machines

MDPI

Article

Fault Prediction of On-Board Train Control Equipment Using a CGAN-Enhanced XGBoost Method with Unbalanced Samples

Jiang Liu [1,2,*], Kangzhi Xu [1], Baigen Cai [2,3] and Zhongbin Guo [1,4]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[2] Frontiers Science Center for Smart High-speed Railway System, Beijing Jiaotong University, Beijing 100044, China
[3] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
[4] Beijing Engineering Research Center of EMC and GNSS Technology for Rail Transportation, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: jiangliu@bjtu.edu.cn

Abstract: On-board train control equipment is an important component of the Train Control System (TCS) of railway trains. In order to guarantee the safe and efficient operation of the railway system, Predictive Maintenance (PdM) is significantly required. The operation data of the on-board equipment allow us to build fault prediction models using a data-driven approach. However, the problem of unbalanced fault samples makes it difficult to achieve the expected modeling performance. In this paper, a Conditional Generative Adversarial Network (CGAN) is adopted to solve the unbalancing problem by generating synthetic samples corresponding to specific fault labels that belong to the minority classes. With this basis, a CGAN-enhanced eXtreme Gradient Boosting (XGBoost) solution is presented for training the fault prediction models. From the pre-processing to the field data, artificial fault samples are generated and integrated into the training sample sets, and the XGBoost models can be derived with multiple decision trees. Both the feature importance sequence list and the knowledge graph are derived to describe the characteristics obtained by the models. Filed data sets from practical operation are utilized to validate the proposed solution. By comparison with conventional machine learning algorithms, it can be found that higher accuracy, precision, recall, and F1 scores, which are up to 99.76%, can be achieved by the proposed solution. By involving the CGAN strategy, the maximum enhancement to the F1 score with the XGBoost approach reaches 6.13%. The advantages of the proposed solution show great potential in implementing equipment health management and intelligent condition-based maintenance.

Keywords: train control equipment; predictive maintenance; fault prediction; Conditional Generative Adversarial Network (CGAN); eXtreme Gradient Boosting (XGBoost); unbalanced samples

Citation: Liu, J.; Xu, K.; Cai, B.; Guo, Z. Fault Prediction of On-Board Train Control Equipment Using a CGAN-Enhanced XGBoost Method with Unbalanced Samples. *Machines* **2023**, *11*, 114. https://doi.org/10.3390/machines11010114

Academic Editor: Davide Astolfi

Received: 7 December 2022
Revised: 7 January 2023
Accepted: 12 January 2023
Published: 14 January 2023

1. Introduction

The Train Control System (TCS) plays an important role in the safety assurance of train operation and has been the key to fostering railway transportation system competitiveness. Great effort has been made in finalizing the technical standards of TCS, including the European Train Control System (ETCS) and the Chinese Train Control System (CTCS) [1,2]. In modern TCSs, the increasing autonomy of the train's on-board system indicates a higher requirement to the capability in the full lifecycle, and there is a growing demand for system health monitoring and early assessment. It can be seen in the literature that great effort has been made in the fault detection and system maintenance of the on-board train control equipment, including fault detection and diagnosis [3–5], fault prediction [6], reliability assessment [7], health monitoring [8], and decision-making for optimized maintenance planning [9–11]. A preventive maintenance strategy for the on-board

equipment has been considered in practical railway operation, which means the system maintenance schedule is planned according to the average or expected life time statistics or prediction. However, it is still difficult to precisely predict the developing trend of the performance characteristics and the probability of faults in advance. Thus, unnecessary corrective actions would usually be performed, which cause the inefficient utilization of resources [12]. Furthermore, a large prediction deviation against the truth may lead to delayed and incompatible maintenance decisions, with increased risks to operational safety and efficiency.

In order to enhance the management and maintenance capability of the railway administration, the Predictive Maintenance (PdM) concept [13], which has been investigated by many researchers and industrial manufacturers, offers an effective solution to deploy maintenance more cost effectively. Great effort has been made to develop novel fault detection, reliability prediction, and Remaining Useful Life (RUL) identification algorithms [14–18]. A conventional measure to realize PdM is the involvement of additional sensors and monitoring functions to comprehensively and timely evaluate the fault probability and trend, which has been successfully applied in many industrial fields, as in [19] and [20]. However, things are different for on-board train control equipment because the TCS is safety critical, with a Safety Integrity Level 4 (SIL-4) requirement. It is not possible to involve extra sensors that are not designed in system specifications. Fortunately, an alternative way can be considered by utilizing the log data of the on-board train control equipment, which provides specified recording entities, including system configurations, operation status of specific components, and fault event records. It enables the opportunity to realize PdM with a good compatibility to the system specifications.

To effectively utilize the operation data, it has to be considered that the raw data sets have obvious unbalanced class distribution. Thus, the conventional data-driven modeling algorithms, such as Decision Tree (DT) [21], Random Forest (RF) [22], and AdaBoost (adaptive boosting) [23], may fail to build effective models and guarantee the classification performance with unbalanced data sets. To overcome this problem, generating new artificial samples belonging to the minority fault class will be an effective solution rather than simply copying samples to enhance the balance level of the data sets. Recently, the Generative Adversarial Network (GAN), which was proposed in [24], has been a hot topic in classification applications in generating 2D and 3D objects, faces, anime characters, and even music [25–27]. The GAN provides a framework designed to train implicit generative models using neural networks, because it is capable of learning the distribution of a data set and generating new samples. Inspired by the capability of GAN, we present a new solution to build a fault prediction model of the on-board train control equipment using unbalanced data sets. The features of different fault types can be learned using a sample augmentation logic, which makes it more effective and feasible in practical applications. The contributions of this paper are summarized as follows.

1. Following the data-driven PdM modeling framework, a fault sample enhancement solution is proposed to solve the unbalanced data set problem by using the GAN method. In order to guarantee the quality of the generated artificial samples, the Conditional GAN (CGAN) is adopted to build the conditional generative model through sample-to-sample translation tasks.

2. Using the enhanced training sample sets with a higher data balance level, a fault prediction modeling solution is proposed by combing CGAN with the eXtreme Gradient Boosting (XGBoost) classification algorithm. Under the CGAN-enhanced XGBoost framework, the data-driven fault prediction model, which can be represented by the characteristic importance sequence and the relationship graph, is established to describe the characteristics of the faults in practical operation. Moreover, the effect of the enhanced samples to the fault prediction model is analyzed.

The remainder of this paper is organized as follows. Section 2 shows the architecture of the on-board train control equipment and describes the fault modes. In Section 3, the CGAN-enhanced XGBoost framework is presented, for which the detailed description to

the procedures is presented in Section 4. Investigation of the performance of the proposed solution is given in Section 5, with comparison analysis. Finally, Section 6 concludes the research and points out the future works.

2. System Architecture and Fault Modes of On-Board TCS

The TCS is a vital system that is in charge of controlling the trains' speed and routes. It consists of the on-board sub-system, track-side sub-system, center sub-system, and the communication sub-system. Due to the critical safety requirements, the hardware systems in the TCS are installed with double or triple module systems for a high reliability level. In China, the CTCS framework was established to define the standards of the signaling systems. According to the specifications, CTCS is divided into five levels [28]. Taking CTCS level 2 as an example, it consists of the digital track circuits (or analog track circuits with multi-information), transponders (Balise), and the on-board equipment with an Automatic Train Protection (ATP) function. Using the fixed block mode for train separation, the on-board equipment obtains all the necessary information for train control through the digital track circuit, which can transmit more information than the analog track circuit. As a typical CTCS level 2 system, the CTCS2-200H on-board train control equipment mainly consists of the Vital Computer (VC), Speed and Distance processing Unit (SDU), Balise Transmission Module (BTM), Specific Transmission Module (STM), Driver Machine Interface (DMI), Train Interface Unit (TIU), and Juridical Recorder Unit (JRU). Figure 1 shows the structure of the CTCS2-200H on-board equipment.

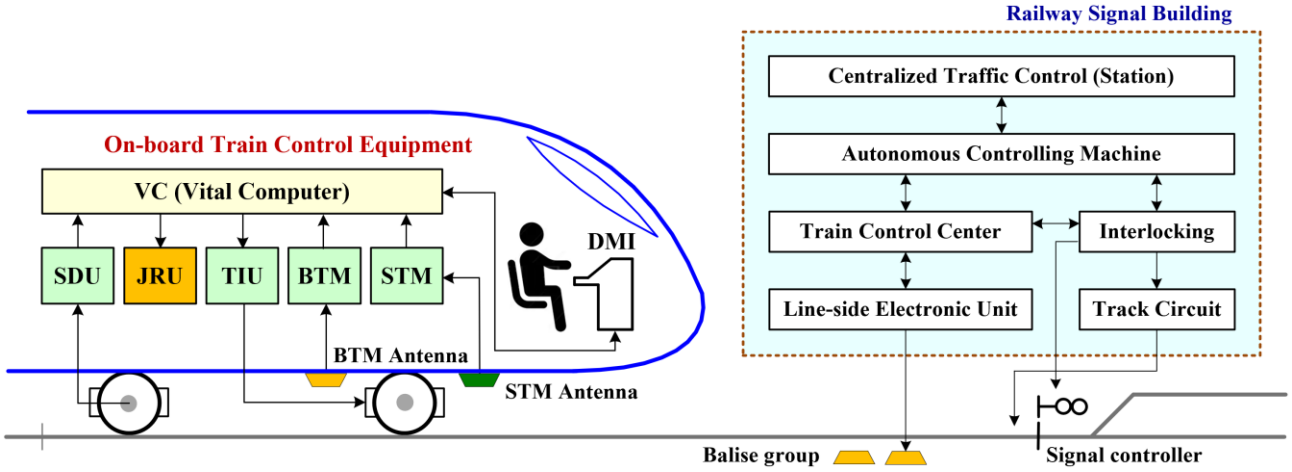

Figure 1. Structure of the CTCS2-200H on-board train control equipment.

It can be seen that the sub equipment cooperate with each other to control the train, and a failure may affect the normal operation of the whole system. In order to guarantee the safe operation of the trains, the active redundancy architecture is adopted in the design of the on-board train control equipment, which requires a fault to be detected before it can be tolerated. When fault detection is identified, typical diagnosis and recovery actions will be triggered. Things may be different under the PdM scheme, with which the occurrence of the fault can be predicted and corresponding maintenance operations will be planned and carried out in advance before it becomes a reality. Using a data-driven method, it is possible for us to build the fault models with data sets under both normal and abnormal operation conditions. Fortunately, JRU in the 200H on-board equipment is designed with the PCMCIA card to record the actions, status, and driver operations to the on-board equipment, which provides a rich information condition and enables the opportunity to utilize machine learning techniques to deliver intelligent decisions for maintenance.

For the CTCS2-200H equipment, several fault types have to be considered by the maintenance system, including hardware faults, software faults, DMI faults, communication faults, BTM faults, STM faults, SDU faults, and environmental interference. For the

data records in the PCMCIA log file, 57 characteristics can be correlated to a specific fault label of each redundant channel. By introducing the normal records and those with fault occurrence in the analytics layer, fault prediction models can be trained and established, which solve the problem in realizing the multivariate analysis or other analytical model-based methods according to the conventional fault prediction solutions.

3. Fault Prediction Modeling Architecture Based on CGAN-Enhanced XGBoost

As an efficient classification method, XGBoost enhances the prediction performance of the final model by combining several poorly performing models. With training data sets from the TCS logs, the normal and fault samples provide the raw measurements that are capable of training the XGBoost model to determine the probability of a specific fault that is covered by the training samples. Different from the Gradient Boosting (GB) method, the XGBoost algorithm offers great performance by controlling the over-learning with more regular pattern formatting [29]. A big concern in building TCS on-board fault prediction models by XGBoost is the distribution of the different types of samples that are available to the model training. Limited fault samples can be obtained by statistical analysis to the whole data platform, making it difficult to effectively utilize the XGBoost and realize its advantages. By relying only on the existing data sets without additional extra assistance, the CGAN-enhanced XGBoost method is proposed to cope with the unbalanced sample problem. The architecture of the proposed method is shown in Figure 2.

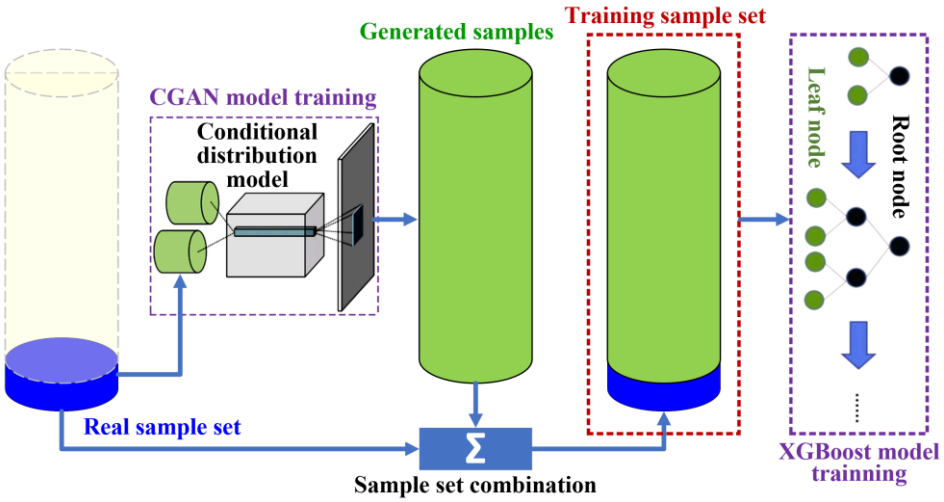

Figure 2. Structure diagram of CGAN-enhanced XGBoost method for fault prediction.

As shown in Figure 2, the establishment of a fault prediction model for the on-board train control equipment can be realized through three key steps: (1) generating new fault samples based on the real sample set through the CGAN model training; (2) combining the real and synthetic data to establish the training sample set; (3) training the XGBoost model for fault prediction. Different from a conventional solution using XGBoost directly, the fault sample balancing operation is adopted to optimize the input training sample set. CGAN is involved in generating artificial samples corresponding to the specific faults of the on-board equipment based on the existing real fault samples from the raw data files. To guarantee the effectiveness of the augmented samples, CGAN is utilized to train a deep learning model for generating new samples that match a given distribution, which constrains the characteristics in the fault samples and ensures the quality of the synthetic samples. Thus, the XGBoost model with the better-balanced training samples can achieve a higher fault prediction capability for PdM applications.

4. Model Training Solution Based on CGAN-Enhanced XGBoost

An enhanced fault prediction model of the on-board train control equipment for PdM can be established with effective model training methods and the desired sample set balance level. Based on the practical features of the raw data samples corresponding to both the normal and fault categories, a premise to ensure the quality of a fault prediction model is the effective solution to alleviate the impact of the imbalance problem. CGAN is applied in this paper to resolve the imbalance effect by generating synthetic samples. The key steps of the fault prediction model training solution are introduced as follows.

4.1. Raw Data Processing

To cope with the data accumulation effect of the operational data logs from the practical on-board equipment, a specific Hadoop platform was established using several servers. Through the time synchronization, firewall configuration, and other system settings, a Hadoop Distributed File System (HDFS) was established, with which the raw data files from the PCMCIA cards in the on-board equipment can be stored for pre-processing and sample set extraction. The HDFS enables the utilization of the raw data from the practical operation. However, it can be seen that the files and raw data cannot be directly utilized because there would be obvious problems in file names, file duplication, and abnormal data formats. Therefore, specific pre-processing operations have to be carried out to ensure the quality of data samples to build fault prediction models. The procedures of the data processing can be found in Figure 3.

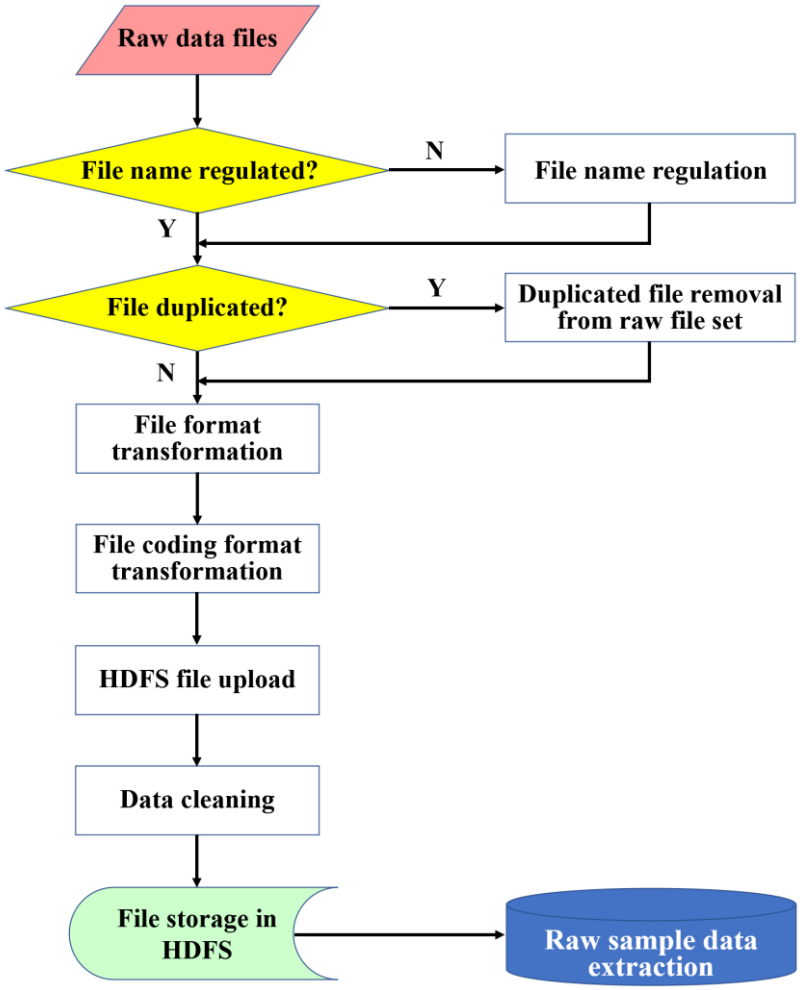

Figure 3. Procedures of raw data processing for training sample extraction.

4.1.1. File Name Regulation

During the practical operation and log file generation process, the file names will be modified by the operators and thus the file name regulation has to be considered in batch processing. For the CTCS2-200H on-board train control equipment, the standardized file name follows the format of "train formation number-head/end label-log generation date (Year/Month/Date)-log generation time (Hour/Minute/Second)", which enables a high degree of recognition for the utilization. According to the standard format of the file name, all the involved files will be examined, and the file name regulation operation will be performed when an abnormal name is detected.

4.1.2. Duplicates Removal

Duplicate files may exist in the raw file sets, including file name duplication and file content duplication. An effective strategy is adopted by statistical analysis with the MD5 algorithm. After the file name regulation, both indices of the file name and file content for the regulated files can be created, with which the frequency of the indices can be evaluated and the file duplication can be identified. For a file that is identified with a duplication label, it will be removed from the raw data set.

4.1.3. Format Transformation

The raw data file was recorded using a specific PCF format. The file format transformation was carried out by the operators through a specific transformation tool. In order to be compatible with the HDFS requirements, the originally used Guojia Biaozhun Kuozhan (GBK) code space will be transformed to the UTF-8 encoding system. Through the file coding format transformation, raw log files can be categorized with different equipment monitoring types, i.e., control information, version information, and Balise information.

4.1.4. Data Cleaning

In order to ensure the quality of the raw data in generating the training samples, the probable mistakes in the data files have to be recognized and corrected. Data cleaning is carried out to enhance the data consistency level through the following operations.

1. Missing field(s) will be filled with the modal number according to the type of the field and the statistical distribution.
2. Abnormal data will be examined and detected according to the corresponding value ranges and correlation of different variables. The default value will be used to fill the field(s) where the original one does not satisfy the range, or when irrationality is detected.
3. Once detected, duplicate data corresponding to the same time instance have to be examined and removed from the raw data set.
4. A new field is constructed because the "train formation number" information only exists in the file name. As an important characteristic quantity, it is added manually to enhance the completeness of the feature set.
5. Data combination is performed to different files corresponding to the same equipment in the same day.

4.1.5. Raw Sample Extraction

Through the data cleaning, all files that pass the data pre-processing will be stored in the HDFS, with which the fault query and statistical analysis can be executed efficiently. To satisfy the requirements of model training for the fault prediction purpose, the raw sample data can be extracted from the whole set concerning specific fault types.

The pre-processing guarantees the quality of the available raw data for model training. Along with the operation of the TCS, the newly collected PCMCIA card data sets will be uploaded into the HDFS. The pre-processing measures will be taken to ensure efficient

data accumulation. Specific program scripts for the above-mentioned operations are developed to realize automatic batch processing.

4.2. Sample Set Augmentation Using CGAN

Statistical analysis has to be performed to monitor the distribution of the raw samples for specific fault types. Unfortunately, for most of the fault types, there is not a large number of negative samples from the field operation. A large number of normal samples will fail to represent and reveal the occurrence and development of the specific faults using conventional classification and modeling algorithms, which ensure that the numbers of samples of each category are at the same level and take the improvement of classification accuracy as a priority. To deal with the emergent unbalance data problem, the generation of synthetic samples by the generative adversarial network allows the opportunity to enhance the quality of the sample set directly rather than tunning the proportion value of different sample categories.

To overcome the unbalanced sample problem, the GAN method provides a framework to train implicit models using neural networks. The GANs are commonly considered as an ideal solution for image generation, with a high quality, stability, and variation compared to the auto-encoders. The GAN is a deep neural network framework that is able to learn from a set of training data and generate new data with the same characteristics as the training data. It consists of two neural networks, the Generator (*G*) and the Discriminator (*D*), which compete against each other [30]. The generator is trained to produce fake data, and the discriminator is trained to distinguish the generator's fake data from real examples. If the generator produces fake data that the discriminator can easily recognize as implausible, the generator will be penalized. Over time, the generator will learn the capability of generating more plausible samples.

Different from the standard GAN, the Conditional GAN method was proposed by [31] to improve the GAN by enabling the opportunity to make inter-class predictions and generate new ballistic samples. It releases the drawback of a standard GAN, which means the output of the Generator (*G*) is limited to data representative of the training set that it was trained on.

The Generator, *G*, is designed to capture the data distribution, which can be represented as $G: \boldsymbol{Z} \to \boldsymbol{\Omega}$, where $\boldsymbol{Z}$ and $\boldsymbol{\Omega}$ represent the noise space of arbitrary dimension, $d_{\boldsymbol{z}}$, corresponding to a hyperparameter and data space. The aim of the Discriminator, *D*, is to determine the probability that the obtained sample is real, which can be described as $D: \boldsymbol{\Omega} \to [0,1]$. Different from the standard GAN, in the CGAN, both the Generator, *G*, and Discriminator, *D*, receive additional information, $\boldsymbol{U}$, to control the sample generation process in a supervised manner, where the fault type of the on-board equipment is used to represent the additional information. CGAN is able to control the number of instances corresponding to a particular label, which cannot be realized by the standard GAN. Beyond the initial definition to the generator and discriminator, they are modified under the CGAN framework as:

$$G_{\mathrm{C}}: \boldsymbol{Z} \times \boldsymbol{U} \to \boldsymbol{\Omega}_{\mathrm{C}} \tag{1}$$

$$D_{\mathrm{C}}: \boldsymbol{\Omega}_{\mathrm{C}} \times \boldsymbol{U} \to [0,1] \tag{2}$$

Thus, the training of the CGAN model involving both the generator and the discriminator plays a minimax game with the following objective function, $F(G_{\mathrm{C}}, D_{\mathrm{C}})$, as in [31]

$$\begin{aligned} \min_{G_{\mathrm{C}}} \max_{D_{\mathrm{C}}} F(G_{\mathrm{C}}, D_{\mathrm{C}}) = {} & E_{x,u \sim P_{data}(x,u)} \left[log\left(D_{\mathrm{C}}(x,u)\right) \right] + \\ & E_{z \sim P_z(z), u \sim P_{data}(u)} \left[log\left(1 - D_{\mathrm{C}}(G_{\mathrm{C}}(z,u),u)\right) \right] \end{aligned} \tag{3}$$

where $(x,u) \in \boldsymbol{\Omega}_C \times \boldsymbol{U}$ is sampled from the distribution $P_{data}(x,u)$, $z \in \boldsymbol{Z}$ and $u \in \boldsymbol{U}$ are sampled from the noise distribution, $P_z(z)$, and the conditional data vectors in the training data are represented by the density function, $P_{data}(u)$.

By using the minority real fault class samples from the raw sample set corresponding to the specific fault type, the generator, G_C, is trained entirely on the performance of the discriminator, D_C, and its parameters will be optimized in accordance with the output $D_C(G_C(z))$. Figure 4 illustrates the workflow of the CGAN augmentation process. When the discriminator is trained with an expected capability of successfully classifying the fake samples, G_C will have to be rigorously updated to enhance the capability of sample generation. Conversely, when the discriminator is less successful in identifying the fake samples, D_C will be highly concerned and updated more rigorously. The evaluation and updating strategies indicate the zero-sum adversarial relationship between G_C (attack-side) and D_C (defense-side). A combined model is trained to update and form the CGAN network that stacks both the G_C and D_C. Through the competition of the two networks against each other, the generator becomes better at creating the synthetic realistic fault samples corresponding to the minority class. The generation capability will be applied in enhancing the sample balance level and building training sets for establishing the fault prediction models.

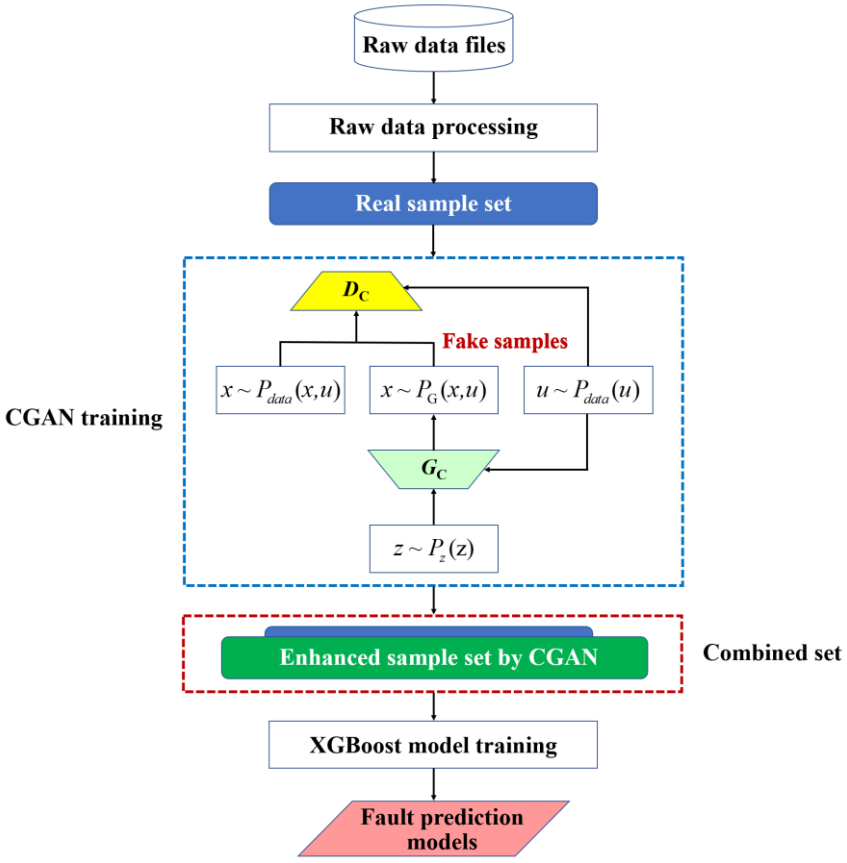

Figure 4. Work low of the CGAN augmentation process.

4.3. XGBoost Training with Enhanced Sample Set

Using the CGAN, the problem of unbalanced samples can be solved with more fault samples, and the distribution of different classes can be effectively controlled. Establishment of the fault prediction model can be achieved through three steps (see Figure 5).

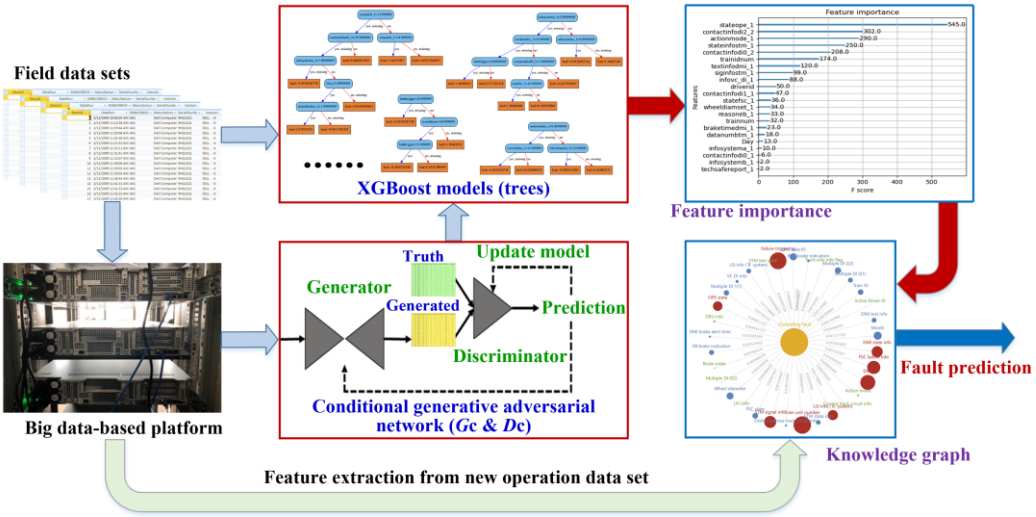

Figure 5. Procedures of fault prediction model training by CGAN and XGBoost.

1. Training set synthesis

With a preset ratio of the fault samples in the whole training set, an expected number of fault samples will be generated by the CGAN network, and the derived samples will be integrated with the real fault and normal samples to form the target sample set for the final training phase.

2. XGBoost model training

Based on the gradient boosting method, the XGBoost algorithm is capable of realizing a better performance level over existing solutions such as RF [22], Gradient-boosted Decision Tree (GBDT) [32], and AdaBoost [23]. It is a scalable, distributed gradient-boosted decision tree machine learning method, providing parallel tree boosting for regression, classification, and ranking problems. Using the training sample set $\boldsymbol{S}_t$ with both the real and CGAN-derived synthetic samples, which consists of fault label pairs, $\{\boldsymbol{f}_t, v_t\}$, the j th regularized objective function of the additive training method can be obtained as:

$$\Delta_j = \sum_{\{\boldsymbol{f}_t, v_t\} \in \boldsymbol{S}_t} q(v_t, M_{j-1}(\boldsymbol{f}_t) + m_j(\boldsymbol{f}_t)) + h(m_j) \tag{4}$$

where $\boldsymbol{f}_t$ denotes the feature vector of a training sample at instant t, v_t represents the fault state label of the feature sample $\boldsymbol{f}_t$, $q(*)$ is the log-likelihood loss function between the label v_t and the model prediction output, $m_j(*)$ denotes the weak learners at the j th boosting round, $M_{j-1}(*) = \sum_{k=0}^{j} m_k(*)$ is the ensemble at the $(j–1)$ th boosting round, and $h(*)$ indicates the regularization function to penalize the model complexity of the weak learner, m_j, of the XGBoost.

Through the additive training, the final output, $M_R(*)$, will be taken as the final classifier, where R represents the required number of the boosting rounds.

3. Model-based fault probability prediction

The new-coming data set can be utilized to realize prediction using $M_R(*)$. By extracting the feature vector, $\boldsymbol{f}_w$, at instant w, the occurrence probability of the target fault event, v, can be evaluated using the XGBoost as:

$$p_{\mathrm{XGB}}(\boldsymbol{f}_w) = M_R(\boldsymbol{f}_w) \tag{5}$$

The procedures of the XGBoost training method are summarized as Algorithm 1.

Algorithm 1 XGBoostTraining

1: Initialize the regression tree.
2: Establish the candidate set based on the feature sub-set $\boldsymbol{c}_j$.
3: ***while*** the tree depth is less than the maximum depth, **do**
4: ***for*** k = 1,2,…,T **do**
5: Calculate the partial derivatives of each feature in the candidate sub-set.
6: Evaluate the minimum gain that is used as the split point.
7: Inject the features out of the candidate sub-set into $\boldsymbol{c}_j$, and repeat 5 and 6.
8: $n + 1 \rightarrow n$
9: Calculate the sample weight of the leaf node as W_n.
10: ***if*** $W_n < \tilde{W}$ (where $\tilde{W}$ represents the minimum leaf node weight)
11: Terminate the node split.
12: ***end if***
13: ***end for***
14: Continue to generate trees until the number of trees reaches N_{T}.
15: ***end while***
16: Calculate the model prediction, $p_{\mathrm{XGB}}(\boldsymbol{f}_w)$, with the newly extracted features.

With the derived XGBoost model, the importance scores corresponding to the features of the samples can be obtained. Through the ranking of the derived feature importance scores, the relationship of each feature to the specific fault label can be effectively reflected. Besides the score ranking, the knowledge graph corresponding to the target fault type can be constructed to demonstrate the details of the model. Figure 5 depicts the procedures of fault prediction model training by the integration of CGAN and XGBoost and derivation of the knowledge graph.

4.4. Summary of the CGAN-XGBoost Solution

Through the procedures of the CGAN-enhanced XGBoost solution, it can be seen that the proposed solution takes advantages of the XGBoost method by solving the unbalanced sample problem during the training set construction phase. The performance of the proposed solution can be evaluated in terms of the class distribution tolerance, fault prediction accuracy, fault type coverage, and computational efficiency.

1. Class distribution tolerance. The existing machine learning-enabled fault prediction modeling methods achieve effective utilization to the specific machine learning algorithms. However, the significant unbalanced data problem in the TCS application constrains the achievable performance for the fault type classification. The proposed solution chooses a different way to release the requirement to the data balance level. CGAN enables the benefits of constructing additional realistic samples and realizes an enhanced class distribution tolerance ability.
2. Fault prediction accuracy. Both the enhancement of the sample balance level and the utilization of the effective XGBoost method guarantee a high performance of fault prediction. The XGBoost-based model is fed with the enhanced training samples to achieve the desired model performance, which satisfies the practical requirement and the constrained operational conditions of the on-board train control equipment.
3. Feature coverage. The involved data-driven CGAN method for generating synthetic samples corresponding to the minority fault type(s) in the raw data set enables the adaptive performance to different fault-related features. The enhanced ability of covering the minority fault classes leverages the fault recognition accuracy and the sample balance rate over the conventional machine learning solutions.
4. Computational efficiency. Compared with the original XGBoost-based model training method and other solutions using similar classifiers, it can be seen that only a

sample balance enhancement step is embedded by the proposed solution. The newly added part only requires an off-line training operation before the construction of the fault prediction models. The CGAN logic is easy to embed into the following model update phase along with the long-term operation of the on-board train control equipment, which is significant from the full life cycle perspective.

5. Test and Evaluation

In this section, evaluation and analysis are carried out to demonstrate the proposed CGAN-enhanced XGBoost solution using the field operational data sets from real CTCS2-200H on-board train control equipment. A Hadoop data platform was established with three servers (Intel Xeon CPU E5-2620 v4 @ 2.10GHz, 64GB RAM) to manage and process the obtained data sets. Specific software was used for modeling and testing, including Python 3.9.7, XGBoost 1.6.1, and PyTorch 1.13.0. We first investigated the capability of the involved CGAN method using practical fault samples with a limited class volume. Based on that, the CGAN-enhanced XGBoost method was carried out with the integrated sample sets, including the real and CGAN-generated samples. Finally, the performance of the whole CGAN-XGBoost solution was evaluated through comparison analysis concerning the conventional solution using only the standard XGBoost.

5.1. Evaluation of CGAN-Enabled Sample Generation

Data preparation was performed using real operation data sets in 2018. The unbalanced data problem makes it difficult to generating effective fault prediction models. Concerning the fault type "STM fault", it can be seen that the ratio of the fault class samples in the whole data set may be less than 1:1000. Taking a specific raw sample set for example, it can be seen that there are over 400,000 normal samples, while only 442 fault samples were recorded during the practical operation. An obvious unbalancing status can be identified because the limited fault samples are not sufficient to reflect the strong relationship between the involved features and the fault label. Therefore, the CGAN method is adopted to enhance the fault samples for solving the unbalancing problem in the model training phase. In the training of the generator and discriminator, "LeakReLu" is adopted as the activation function in the middle layers, and the output layers use the "Sigmoid". The Adam optimizer, which gives much higher performance than conventional optimizers and outperforms them by a big margin into giving an optimized gradient descent [33], is introduced in the training of both the generator and the discriminator.

With a minority fault sample set corresponding to the "STM fault" label, the CGAN was trained by data transformation and normalization to the real sample feature data. The CGAN was gradually improved with several training cycles. After the training phase, 40 generator models were obtained to be utilized in constructing new fault samples. To evaluate the effectiveness of the derived generation models, the fixed input data of noise was adopted, and the difference of the involved sample features between the generated (G) and the true features (T) can be compared, as depicted in Figure 6, where the results from 9 specific epochs, within 30 training cycles, are demonstrated.

From the comparison results, it can be seen that the capability of the generator models was gradually improved with the training cycle. The generated feature values achieve a higher consistency when more cycles of training have been carried out, and thus these models will be more effective in generating trustworthy artificial samples that precisely reflect the relationship between the features and the corresponding fault label.

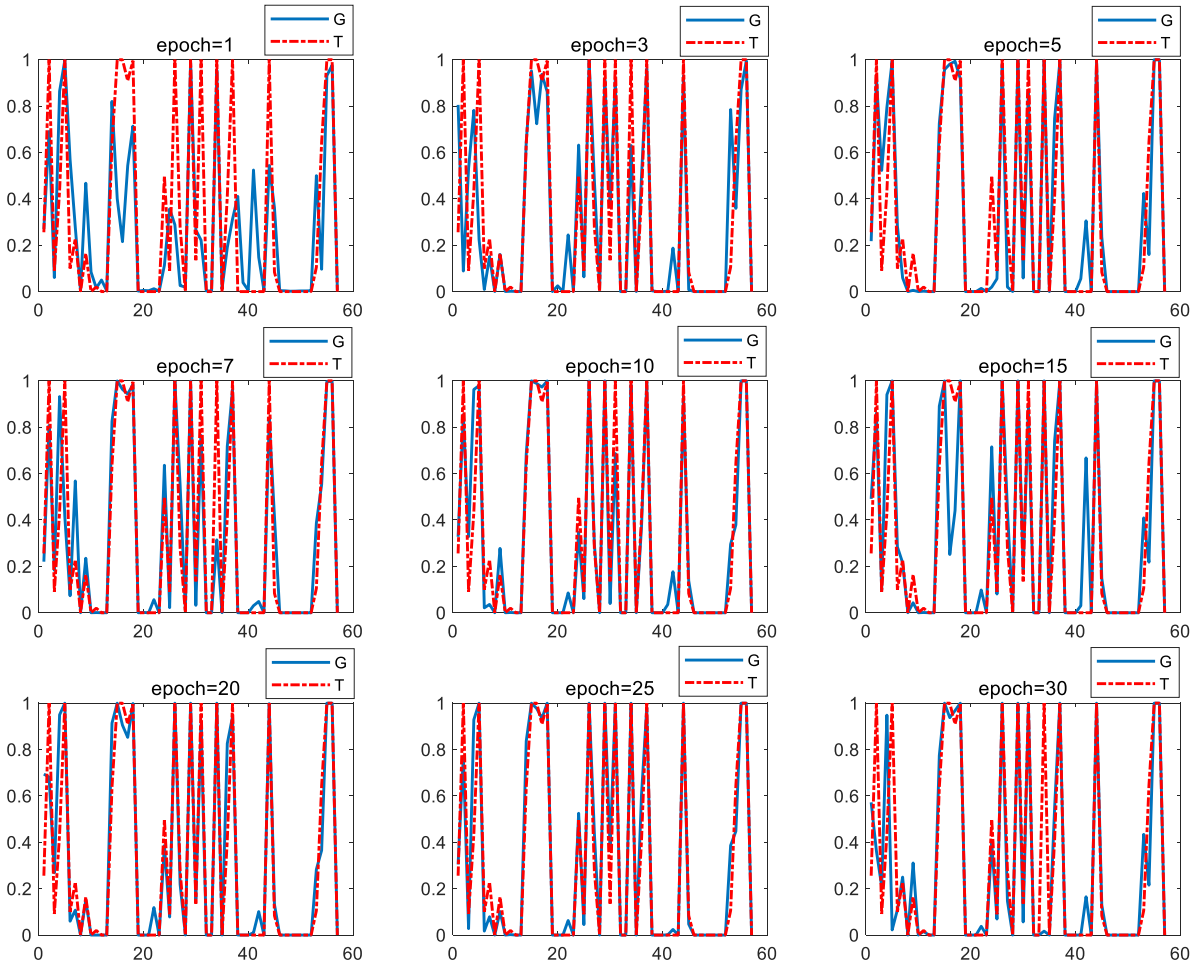

Figure 6. Comparison of the generated and real sample features at different epochs.

Besides the deviation analysis to the real and generated samples, the data distribution provides another way to examine the capability of CGAN in generating realistic samples. By examining the importance of all the sample features using a pre-trained XGBoost classifier, it can be seen that there are six typical features with a higher influence on the modeling description, including "stateope1_1", "contactinfodi0_2", "siginfstm_1", "trainnum", "actionmode_1", and "contactinfodi2_2". To clearly show the statistical characteristics of both the real and generated fault sample set, box plots corresponding to these features are shown in Figure 7, where red central lines indicate the median values and the red dots (with the maker style "+") placed past the edges represent the outliers.

From the comparison between the real and artificial samples, it can be seen that the values of the independently generated samples are well constrained within the expected ranges, which ensure the rationality of the generators. Furthermore, the generated samples are able to realize a similar range of distribution to the real ones. From the statistical analysis perspective, the distribution of the generated samples behaves with a high similarity, especially for those features that will highly affect the fault occurrence. Meanwhile, a good diversity of the sample data can be achieved so that the quality of the integrated training sample set will be well guaranteed.

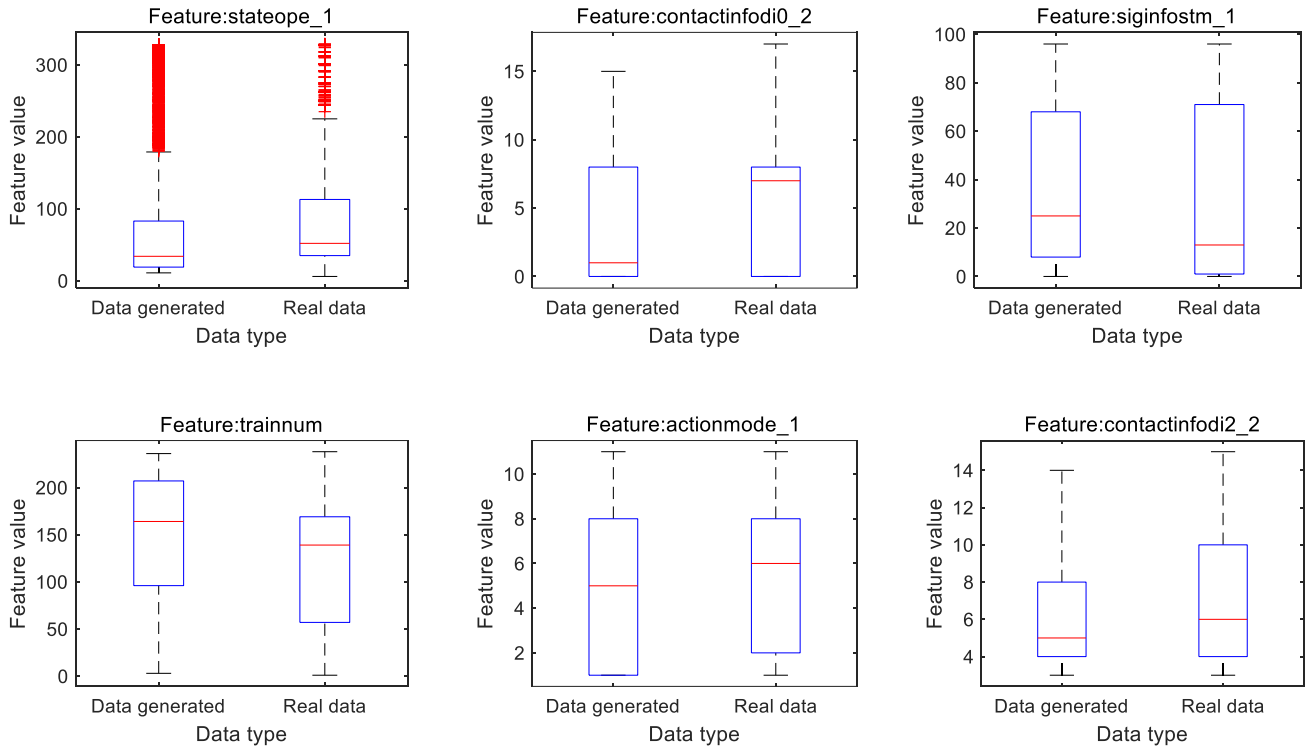

Figure 7. Comparison of the data distribution corresponding to six main features.

5.2. Validation of CGAN-Enhanced XGBoost Modeling

By utilizing the CGAN-enhanced training sample set, the XGBoost classifier can be trained to recognize the occurrence probability of the specific faults. Three raw sample sets, as follows, are utilized in CGAN enhancing and XGBoost training to validate the performance of the proposed modeling solution.

1. The 2018-T(A) sample set contains 442,000 samples with only 442 fault samples. Through enhancement by the CGAN network, an extended fault sample set with 4420 samples is derived to improve the data balancing level.
2. The 2018-T(B) sample set contains 237,000 samples with only 237 fault samples. An enhanced fault sample set with 2370 samples is constructed for XGBoost model training.
3. The 2018-T(C) sample set contains 679,000 samples with only 679 fault samples. Aimed at realizing a fault class ratio of 1:100, an enhanced sample set with 6790 fault samples is established by the well-trained CGAN.

For the XGBoost training, the booster parameters play a significant role in determining the achievable performance of the derived training results. The maximum depth of trees in the model training with the three sample sets is set as 5 to ensure a rapid convergence rate. The number of iterations is set as 500, which indicates the maximum number of the derived decision trees. Figures 8–10 show the results of the XGBoost model training, with both the feature importance sequence and the knowledge graph. It is noted that not all the features in the data sample are involved in describing the feature importance sequence. A threshold to the feature importance score is defined according to an accumulative contribution rate of 80% when the all the feature scores are ranked in a descending order. Thus, only those features with high scores exceeding the threshold will be involved in demonstrating the training results in these figures.

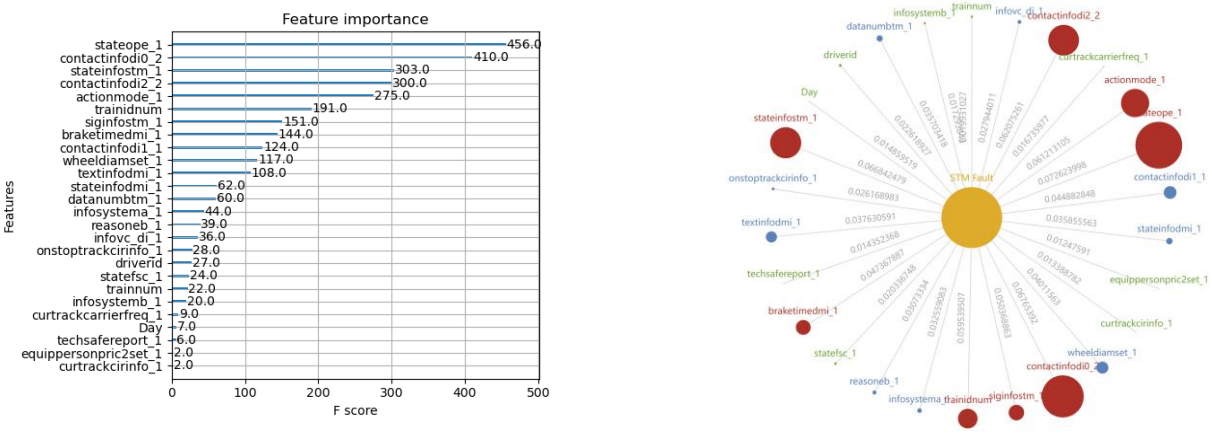

Figure 8. Results of XGBoost training using CGAN-enhanced 2018-T(A) sample set.

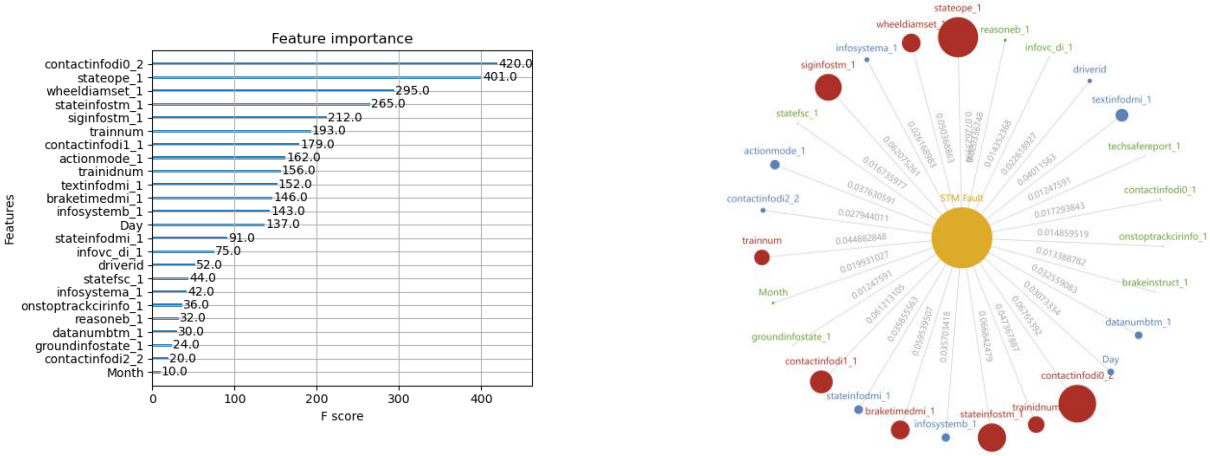

Figure 9. Results of XGBoost training using CGAN-enhanced 2018-T(B) sample set.

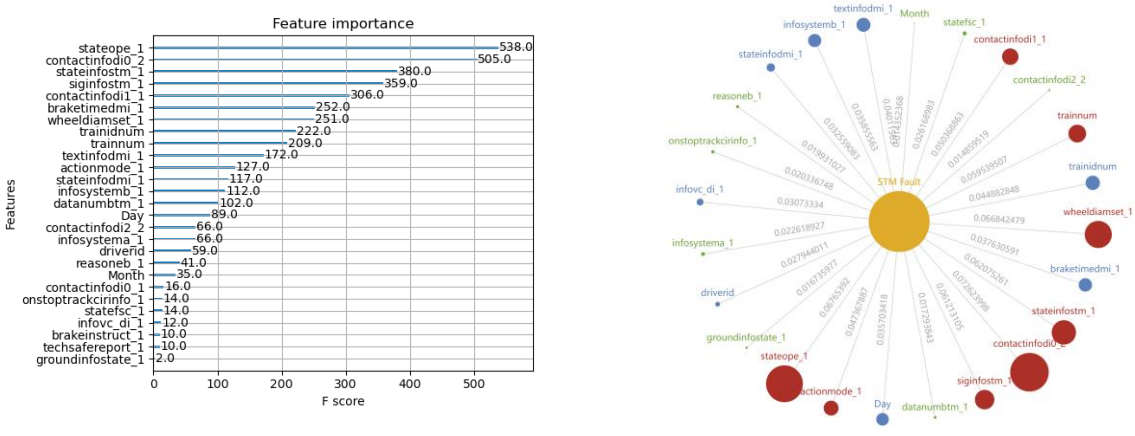

Figure 10. Results of XGBoost training using CGAN-enhanced 2018-T(C) sample set.

The obtained XGBoost model consists of many decision trees, and a new tree will be added into the model in each iteration cycle. With the increasing sub-models along with the training process, the complexity of the ensemble model will grow gradually. When the training is finished, an ensemble of trees can be derived, with which prediction can be performed for the equipment health assessment and other related PdM operations. Figures 11–13 show the typical sub-models in the training with the three training samples.

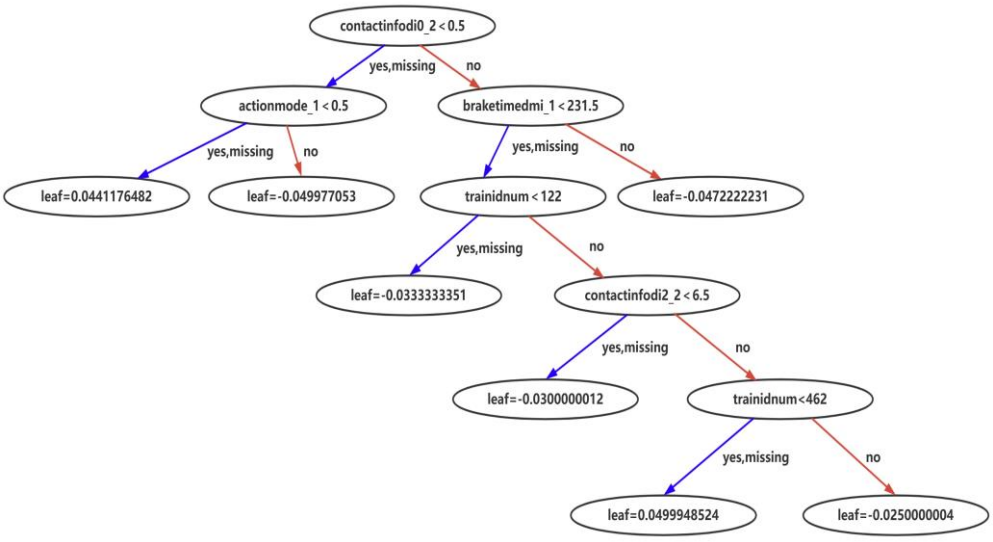

Figure 11. Typical sub-model (tree) of the ensemble by the training with 2018-T(A).

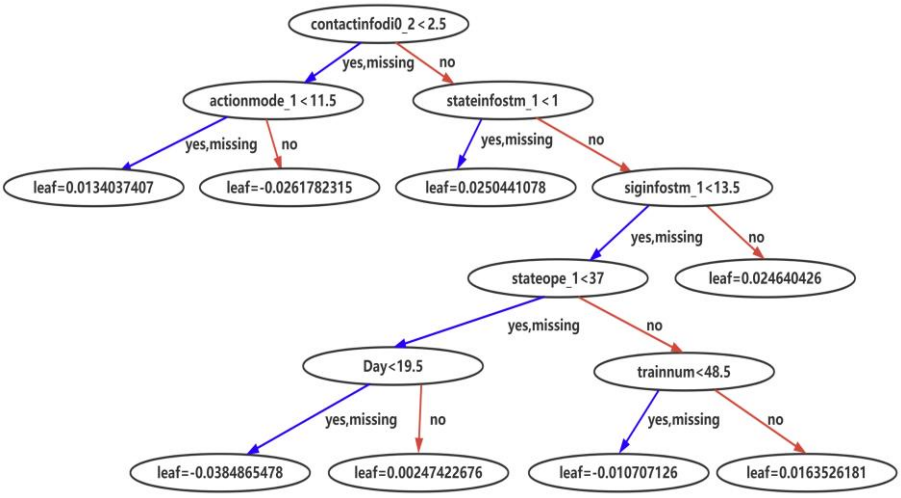

Figure 12. Typical sub-model (tree) of the ensemble by the training with 2018-T(B).

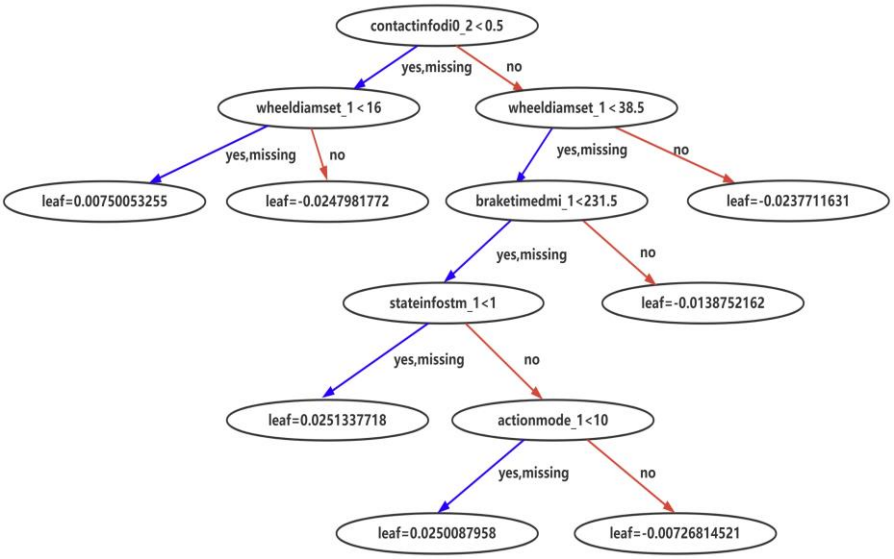

Figure 13. Typical sub-model (tree) of the ensemble by the training with 2018-T(C).

From the model training using XGBoost, it can be seen that the proposed solution is designed with both deep consideration in terms of data balance enhancement and the principles in machine learning. The integrated sample set guarantees the fulfillment of the pre-condition of data balancing level for exploring the potential of XGBoost, while the XGBoost training derives an ensemble of trees for fault prediction. The feature importance sequence and the knowledge graph enable different expressions to the core of the derived models that can be transferred to the knowledge. The results are user-friendly for practical equipment management and maintenance operation.

5.3. Comparison Analysis of the CGAN Enhancement

To further investigate the performance of the proposed CGAN-enhanced XGBoost solution for fault prediction of on-board train control equipment, different typical machine learning algorithms, including Radom Forest, Gradient-boosted Decision Tree, and Adaptive Boosting, are involved for a comparison analysis. To emphasize the necessity and performance of the involvement of the CGAN enhancement logic, all the referencing algorithms and XGBoost are carried out twice in parallel, with both the unbalanced raw sample sets and the CGAN-enhanced sets. An independent test sample set collected in July 2018 is used to evaluate the prediction performance of different algorithms.

Using TP, TN, FP, and FN as basic measures to describe the classification performance, specific criterions can be derived to quantitatively indicate the model prediction capability, where TP represents the number of positive samples that is determined as true by the model, TN indicates the number negative samples that is classified as true by the model, FP denotes the number of positive samples that is determined as false, and FN represents the number of negative samples that is identified as false by the model.

1. Accuracy. Indicates the general classification accuracy of the model, which can be calculated as (TP + TN)/(TP + TN + FP + FN).
2. Precision. Represents the proportion of the actual true positive samples to all the samples that the model recognizes as positive. It is calculated as TP/(TP + FP).
3. Recall. Describes the classification accuracy of the model to positive samples and can be calculated as TP/(TP + FN).
4. F1 score. A statistical measure of the accuracy of the model. It is defined as the harmonic mean of recall and precision as 2 × Recall × Precision/(Recall + Precision).

In the comparative evaluation, it is considered that the iteration number in the model training process may affect the achieved performance level. Therefore, results with different iteration numbers (100, 200, and 300) for XGBoost training are first investigated. As shown in Figures 14 and 15, it can be clearly seen that a large iteration number will result in a higher model performance, no matter whether or not sample enhancement by CGAN is adopted. To realize a detailed analysis to the criterions from different algorithms, all the evaluation results of all involved algorithms after 300 iterations are summarized in Tables 1 and 2.

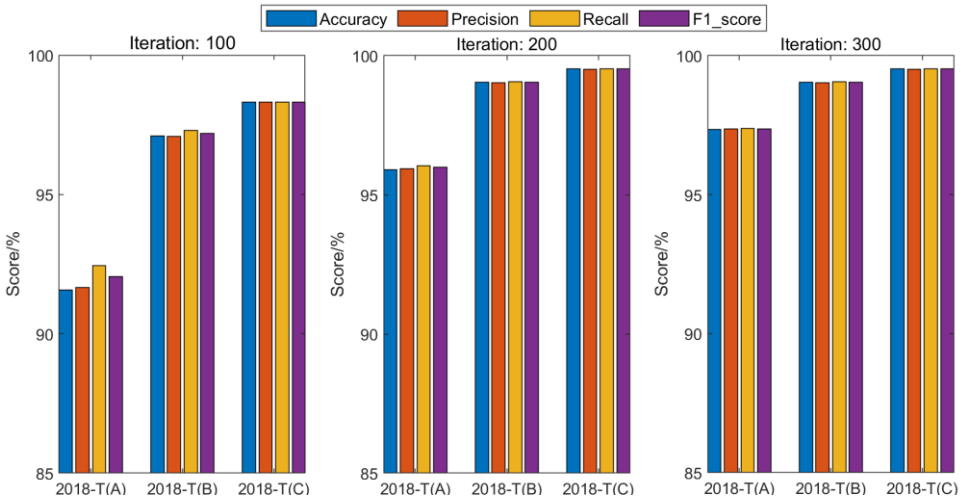

Figure 14. XGBoost performance using raw sample sets with different iteration numbers.

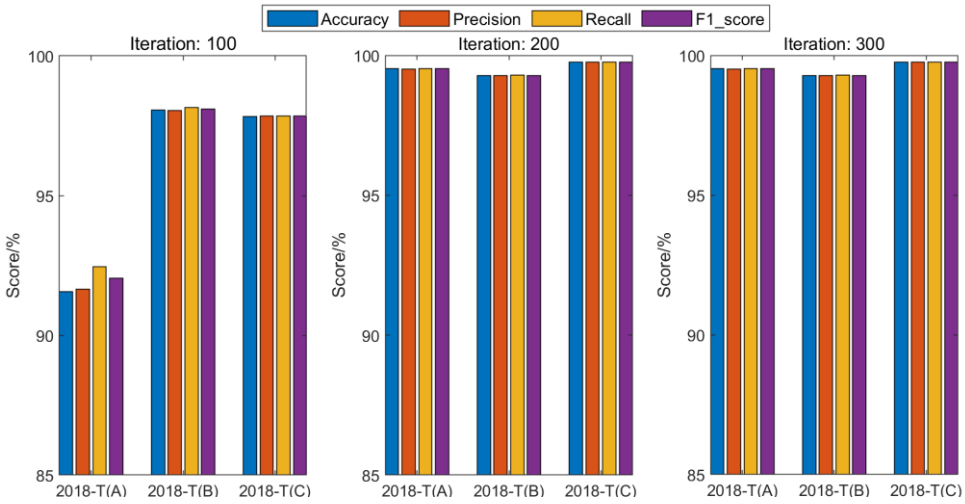

Figure 15. XGBoost performance using CGAN-enhanced sets with different iteration numbers.

Table 1. Performance comparison of different algorithms using raw sample sets (%).

Training Set	Algorithm	Accuracy	Precision	Recall	F1
2018-T(A)	RF	93.73	93.78	93.90	93.84
	GBDT	88.92	89.04	90.66	89.84
	AdaBoost	99.04	99.02	99.06	99.04
	XGBoost	97.35	97.36	97.38	97.37
2018-T(B)	RF	98.55	98.54	98.61	98.57
	GBDT	96.87	96.86	96.87	96.86
	AdaBoost	98.31	98.30	98.38	98.34
	XGBoost	99.04	99.02	99.06	99.04
2018-T(C)	RF	98.55	98.54	98.61	98.57
	GBDT	97.83	97.84	97.84	97.84
	AdaBoost	97.83	97.80	97.94	97.76
	XGBoost	99.52	99.51	99.53	99.52

Table 2. Performance comparison of different algorithms using CGAN-enhanced sample sets (%).

Training Set	Algorithm	Accuracy	Precision	Recall	F1
2018-T(A) (CGAN+)	RF	95.42	95.44	95.46	95.45
	GBDT	93.49	93.56	93.99	93.77
	AdaBoost	99.04	99.02	99.06	99.04
	XGBoost	99.52	99.51	99.53	99.52
2018-T(B) (CGAN+)	RF	98.31	98.30	98.38	98.34
	GBDT	98.31	98.32	98.32	98.32
	AdaBoost	98.80	98.78	98.84	98.81
	XGBoost	99.28	99.27	99.30	99.28
2018-T(C) (CGAN+)	RF	98.55	98.54	98.61	98.57
	GBDT	98.31	98.32	98.32	98.32
	AdaBoost	98.80	98.78	98.84	98.81
	XGBoost	99.76	99.76	99.76	99.76

From the comparison results, it can be seen that the iteration number plays an important role in determining the achieved performance by XGBoost training. After 300 iterations, an improved performance level of the trained XGBoost models can be obtained in both the raw set and the CGAN-enhanced set situations. Considering the effectiveness of the CGAN network for sample balancing, the XGBoost model with the enhanced sample set obviously outperforms the model that uses the raw data set only. Results of the 2018-T(A) sample set illustrate the most obvious improvement in all the criterions, which are increased by 2.23%, 2.21%, 2.21%, and 2.21%, respectively. The 2018-T(A) set has a time interval of over six months from the test sample set, which is the largest among the three training sets. The effective enhancement by the CGAN to the 2018-T(A) set demonstrates the significance of solving the data balancing problem in realizing the desirable model adaptability and the temporal coverage to the raw filed data sets. By comparing XGBoost with other machine learning algorithms, it can be seen that XGBoost does not perform the best when a specific raw sample set is adopted. For the 2018-T(A) sample case, AdaBoost earns a higher performance level in constructing the fault prediction model, while XGBoost models perform better using the 2018-T(B) and 2018-T(C) sets. When the integrated sample sets are adopted with the CGAN-based enhancement, it can be seen that the performance of all these algorithms can be improved over the unbalanced sample set cases, while XGBoost outperforms all other algorithms. By examining the F1 score results, for example, the maximum enhancement by XGBoost reaches 6.13%, 0.98%, and 1.46%, corresponding to the three sample sets, respectively. The derived results demonstrate that the CGAN-enhanced XGBoost solution offers more opportunities for a fault prediction model to map the features of the sample sets and realize better prediction outputs. It reveals that the proposed architecture for fault prediction model training enable more possibilities to capture and evaluate the fault characteristics. This solution strengthens the potential in realizing the PdM of on-board train control equipment.

6. Conclusions

This paper points out the deficiencies of conventional machine learning methods in coping with unbalanced samples for the predictive maintenance of on-board train control equipment. A novel CGAN-enhanced XGBoost solution is proposed for establishing the fault prediction models using only the raw unbalanced operational log data. The proposed solution can actively generate artificial fault samples that belong to the minority class in the raw field data sets and provide great support to the model training by the XGBoost method. It is different from conventional solutions that only use the standard modeling algorithms without augmentation to the unbalanced samples. The derived fault prediction models can take full advantages of the XGBoost approach and realize an

improved prediction performance level. The feature importance sequence and knowledge graph by the derived XGBoost models enable effective interfaces to potential advanced PdM applications. It is indicated that an enhanced model adaptability and temporal coverage can be achieved, enabled by the active sample generation and tunning capabilities of the CGAN-based sample augmentation logic. In the next phase of this research, the operational reliability assessment and maintenance decision-making will be used to utilize the fault prediction models in the advanced maintenance of the train control system.

Author Contributions: Conceptualization, B.C. and J.L.; methodology, K.X. and Z.G.; software, Z.G.; validation, K.X. and J.L.; formal analysis, J.L.; investigation, K.X.; resources, B.C.; data curation, K.X.; writing—original draft preparation, J.L.; writing—review and editing, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beijing Natural Science Foundation (L191014), National Natural Science Foundation of China (U2268206, T2222015), Key Fields Project of DEGP (2021ZDZX1110), and Fundamental Research Funds for the Central Universities (2022JBQY003).

Data Availability Statement: Not applicable.

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

References

1. Cao, Y.; Ma, L.; Xiao, S.; Zhang, X.; Xu, W. Standard analysis for transfer delay in CTCS-3. *China J. Electron.* **2017**, *26*, 1057–1063.
2. Ranjbar, V.; Olsson, N.; Sipila, H. Impact of signalling system on capacity—Comparing legacy ATC, ETCS level 2 and ETCS hybrid level 3 systems. *J. Rail Transp. Plan. Manag.* **2022**, *23*, 1–14.
3. Sang, J.; Guo, T.; Zhang, J.; Zhou, D.; Chen, M.; Tai, X. Incipient fault detection for air brake system of high-speed trains. *IEEE Trans. Control Syst. Technol.* **2021**, *29*, 2026–2037.
4. Chen, H.; Jiang, B.; Chen, W.; Li, Z. Edge computing-aided framework of fault detection for traction control systems in high-speed trains. *IEEE Trans. Veh. Technol.* **2020**, *69*, 1309–1318.
5. Liu, Z.; Zhang, M.; Liu, F.; Zhang, B. Multidimensional feature fusion and ensemble learning-based fault diagnosis for the braking system of heavy-haul train. *IEEE Trans. Ind. Inform.* **2020**, *17*, 41–51.
6. Kang, R.; Wang, J.; Chen, J.; Zhou, J.; Pang, Y.; Guo, L.; Cheng, J. A method of online anomaly perception and failure prediction for high-speed automatic train protection system. *Reliab. Eng. Syst. Saf.* **2022**, *226*, 1–11.
7. Chai, N.; Zhou, W. Evaluating operational risk for train control system using a revised risk matrix and FD-FAHP-Cloud model: A case in China. *Eng. Fail. Anal.* **2022**, *137*, 1–27.
8. Shi, L.; Chen, L. Hazard recognition and reliability analysis of CTCS-3 on-board subsystem. *Comput. Commun.* **2020**, *151*, 145–153.
9. Lu, C.; Cai, C. Overview on safety management and maintenance of high-speed railway in China. *Transp. Geotech.* **2020**, *25*, 1–8.
10. Lin, B.; Zhao, Y. Synchronized optimization of EMU train assignment and second-level preventive maintenance scheduling. *Reliab. Eng. Syst. Saf.* **2021**, *215*, 1–11.
11. Tian, Q.; Wang, H. Optimization of preventive maintenance schedule of subway train components based on a game model from the perspective of failure risk. *Sustain. Cities Soc.* **2022**, *81*, 1–14.
12. Yu, W.; Dollon, T.; Mostafa, F.; Rahayu, W.; Liu, X. A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Trans. Ind. Inform.* **2020**, *16*, 183–192.
13. Nunes, P.; Santos, J.; Rocha, E. Challenges in predictive maintenance—A review. *CIRP J. Manuf. Sci. Technol.* **2023**, *40*, 53–67.
14. Li, Z.; He, Q. Prediction of railcar remaining useful life by multiple data source fusion. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2226–2235.
15. Leite, M.; Costa, M.; Alves, T.; Infante, V.; Andrade, A. Reliability and availability assessment of railway locomotive bogies under correlated failures. *Eng. Fail. Anal.* **2022**, *135*, 1–18.
16. Chen, Y.; Tian, Z.; Roberts, C.; Hillmansen, S.; Chen, M. Reliability and life evaluation of a DC traction power supply system considering load characteristics. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 958–968.
17. Atamuradov, V.; Medjaher, K.; Camci, F.; Dersin, P.; Zerhouni, N. Railway point machine prognostics based on feature fusion and health state assessment. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 2691–2704.
18. Chen, Y.; Wang, Z.; Cai, Z. Optimal maintenance decision based on remaining useful lifetime prediction for the equipment subject to imperfect maintenance. *IEEE Access* **2020**, *8*, 6704–6716.
19. Liu, B.; Do, P.; Lung, B.; Xie, M. Stochastic filtering approach for condition-based maintenance considering sensor degradation. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 177–190.
20. Yildirim, M.; Gebraeel, N.; Sun, X. Integrated predictive analytics and optimization for opportunistic maintenance and operations in wind farms. *IEEE Trans. Power Syst.* **2017**, *32*, 4319–4328.

21. Jin, C.; Li, F.; Ma, S.; Wang, Y. Sampling scheme-based classification rule mining method using decision tree in big data environment. *Knowl. -Based Syst.* **2022**, *244*, 1–14.
22. Gao, W.; Xu, F.; Zhou, Z. Towards convergence rate analysis of random forests for classification. *Artif. Intell.* **2022**, *313*, 1–34.
23. Biag, M.; Awais, M.; El-alfy, E. AdaBoost-based artificial neural network learning. *Neurocomputing* **2017**, *248*, 120–126.
24. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144.
25. Haque, K.; Rana, R.; Liu, J.; Hansen, J.; Cummins, N.; Busso, C.; Schuller, B. Guided generative adversarial neural network for representation learning and audio generation using fewer labelled audio data. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 2575–2590.
26. Abu-Srhan, A.; Abushariah, M.; Al-Kadi, O. The effect of loss function on conditional generative adversarial networks. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 6977–6988.
27. Su, Y.; Meng, L.; Kong, X.; Xu, T.; Lan, X.; Li, Y. Small sample fault diagnosis method for wind turbine gearbox based on optimized generative adversarial networks. *Eng. Fail. Anal.* **2022**, *140*, 1–16.
28. Zhang, Y.; Wang, H.; James, P.; Roggenbach, M.; Tian, D. A train protection logic based on topological manifolds for virtual coupling. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 93–11945.
29. Binson, V.; Subramoniam, M.; Sunny, Y.; Methew, L. Prediction of pulmonary diseases with electronic nose using SVM and XGBoost. *IEEE Sens. J.* **2021**, *21*, 20886–20895.
30. Thompson, S.; Teixeira-Dias, F.; Paulino, M.; Hamilton, A. Predictions on multi-class terminal ballistics datasets using conditional Generative Adversarial Networks. *Neural Netw.* **2022**, *154*, 425–440.
31. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
32. Ma, X.; Ding, C.; Luan, S.; Wang, Y.; Wang, Y. Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2303–2310.
33. Kingma, D.P.; Ba, J. ADAM: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.