Dataset       | DES-MI+RF | GAN+RF | CGAN+RF | CWGAN-GP+RF | ACWGAN-GP |
|---------------|-----------|--------|---------|-------------|-----------|
| Actual data   | 0.9090    | 0.9465 | 0.9572  | 0.9427      | 0.9805    |