# A Fault Diagnosis Method for Smart Meters via Two-layer Stacking Ensemble Optimization and Data Augmentation

Leijiao Ge, *Senior Member*, *IEEE*, Tianshuo Du, Zhengyang Xu, Luyang Hou, Jun Yan, *Member*, *IEEE*, and Yuanliang Li

***Abstract*—The accurate identification of smart meter (SM) fault types is crucial for enhancing the efficiency of operation and maintenance (O&M) and the reliability of power collection systems. However, the intelligent classification of SM fault types faces significant challenges owing to the complexity of features and the imbalance between fault categories. To address these issues, this study presents a fault diagnosis method for SM incorporating three distinct modules. The first module employs a combination of standardization, data imputation, and feature extraction to enhance the data quality, thereby facilitating improved training and learning by the classifiers. To enhance the classification performance, the data imputation method considers feature correlation measurement and sequential imputation, and the feature extractor utilizes the discriminative enhanced sparse autoencoder. To tackle the interclass imbalance of data with discrete and continuous features, the second module introduces an assisted classifier generative adversarial network, which includes a discrete feature generation module. Finally, a novel Stacking ensemble classifier for SM fault diagnosis is developed. In contrast to previous studies, we construct a two-layer heuristic optimization framework to address the synchronous dynamic optimization problem of the combinations and hyperparameters of the Stacking ensemble classifier, enabling better handling of complex classification tasks using SM data. The proposed fault diagnosis method for SM via two-layer stacking ensemble optimization and data augmentation is trained and validated using SM fault data collected from 2010 to 2018 in Zhejiang Province, China. Experimental results demonstrate the effectiveness of the proposed method in improving the accuracy of SM fault diagnosis, particularly for minority classes.**

***Index Terms*—Data augmentation, fault diagnosis, feature extraction, smart meter, Stacking ensemble optimization.**

## I. Introduction

SMART meters (SMs), communication networks, and data management systems form an advanced metering infrastructure that plays a vital role in information integration, tariff improvement, and energy management [1]. In recent years, the global SM market has witnessed significant growth owing to technological advancements and the implementation of carbon peaking and carbon neutrality targets. According to statistics, the global SM market will reach 40.45 billion dollars by 2030 [2], resulting in increased operation and maintenance (O&M) work. The increasing number of manufacturers supplying terminal devices, coupled with the influence of multiple modalities such as manufacturing processes, components, and installation environments, has contributed to complex and diverse fault types. These faults pose significant threats to the long-term stability of intelligent industrial systems. Manual experience alone has proven insufficient for the rapid and accurate identification of SM fault types [3]. Furthermore, current research works on SM fault diagnosis are limited because of the challenges in obtaining fault data from SMs. Therefore, developing a robust SM fault diagnosis model holds immense significance, as it can assist O&M personnel in efficiently and accurately identifying fault types and ensuring the stable operation of the power system.

Deep learning is currently the leading method for diagnosing smart device faults. This process involves learning intricate mapping relationships from large volumes of data. However, this methodology is often hindered by missing data, redundant features, and imbalanced data distributions among different fault categories [4]. Considering these data quality issues, this study aims to address the specific challenges associated with industrial equipment fault diagnosis by focusing on the SM fault diagnosis. The objective is to enhance the classification performance of SM faults.

Methods to overcome feature redundancy include feature extraction and selection, which can improve the generalization ability and performance of the model. Feature selection may disrupt the correlation between certain features, leading to the loss of important information. In contrast, feature ex-

Manuscript received: November 20, 2023; revised: December 19, 2023; accepted: January 22, 2024. Date of CrossCheck: January 22, 2024. Date of online publication: February 12, 2024.

This work was supported by the National Key R&D Program of China (No. 2022YFB2403800), the National Natural Science Foundation of China (No. 52277118), the Natural Science Foundation of Tianjin (No. 22JCZDJC00660), and the Open Fund in the State Key Laboratory of Alternate Electrical Power System With Renewable Energy Sources (No. LAPS23018).



L. Ge (corresponding author), T. Du, and Z. Xu are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: legendglj99@tju.edu.cn; tjudts@126.com; xuzhengyang@tju.edu.cn).

L. Hou is with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: luyang.hou@bupt.edu.cn).

J. Yan and Y. Li are with Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: jun.yan@concordia.cn; yuanliang.li@ericsson.comins).

DOI: 10.35833/MPCE.2023.000909

traction can better preserve information in the original data by mapping the original features to a new feature space. In this study, we encounter the following difficulties. The first is to obtain more discriminative features during feature extraction. Several studies have introduced feature extraction methods for fault diagnosis to capture general and essential information from extensive data and achieve feature representations with enhanced generalization capabilities [5]. Deep-learning-based feature extraction techniques, particularly those utilizing autoencoders (AEs), have shown significant promise for automatically learning complex features from large-scale data for industrial equipment fault diagnosis [6]. Although only a few studies have specifically focused on SM fault diagnosis, insights can be drawn from research works on other smart devices. Previous research works [6] and [7] have utilized the encoder part of an AE as a feature extractor for industrial equipment fault information, addressing issues related to data noise and high feature dimensionality. However, without any constraints on the intermediate layer nodes, an AE might learn redundant information from the data. To address this issue, some scholars have introduced sparsity constraints to guide AEs to capture crucial features and improve the accuracy of fault identification in smart devices [8]. The effectiveness of these methods has been extensively demonstrated in [8]-[10]. The feature extraction methods described above focus on obtaining key downscaled data representations to improve the efficiency of most machine-learning models in processing data. However, this broad effect does not necessarily apply to all supervised tasks [4]. In other words, the aforementioned feature extraction methods are not specifically designed for classification problems. For example, these methods do not fully leverage the label information of samples, resulting in a waste of information resources and a lack of discriminative features in the extracted representations [11]. Consequently, obtaining more discriminative fault feature information for SM fault diagnosis is a significant challenge that must be addressed.

The second challenge is the classification bias caused by an imbalance between fault categories. The occurrence frequencies of different SM faults exhibit significant variations in practical applications, resulting in severe category imbalance. This poses challenges for classification models, as they tend to be more sensitive to the majority classes of samples, leading to poor generalization ability. To address the issue of catogary imbalance, various techniques focus on the original dataset and employ oversampling (generating synthetic minority samples) or undersampling (removing majority classes of samples) based on the sample characteristics. Typical undersampling methods include random and clustered undersampling methods [12]. However, removing data can result in the loss of important information. Traditional oversampling methods such as random oversampling (ROS), synthetic minority class oversampling (called SMOTE), and borderline SMOTE [13] are easy to implement, but fail to learn the underlying data distribution and are prone to synthesizing noisy samples, increasing the risk of overfitting [14].

Recently, generative adversarial networks (GANs) have emerged as powerful tools for learning the latent distribution of data through competitive training, and have been utilized to address the imbalance problem of fault diagnosis in smart devices [15]. However, GANs face challenges in generating samples of specific classes, making it difficult to adapt them for the diagnosis of multiple fault types (12 in this study) in SMs. References [16] and [17] propose a fault data expansion method based on an auxiliary classifier GAN (ACGAN) that can generate samples of specified classes and has demonstrated its effectiveness. Unfortunately, SM fault data often contain discrete features such as manufacturers, SM models, affiliated departments, and city companies, making it challenging for the classical ACGAN to satisfy the augmentation demands of SM data.

The final challenge is the design of a high-precision SM fault diagnosis classifier. SM fault diagnosis can be considered a classification problem, where the accuracy of fault identification relies on the performance of the classifier. However, as the dataset grows larger, it becomes increasingly challenging for a single classifier to obtain accurate decision boundaries [18], [19]. To address this issue, researchers have explored the use of ensemble classification models to improve performance [18]. The motivation behind this is to reduce the variance and bias, thereby reducing the dependence of the results on the characteristics of a single training set, and the combination of multiple classifiers can learn a model that is more expressive than a single classifier. A notable study on SM fault diagnosis [3] uses random forests to demonstrate the effectiveness of integrated learning in this context. However, traditional ensemble learning methods such as bagging and boosting typically rely on voting or weighted voting to combine homogeneous learners, making it difficult to leverage the advantages of heterogeneous models [19]. The Stacking ensemble strategy offers a novel method for fusing heterogeneous models, providing better flexibility, generalization ability, and the capacity to tackle complex practical problems [20]. Previous research works have highlighted the superiority of Stacking integration strategies for fault diagnosis in smart devices [19], [21]. However, most research works have overlooked the optimal configuration of Stacking ensemble learners in the vast optional model space, assuming a specific model configuration [19] - [22]. Better combinations of base and meta models can improve the performance of integrated models by complementing their strengths, thus making them better suited to the requirements of a particular task. This necessitates investigating the performance differences when selecting the base and meta-classifiers. Furthermore, each heterogeneous model possesses a unique set of hyperparameters that significantly affect its performance. Improper hyperparameter settings can render the optimized Stacking configuration less appropriate. Incorporating the synchronous dynamic optimization of the hyperparameters of heterogeneous models during the optimal configuration further complicates this task. Therefore, the primary challenge lies in developing a Stacking ensemble classifier specifically tailored for SM fault diagnosis.

To address the aforementioned challenges, this study proposes an SM fault diagnosis framework that incorporates techniques to tackle sample imbalance and employs a two-layer Stacking ensemble optimization and data augmentation method. The key contributions of this study are as follows.

1) A novel SM fault diagnosis framework is developed to address the challenges of standardization, data imputation, and feature extraction. The framework incorporates a *K*-nearest neighbor (KNN) sequential imputation method that utilizes mutual information degree measurements to achieve a high-quality filling of missing features. In addition, a supervised fine-tuning process guided by category labels is integrated into the AE to enhance the extraction of discriminative feature information from the SM data.

2) A novel data augmentation method is proposed specifically for SM fault data. The method utilizes an ACGAN to augment SM fault data. Furthermore, a discrete feature generation module is introduced to alleviate the imbalance among different SM fault categories and enhance the diversity and representation of the augmented data.

3) A two-layer optimization configuration model is constructed based on a heuristic optimization algorithm for the Stacking ensemble strategy for SM fault diagnosis. The upper-layer optimization focuses on obtaining the optimal configuration of the base classifier and meta-classifier, whereas the lower-layer optimization dynamically optimizes the hyperparameters of the heterogeneous model based on the model configuration.

4) An in-depth analysis of massive real-world SM fault data in a comprehensive case study is presented in this study, which serves as a crucial theoretical and experimental foundation for further advancements in the research of SM fault diagnosis. The experimental results show that the proposed fault diagnosis framework can effectively improve the accuracy of SM fault diagnosis. In particular, the proposed data augmentation method improves the classification accuracy of minority-class faults by 5.86% on average.

The remainder of this study is organized as follows. Section II describes the proposed SM fault diagnosis framework. Section III presents the feature engineering. Section IV presents the SM data augmentation. Section V develops the two-layer optimized Stacking ensemble classifier for SM fault diagnosis. Section VI presents a simulation analysis with actual SM fault data. Finally, Section VII concludes this study.

## II. SM Fault Diagnosis Framework

The primary objective of SM fault diagnosis is to achieve an optimal comprehensive classification accuracy. However, the accuracy of the diagnosis is influenced by multiple factors that can complicate and hinder the training process of the classifier. To improve the accuracy of SM fault diagnosis, we propose a novel method that combines several techniques such as sequential imputation and discriminative enhanced sparse autoencoder (DESAE), a Gumbel-softmax-based ACGAN (GS-ACGAN) data enhancement technique, and a Stacking ensemble classifier, considering the optimization configuration.

Figure 1 presents the diagram of SM fault diagnosis framework, which includes feature engineering, data augmentation, and SM fault diagnosis. The feature engineering module aims to obtain high-quality input data. Standardization plays a crucial role in the overall framework, as it addresses issues such as computational bias in the nearest neighbor calculations and convergence difficulties in gradient descent caused by varying magnitudes [20]. After standardization, the next step involves data imputation, which can effectively handle missing values during training. The proposed KNN sequential data imputation technique considers the information gain provided by the feature correlation and estimation order. Subsequently, the complete dataset is input to DESAE to extract more discriminative feature information. To address the challenge of catogary imbalance, the complete dataset is fed into a catogary balancer called the GS-ACGAN. This balancer learns the distribution of the SM fault data, which includes both discrete and continuous features, and effectively tackles the class imbalance problem. Once the data are prepared, they are input into a Stacking ensemble model for classification. The performance of the Stacking ensemble model relies on the configuration of both the base and meta-models. To optimize this configuration and its corresponding hyperparameters, a two-layer whale optimization algorithm (WOA) is developed. The upper-layer optimization provides a combination of base and meta-models, whereas the lower-layer optimization adjusts the corresponding hyperparameters according to the *K*-fold cross-validation scores. In addition, to overcome the limitations of traditional heuristics in continuous numerical optimization, we introduce a time-varying binary transfer function that updates the position of the search agent. To accelerate the hyperparameter optimization process, an external archive repository is introduced to store the historical optimal hyperparameters for each classifier. These modules are further elaborated in the subsequent sections.

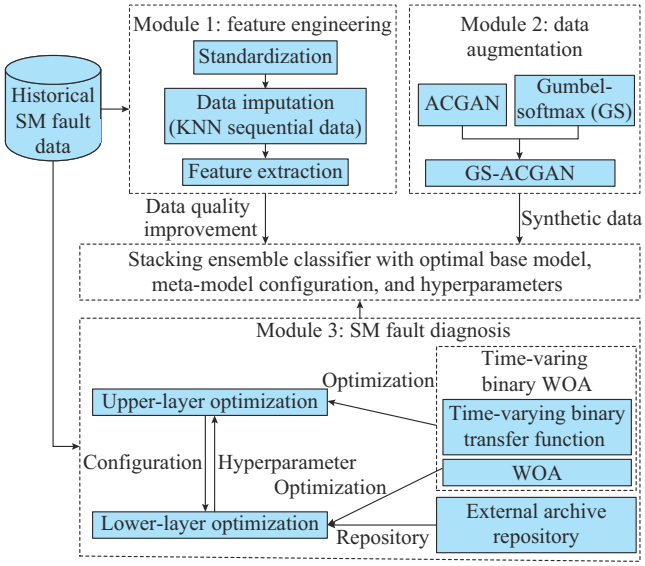


Fig. 1. Diagram of SM fault diagnosis framework.

## III. Feature Engineering

Feature engineering plays a crucial role in analyzing big-data problems, which is an initial and essential step [4]. By employing techniques such as data recovery, noise removal, and extraction of relevant information, feature engineering can enhance the quality of the training data. Feature engineering is particularly important in the context of the SM fault diagnosis, which focuses on three key technical aspects of feature engineering tailored to the data characteristics of

SMs: standardization, data imputation, and feature extraction.

### A. Standardization

Owing to the varying magnitudes and orders of magnitude of SM fault features such as the average daily timing error and average monthly electricity consumption, the continuous features need to be standardized. By applying standardization, multidimensional features can be transformed into values of similar scales. This step not only enhances the reliability of the subsequent data imputation but also facilitates faster convergence of the gradient descent algorithm. Standardization is employed as a preprocessing technique to ensure consistent data treatment, which is outlined as:

$$x^{sd}=\frac{x-\bar{x}}{\sigma(x)} \tag{1}$$

where $x^{sd}$ is the standardized data; $x$ is the sample of the original data for a feature; $\bar{x}$ is the mean of the sample; and $\sigma(x)$ is the standard deviation of the sample.

### B. Data Imputation

SM fault diagnosis involves mapping the feature information of fault samples to specific fault types. However, SM fault samples often contain discrete data and missing multivariate information, which can negatively affect the performance of machine-learning models. Although traditional linear interpolation methods are suitable for partially missing time-series data, they are not ideal for independent SM fault samples. Mindlessly filling missing values with zeros can lead to redundant feature information. Previous studies have shown that interpolating the missing features of instances in a specific order yields better results than directly interpolating all the missing values. In addition, estimating a missing feature becomes more significant when there is a stronger relationship between the feature and the corresponding fault category [23]. Considering the characteristics of SM data, we adopt the mutual information degree $I$ to assess the correlation between feature $F$ and the category label $Y$.

For discrete features, we use the following equation:

$$I(F,Y)=\sum_{f\in F}\sum_{y\in Y}p(f,y)\lg\frac{p(f,y)}{p(f)p(y)} \tag{2}$$

where $p(\cdot)$ is the probability density function.

For continuous features, we use the following equation:

$$I(F,Y)=\sum_{y\in Y}\int p(f,y)\lg\frac{p(f,y)}{p(f)p(y)}\mathrm{d}f \tag{3}$$

Features with stronger correlations will be interpolated earlier in this study.

To address the varying degrees of missingness across instances, we introduce the missing rate for each instance as a secondary criterion. This study offers the advantage of maximizing the utilization of the available information. The missing rate of the $m^{\text{th}}$ instance is:

$$R_m=\frac{N_m}{N_F} \tag{4}$$

where $N_F$ is the total number of features; and $N_m$ is the number of missing features in an instance. An instance with a lower missing rate is first interpolated.

The KNN algorithm is a widely used nonparametric imputation method that has proven to be effective in various estimation tasks [24]. It provides highly accurate imputation data suitable for both continuous and discrete values. Given these benefits, KNN is selected as the estimator for sequential imputation in this study.

To estimate the missing features, for the instance $x_p$, the Manhattan distance between $x_p$ and each remaining instance $x_q$ in the dataset is calculated using (5). This method estimates the distances for continuous and discrete features as:

$$dist(x_p,x_q)=\sum_{i=1}^{N_c}\left|x_{pi}-x_{qi}\right| \tag{5}$$

where $x_{pi}$ is the $i^{\text{th}}$ feature of the missing instances; and $x_{qi}$ is the $i^{\text{th}}$ feature of the remaining instances. The distances are sorted in ascending order, and the first $K_N$ are selected as fill candidates, where $K_N$ is an important hyperparameter in KNN. Notably, once an instance is fully populated, it is considered as a complete instance and participates in the calculation. For continuous features, the mean of the first $K_N$ instances is used for filling as:

$$x_{pj}=\frac{\sum_{k=1}^{K_N}x_{kj}}{k} \tag{6}$$

Discrete features use the voting results of the first $K_N$ instances:

$$x_{pj}=\arg\max\left(count\left(x_{kj}\right)\right) \tag{7}$$

where $count\left(x_{kj}\right)$ is the count result of the $j^{\text{th}}$ feature for all the candidate instances. $K_N$ is set to be 9 in this study with reference to existing studies [23], [24] and experimental results.

### C. Feature Extraction

The AE consists of two primary phases: encoding and decoding, as shown in (8) and (9). In the encoding phase, the AE utilizes weighting and biasing operations to transform input data into a compressed feature representation. This process, known as feature downscaling, has been demonstrated to be effective [6], [7].

$$H=f_1\left(W_1X+b_1\right) \tag{8}$$

$$X^R=f_2\left(W_2H+b_2\right) \tag{9}$$

where $W_1$ and $b_1$ are the weight and deviation from the input layer to the hidden layer, respectively; $W_2$ and $b_2$ are the weight and deviation from the hidden layer to the output layer, respectively; $X$, $H$, and $X^R$ are the original input, hidden layer output, and reconstructed data, respectively; and $f_1(\cdot)$ and $f_2(\cdot)$ are the sigmoid activation functions.

Although the AE can learn features from input data, the learned features may not necessarily capture the critical aspects of the samples [11]. To address this limitation, sparsity constraints can be incorporated into an AE. By imposing sparsity constraints, redundant information in the input data can be filtered out, allowing the AE to focus more effectively on the critical features. After incorporating the sparsity constraint, the loss function of the AE can be expressed as:

$$J_{SAE}(W,b)=\left\| X^R-X \right\|^2+\beta\sum_v KL\left(\rho\,//\,\hat{\rho}_v\right) \tag{10}$$

$$KL\left(\rho\,//\,\hat{\rho}_v\right)=\rho\lg\frac{\rho}{\hat{\rho}_v}+\left(1-\rho\right)\lg\frac{1-\rho}{1-\hat{\rho}_v} \tag{11}$$

where $J_{SAE}(W,b)$ is the loss function after the sparsity constraint is introduced; $\beta$ is the weight coefficient of Kullback-Leibler (KL) divergence $KL(\cdot)$; $\rho$ is the sparsity parameter of KL divergence; and $\hat{\rho}_v$ is the average activation degree of node $v$ in the hidden layer.

The ability of classifiers to accurately identify fault categories relies on the discriminative power of the features, which has received considerable attention from researchers [7]. In this study, we introduce a supervised training process for SAE called the DESAE. This additional training process incorporates category labels to guide the learning of more discriminative features.

Clustering algorithms are widely employed to analyze feature similarities and serve as valuable tools for understanding the characteristics of mixed data types in SM fault diagnosis [25]. In this study, we utilize the *k*-prototype clustering algorithm [25] to cluster the implicit feature representation $H$ obtained from the encoder into $N_K$ clusters. The aim is to analyze the differentiation between the different categories of features. To ensure an accurate measure of feature distinctiveness, we set the number of clusters $N_K$ equal to the number of SM fault categories $N_C$. Incremental clustering is adopted during the training to mitigate the computational burden associated with repeated clustering, which uses the previous clustering results as the initial state for subsequent clustering iterations.

Our objective is to minimize the number of mixed categories within each cluster to maximize the distinguishability of features between different fault categories. Ideally, each cluster should contain only one fault category. However, achieving this goal is challenging. To represent the fault categories contained in different clusters, we define a matrix $\boldsymbol{B}=\left(b_{ij}\right)$ of size $N_C\times N_K$. To describe the affiliation of fault categories with clusters, we introduce an affiliation matrix $\boldsymbol{U}=\left(u_{ij}\right)$ of the same size as $\boldsymbol{B}$. The value of $u_{ij}$ is determined using (12).

$$u_{ij}=\begin{cases}0 & b_{ij}<T_i\\ 1 & b_{ij}\geq T_i\end{cases} \tag{12}$$

where $b_{ij}$ is the number of samples in the $i^{\text{th}}$ fault category of the $j^{\text{th}}$ cluster; and $T_i$ is the threshold for the $i^{\text{th}}$ fault category belonging to the $j^{\text{th}}$ cluster, which is set to be $N_C^i/N_K$ in this model, and $N_C^i$ is the number of samples in the $i^{\text{th}}$ fault category. Exceeding the threshold indicates that the category belongs to the $j^{\text{th}}$ cluster.

We define function $D$ to quantify the degree of feature differentiation. This function measures the dissimilarity between features and is then multiplied by a scaling factor $\gamma$. The resulting value is incorporated into the loss function formula for the supervised training process of the stacked sparse AE (SSAE). During the training of the SSAE, the parameters of the network structure are fine-tuned based on this loss function. Note that both the reconstruction loss and feature discriminative loss play a combined role in guiding the model to acquire key and discriminative features, which is aligned with the idea of the regularization term.

$$D=\left\|\sum_{i=1}^{N_C}\sum_{j=1}^{N_K}u_{ij}-N_C\right\| \tag{13}$$

$$J_{DESAE}(W,b)=J_{SAE}(W,b)+\gamma D \tag{14}$$

where $J_{DESAE}(W,b)$ is the loss function of DESAE.

## IV. SM Data Augmentation

Although feature engineering aims to enhance the quality of raw data, it often struggles to address the significant classification bias originating from data imbalance. To address this challenge, a novel method called GS-ACGAN is developed and applied in the context of SM fault diagnosis. The GS-ACGAN is utilized to generate fault samples specifically for SMs, thereby alleviating the data-imbalance problem.

### A. Principle of ACGAN

A GAN is a game theoretically inspired architecture consisting of a generator G and a discriminator D. The generator captures the latent distribution of the real data samples and generates data from Gaussian random noise vectors to "trick" the discriminator. In contrast, the discriminator is trained to distinguish between real and generated data. The GAN alternately trains G and D, leveraging the concept of a game to reach the Nash equilibrium.

The ACGAN is a variation of the GAN architecture that adds auxiliary classifiers to the GAN to address the limitations of the traditional GAN in generating specified categories. The ACGAN introduces labeling information during the training of G and generates samples of the specified categories $X_{fake}=G(z,c)$ using the random noise $z$ and the labeling information $c$. The probability distributions of the source and category labels $P\left(DS|X\right)$ and $P\left(C|X\right)$, respectively, are then given by D. The loss function of the ACGAN is composed of two parts: the loss function of the source and the label of the category, and the loss function of the label of the category. The two components of the loss function of the ACGAN are expressed as:

$$L_S=E_{x\sim P_{data}}\left(\lg P\left(DS=real|X_{real}\right)\right)+E_{z\sim P_z}\left(\lg P\left(DS=fake|X_{fake}\right)\right) \tag{15}$$

$$L_C=E_{x\sim P_{data}}\left(\lg P\left(C=c|X_{real}\right)\right)+E_{z\sim P_z}\left(\lg P\left(C=c|X_{fake}\right)\right) \tag{16}$$

where $E_{x\sim P_{data}}$ and $E_{z\sim P_z}$ are the expectations over the real data and noise distributions, respectively; $L_S$ is the probability of correct judgment when judging the authenticity of the data; $L_C$ is the probability of correctly classifying data; $X_{real}$ and $X_{fake}$ are the real and synthetic samples, respectively; and $DS$ and $C$ are the data sources and label categories, respectively. During the training process, the loss function of the discriminator is $L_S+L_C$ and the loss function of the generator is $L_C-L_S$.

### B. GS-ACGAN

The ACGAN is primarily designed for real data values, which poses challenges when dealing with feature variables

in the case of SMs consisting of both continuous and discrete variables. Using discrete data as the input can effectively undermine the ability of the discriminator to learn distributional features. Furthermore, when the generator produces probability vectors for discrete variables, the direct sampling of the maximum can terminate the training of the generator early, resulting in a loss of exploratory ability [26].

In this study, we first employ the Gumbel-max trick to sample the discrete distributions. Assuming that the generator produces a probability set $P=\left\{p_\zeta|\zeta=1,2,\ldots,N_C\right\}$ of discrete variables, the corresponding Gumbel-max representation is given by:

$$x_c=\underset{i\in\{1,2,\ldots,N_C\}}{\arg\max}\left(\lg\left(p_\zeta+g_\zeta\right)\right) \tag{17}$$

$$g_\zeta=-\lg\left(-\lg\left(u_\zeta\right)\right) \tag{18}$$

where $u_\zeta\sim U(0,1)$; and $g_\zeta$ is the noise value that satisfies the Gumbel distribution.

GS is used to address the nondifferentiability of the argmax operation. GS facilitates the sampling of discrete variables while maintaining the ability to compute gradients through continuous operations, which enables the model to be optimized using gradient-based methods. The discrete variable results can be obtained using the following equation.

$$\tilde{x}_c=\frac{\exp\left(\lg\left(p_\zeta+g_\zeta\right)/T\right)}{\sum_{\zeta=1}^{N_C}\exp\left(\lg\left(p_\zeta+g_\zeta\right)/T\right)} \tag{19}$$

where $T\in(0,\infty)$ is a temperature hyperparameter used to control the likelihood that the generated vectors are one-hot. $\tilde{x}_c$ approximates the one-hot vectors as $T$ approaches zero. To simulate the annealing process, $T$ is relatively large at the beginning of the generator training and then gradually decreases to stabilize the generator performance.

## V. Two-layer Optimized Stacking Ensemble Classifier for SM Fault Diagnosis

Although the aforementioned techniques can enhance the quality and diversity of input samples for the classifier, the performance of the classifier remains the primary factor influencing the accuracy [27]. Therefore, in this section, we propose a heterogeneous model ensemble classifier using a Stacking strategy to address the classification of SM faults effectively. Furthermore, we explore the optimal configurations of the base model, meta-model, and hyperparameters.

### *A. Stacking Ensemble Classification Model*

The fundamental concept of Stacking involves the construction of two layers of classification models: base classifiers and meta-classifiers. The base classifiers classify the data to be diagnosed and obtain the classification results as inputs for the meta-classifiers. The base classifiers independently classify the data and generate their respective classification outcomes. The outcomes from multiple base classifiers are then utilized as inputs to the meta-classifiers. In turn, the meta-classifiers learn the mapping relationship between the base classifier outputs and the actual SM fault categories.

### *B. Heterogeneous Model Optimization*

#### *1) Problem Description*

As Stacking involves the integration of different models, the selection of classifiers can significantly affect the performance, which can vary across datasets and problems. Therefore, it is necessary to approach this selection as a preference rather than following a standardized configuration. To address this, we transform the heterogeneous model selection into an optimization problem with decision variables denoted as $S=\left[s_1,s_2,\ldots,s_{N_S},s_{N_S+1}\right]$, where $s_1$-$s_{N_S}$ are the binary variables, and the value of which indicates the selection of a model as a base classifier. In contrast, $s_{N_S+1}$ is an integer ranging from 1 to $N_S$, indicating the selection of meta-classifiers. Because of the large search space involved, exhaustive search methods are often impractical because they require the evaluation of all possible solutions. Instead, heuristic algorithms employ heuristic information to guide the search process. These algorithms assess the potential value or superiority of each candidate solution in the search space, leading to a smaller search space and faster convergence than exhaustive search methods [28].

However, a new challenge arises when the Stacking configuration is optimized. The number and types of hyperparameters vary across different combinations of heterogeneous models, making it challenging to optimize both heterogeneous model combinations and hyperparameters simultaneously [29], [30]. To address this issue, we propose a two-layer stacking ensemble optimization algorithm. The upper layer focuses on determining the base model and meta-model configurations, whereas the lower layer provides feedback based on the provided model configurations, enabling the optimization of the corresponding hyperparameters.

#### *2) Objective Function*

*k*-fold cross-validation enables each fold of a dataset to be used for both training and validation, which maximizes the utilization of all samples and effectively addresses issues such as model overfitting and inaccurate evaluation caused by limited validation data. To evaluate the performance of the Stacking framework, we adopt the average *K*-fold cross-validation score of the training set as an objective function:

$$\min f(x)=-\frac{1}{K_V}\sum_{\mu=1}^{K_V}\frac{1}{N_C}\sum_{c=1}^{N_C}2\frac{PR_c^\mu\cdot RE_c^\mu}{PR_c^\mu+RE_c^\mu} \tag{20}$$

$$P_c=\frac{TP_c}{TP_c+FP_c} \tag{21}$$

$$R_c=\frac{TP_c}{TP_c+FN_c} \tag{22}$$

where $K_V$ is the number of folds for *K*-fold cross-validation, which is set to be 5 in this study; $PR_c^\mu$ and $RE_c^\mu$ are the recognition accuracy and recall of the $c^{\text{th}}$ fault category when the first fold is used as the validation set, respectively; and $TP_c$, $FP_c$, and $FN_c$ are the true-positive, false-positive, and false-negative examples of the $c^{\text{th}}$ fault category, respectively.

#### *3) Two-layer Stacking Ensemble Optimization Algorithm*

The WOA is a heuristic optimization algorithm that has attracted attention in recent years. Inspired by the behavior of

humpback whales, the WOA mimics their foraging and bubble-net hunting techniques [31]-[33]. It offers several advantages, including fast convergence and robust search capabilities. In the context of optimization problems, the WOA has demonstrated a superior search performance compared with various classical optimization algorithms such as particle swarm optimization, genetic algorithm, differential evolution, and ray optimization. The WOA has gained wide acceptance and application in diverse domains [34]. Given the efficacy of the WOA in optimization tasks, we construct a two-layer optimization configuration model utilizing the WOA algorithm.

In the WOA, whale behavior is divided into two phases: exploitation and exploration. The exploitation phase first updates the individual position by encircling the prey and approaching the best search agent using the following position calculation formula:

$$D_{WOA}=\left|CX_t^B-X_t\right| \tag{23}$$

$$X_{t+1}=X_t^B-AD_{WOA} \tag{24}$$

where the subscript $t$ represents the number of iterations; $D_{WOA}$ is the distance between the current individual whale and the optimal solution; $X_t^B$ is the optimal position for the current number of iterations; $X_t$ is the current position of an individual whale; and $A$ and $C$ are the position update coefficients.

As the whale swims along a spiral-shaped path while encircling its prey, a probability of 50% is set to choose between the two behaviors randomly, thus enabling an update of the position of the whale [34].

$$X_{t+1}=\begin{cases}X_t^B-\mathrm{A}D^{WOA} & r_1<0.5\\ \mathrm{e}D_{WOA}\cos(2\pi l)+X_t^B & r_1\geq 0.5\end{cases} \tag{25}$$

where $r_1$ is a random number from 0 to 1.

Figure 2 shows the flowchart of two-layer Stacking ensemble optimization. The upper-layer optimization involves the selection of heterogeneous models with the decision variable denoted as $S$=[1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 5], which indicates that classification models 1, 3, 6, and 7 are chosen as the base models, whereas classification model 5 is selected as the meta-model. In the lower-layer optimization, the hyperparameters of models 1, 3, 5, 6, and 7 become the decision variables for optimization. An external archive is constructed to expedite the heuristic algorithm in determining the optimal hyperparameters. This archive records the hyperparameters associated with the historical optimal objective function for each classification model. During the optimization process in the lower layer, instead of generating initial values randomly, the external archive is referenced to obtain the initial values for the hyperparameters of each classification model.

Note that the upper-layer optimization involves binary variables, and traditional continuous numerical optimization makes it difficult to find the optimal solution in discrete space [35]. Hence, we introduce a binary time-varying transfer function $TV\left(XM,\varphi\right)$ as:

$$TV\left(XM,\varphi\right)=\frac{1}{1+\mathrm{e}^{-\frac{XM}{\varphi}}} \tag{26}$$

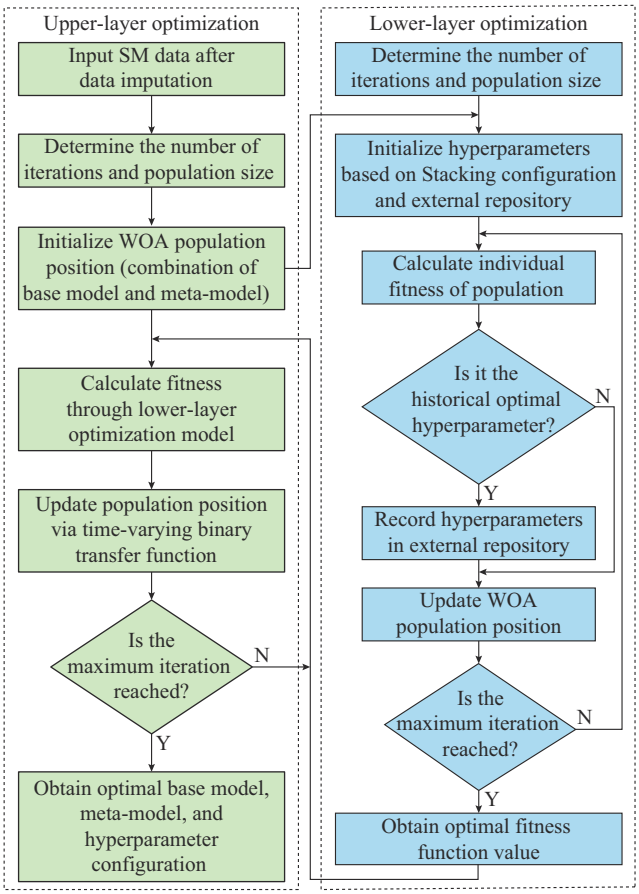


Fig. 2. Flowchart of two-layer Stacking ensemble optimization.

where $XM$ is the position of an individual heuristic algorithm; and $\varphi=\varphi_{\max}-t\left(\frac{\varphi_{\max}-\varphi_{\min}}{t_{\max}}\right)$ is the control parameter, and $t_{\max}$ is the maximum iteration, and $\varphi_{\max}$ and $\varphi_{\min}$ are the upper and lower limits of the control parameter, which are set to be 4 and 0.05, respectively.

Thus, the locations of the decision variables of the upper-layer can be transformed as:

$$XM_m^d(t+1)=\begin{cases}1 & rand<TV\left(XM_m^d(t),\varphi\right)\\ 0 & rand\geq TV\left(XM_m^d(t),\varphi\right)\end{cases} \tag{27}$$

where $XM_m^d$ is the $d^{\text{th}}$ decision variable of the $m^{\text{th}}$ search agent; and *rand* is a random number ranging from 0 to 1.

## VI. Case Study

The difficulty in obtaining samples is an essential reason for the lack of research on SM fault diagnosis. Fortunately, we obtain nearly 200000 SM fault samples from Zhejiang Province, China, for analysis. Therefore, this case study utilizes the data obtained from SM fault samples in the urban areas of the Zhejiang Province from 2010 to 2018, encompassing 11 urban areas. After data processing, the dataset comprises 15 features and 12 fault categories. To enhance the data quality, we calculate the feature "usage duration" by subtracting the fault date from the installation date, effectively replacing the two dates. Consequently, the SM fea-

tures encompass meter manufacturer (feature 1), meter model (feature 2), department (feature 3), city company (feature 4), county company (feature 5), communication mode (feature 6), meter reading data (feature 7), average monthly consumption in the last three months (feature 8), average basic error (feature 9), average daily timing error (feature 10), number of the same batch (feature 11), length of use (feature 12), arrival batch number (feature 13), tender lot number (feature 14), and production lot number (feature 15). The data types for SM features are presented in Table I.

TABLE I
DATA TYPES OF SM FEATURES

| Data type | Feature No. |
|---|---|
| Discrete type | 1, 2, 3, 4, 5, 6 |
| Numerical type | 7, 8, 9, 10, 11, 12, 13, 14, 15 |

Failure samples (fault categories) from the most to the least are as follows: clock battery undervoltage (category 7), clock misalignment (category 6), timing function abnormality (category 2), broken appearance (category 9), communication interface failure (category 3), measurement function abnormality (category 11), stop reading battery undervoltage (category 5), preset parameter error (category 10), cost control function abnormality (category 1), LCD data display abnormality (category 12), current change caused by the percentage of error (category 4), and other faults (category 8). Table II lists the sample size of each fault category, highlighting the significant imbalance among the SM fault categories.

TABLE II
SAMPLE SIZE OF EACH FAULT CATEGORY

| Category | Sample size | Imbalance ratio | Category | Sample size | Imbalance ratio |
|---|---|---|---|---|---|
| 1 | 3849 | 10.96 | 7 | 53157 | 1.00 |
| 2 | 33113 | 1.61 | 8 | 2203 | 16.60 |
| 3 | 7138 | 7.45 | 9 | 28761 | 1.85 |
| 4 | 2426 | 15.52 | 10 | 4767 | 9.22 |
| 5 | 4915 | 10.81 | 11 | 5213 | 8.56 |
| 6 | 46760 | 1.14 | 12 | 2468 | 15.33 |

The degree of imbalance for each fault category is quantified, as shown in (28). Categories 1, 3, 4, 5, 8, 10, 11, and 12, which exhibit imbalance rates exceeding five, are identified as minority categories for subsequent data augmentation purposes as:

$$p_i = \frac{N_{\max}}{N_C^i} \tag{28}$$

where $p_i$ is the imbalance rate of fault category $i$; and $N_{\max}$ is the maximum number of samples in the fault category.

In this section, we partition the dataset into training and test sets using a stratified sampling method, maintaining a ratio of 8 : 2. The entire SM fault diagnosis process is then simulated and analyzed to assess the effectiveness of sequential data imputation, feature extraction, data augmentation, and Stacking ensemble classifiers using the test set. The performance of the method is evaluated based on its accuracy and F1-score. Although the accuracy provides an intuitive measure of the classification performance of a method, it can be biased toward majority classes. To address this, the F1-score combines the accuracy and recall metrics, making it suitable for evaluating the classification performance in the presence of a class imbalance.

### *A. Analysis of Fault Diagnosis Results*

The input feature data are normalized to mitigate the impact of varying magnitudes and orders of magnitude among the SM fault features on the classification performance. Furthermore, discrete features are transformed into a one-hot code. In addition, missing values within each sample are addressed using the sequential imputation method outlined in Section III. In this experiment, we evaluate 11 efficient and widely used classifiers.

1) Traditional machine learning: support vector machine (SVM), decision tree (DT), KNN, and naive Bayes (NB).

2) Neural networks: backpropagation neural network (BPNN) and convolutional neural network (CNN).

3) Ensemble learning: random forest (RF), extreme gradient boosting machine (XGBoost), light gradient boosting machine (LightGBM), adaptive boosting (AdaBoost), and categorical boosting (CatBoost).

The optimized configuration results for stacking ensemble classifiers are presented in Table III. Ensemble classification models such as LightGBM, CatBoost, and XGBoost are selected more often, indicating a better classification performance. RF belongs to the category of bagging ensemble methods, and LightGBM, CatBoost, and XGBoost belong to the category of boosting ensemble methods. To assess the performance of the optimized model, we compare the F1-scores of the resulting Stacking ensemble classifier with those of other classifiers, as presented in Table IV. Data with the highest accuracy are highlighted in bold. Note that, during the two-layer optimization configuration, our external repository is responsible for recording the historical optimal hyperparameters of each model. Each model historically achieves the optimal performance under the set objective function. Therefore, the hyperparameters of the comparative model are all obtained from our external repository and are not set arbitrarily. Fault categories 1, 3, 4, 5, 8, 10, 11, and 12, characterized by a small number of samples, exhibit significantly lower recognition accuracies compared with fault categories 2, 6, 7, and 9, which have a larger number of samples. This indicates that the classification accuracy of the model decreases when there is a lack of instances in a particular category. The results demonstrate that the classifiers developed in this study offer an optimal classification performance for almost every category. The poor performance of the NB classifier in the experiments suggests that the assumption of disregarding feature correlations does not align with the SM fault data used in this study, further affirming the existence of complex interdependencies among the SM features.

TABLE III
OPTIMIZED CONFIGURATION RESULTS FOR STACKING ENSEMBLE CLASSIFIERS

| Optimization type | Classifier type | Optimization configuration |
|---|---|---|
| Upper-layer | Base | LightGBM, XGBoost, KNN, and CatBoost |
| | Meta | XGBoost |
| Lower-layer | LightGBM | learning_rate = 0.0641,<br>n_estimators = 554,<br>Num_leaves = 43,<br>min_child_sample = 15,<br>max_depth = 6,<br>reg_lambda = 0.209 |
| | XGBoost (base) | learning_rate = 0.0592,<br>n_estimators = 496,<br>Num_leaves = 39,<br>min_child_sample = 14,<br>max_depth = 6,<br>reg_lambda = 0.177 |
| | KNN | n_neighbors = 11,<br>algorithm = 'ball_tree',<br>leaf_size = 53,<br>weights = 'distance',<br>CatBoost: learning_rate = 0.0783,<br>iterations = 1042,<br>depth = 6,<br>Od_wait = 71 |
| | XGBoost (meta) | learning_rate = 0.0582,<br>n_estimators = 368,<br>Num_leaves = 31,<br>min_child_sample = 13,<br>max_depth = 5,<br>reg_lambda = 0.149 |

To demonstrate the importance of employing optimized configurations through Stacking, we compare the obtained classifiers with the configurations utilized in other fault diagnosis studies. The comparison of effectiveness of different missing value imputation methods is listed in Table V. The experiments are repeated 10 times for each method without setting a random number seed, and the average accuracy (macro-precision) and average F1-score (macro-F1-score) are computed, as shown in Fig. 3. Reference [22] adopts the simplest Stacking configuration, which makes it challenging to handle SM fault diagnosis tasks; therefore, it has the lowest accuracy. References [19] and [20] use Boosting ensemble learning and neural networks, but do not consider the optimization of the Stacking ensemble configuration, resulting in a lower accuracy than that of the proposed method. The proposed method significantly outperforms the other methods, indicating that Stacking configurations that exhibit a superior performance in other fault diagnosis domains may not necessarily be suitable for SM fault diagnosis. In other words, it is essential to optimize the configuration for a specific problem.

To assess the effectiveness of the data imputation proposed in Section III, we compare it with four other methods: missing value zero padding (Method I), sequential data imputation following KNN nonsequential data imputation (Method II), sequential data imputation using Pearson's correlation coefficient to evaluate feature relevance (Method III), and GANs (Method IV). The quality of the data imputation is evaluated based on the classification accuracy. The recognition accuracies of the different SM fault categories using the Stacking ensemble classifier as the medium are presented in Table VI.

Although easy to implement, the zero-padding method introduces a significant amount of redundant information, leading to a decrease in the fault recognition accuracy, particularly for some classes. GANs are data imputation methods based on deep learning. However, a mixture of discrete and continuous features makes it difficult for GANs to learn the distribution of data features, resulting in a poor data imputation performance. In contrast, the data-filling method proposed herein considers the information gain between features and classes, effectively addressing the information loss caused by missing values and yielding the best data imputation results.

TABLE IV
SM FAULT DIAGNOSIS ACCURACY WITH DIFFERENT CLASSIFIERS

| Category | Accuracy | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | RF | XGBoost | CNN | KNN | LightGBM | Stacking | SVM | CatBoost | DT | NB | AdaBoost |
| 1 | 0.382 | 0.523 | **0.583** | 0.409 | 0.497 | 0.571 | 0.506 | 0.448 | 0.390 | 0.188 | 0.361 |
| 2 | 0.714 | 0.779 | 0.639 | 0.742 | 0.790 | **0.806** | 0.732 | 0.740 | 0.709 | 0.501 | 0.707 |
| 3 | 0.557 | 0.731 | 0.610 | 0.654 | 0.743 | 0.743 | 0.656 | 0.679 | 0.562 | 0.514 | 0.629 |
| 4 | 0.381 | 0.412 | 0.272 | 0.373 | 0.351 | 0.437 | 0.228 | **0.461** | 0.369 | 0.270 | 0.313 |
| 5 | 0.407 | 0.590 | 0.642 | 0.555 | 0.561 | **0.672** | 0.633 | 0.657 | 0.396 | 0.228 | 0.573 |
| 6 | 0.693 | 0.682 | 0.629 | 0.665 | 0.699 | **0.736** | 0.608 | 0.623 | 0.694 | 0.415 | 0.514 |
| 7 | 0.757 | 0.767 | 0.685 | 0.775 | 0.783 | **0.807** | 0.616 | 0.713 | 0.754 | 0.554 | 0.659 |
| 8 | 0.506 | **0.645** | 0.357 | 0.581 | 0.617 | 0.628 | 0.267 | 0.409 | 0.419 | 0.218 | 0.506 |
| 9 | 0.914 | 0.877 | 0.803 | 0.847 | 0.891 | **0.925** | 0.741 | 0.821 | 0.893 | 0.592 | 0.815 |
| 10 | 0.454 | 0.528 | 0.410 | 0.546 | 0.506 | **0.572** | 0.319 | 0.519 | 0.421 | 0.218 | 0.460 |
| 11 | 0.425 | **0.579** | 0.549 | 0.488 | 0.533 | 0.577 | 0.395 | 0.558 | 0.335 | 0.481 | 0.524 |
| 12 | 0.272 | 0.238 | 0.133 | 0.233 | 0.231 | **0.358** | 0.126 | 0.385 | 0.272 | 0.127 | 0.254 |

TABLE V
CONFIGURATION OF STACKING ENSEMBLE CLASSIFIER FOR DIFFERENT REFERENCES

| Reference | Classifier type | Configuration of stacking ensemble classifier |
|---|---|---|
| [19] | Base | NB, BPNN, SVM, AdaBoost, LightGBM |
| | Meta | Logistic regression (LR) |
| [20] | Base | SVM, RF, gradient boosting decision tree (GBDT) |
| | Meta | Temporal convolutional network (TCN) |
| [21] | Base | KNN, AdaBoost, LR |
| | Meta | AdaBoost |
| [22] | Base | LR, NB, DT |
| | Meta | LR |

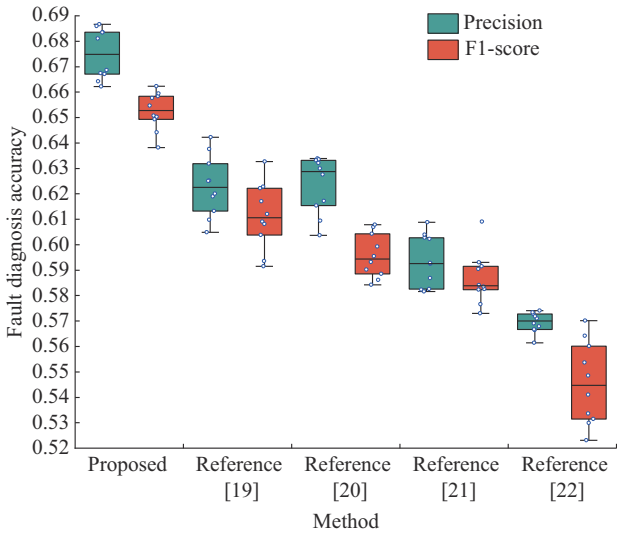


Fig. 3. Comparison of SM fault diagnosis accuracies of different methods.

Furthermore, the proposed method utilizes mutual information to quantify feature correlations by considering both numerical and discrete features. This method outperforms the Pearson correlation coefficient, which measures only linear correlations.

TABLE VI
RECOGNITION ACCURACY OF DIFFERENT SM FAULT CATEGORIES

| Category | Accuracy | | | | |
|---|---|---|---|---|---|
| | Method I | Method II | Method III | Method IV | Proposed |
| 1 | 0.553 | 0.561 | 0.566 | 0.557 | 0.571 |
| 2 | 0.794 | 0.792 | 0.802 | 0.793 | 0.806 |
| 3 | 0.726 | 0.739 | 0.737 | 0.731 | 0.743 |
| 4 | 0.402 | 0.425 | 0.431 | 0.416 | 0.437 |
| 5 | 0.644 | 0.653 | 0.669 | 0.663 | 0.672 |
| 6 | 0.713 | 0.722 | 0.728 | 0.734 | 0.736 |
| 7 | 0.792 | 0.808 | 0.805 | 0.796 | 0.807 |
| 8 | 0.614 | 0.616 | 0.626 | 0.611 | 0.628 |
| 9 | 0.911 | 0.913 | 0.917 | 0.908 | 0.925 |
| 10 | 0.542 | 0.560 | 0.566 | 0.556 | 0.572 |
| 11 | 0.539 | 0.574 | 0.575 | 0.542 | 0.577 |
| 12 | 0.307 | 0.342 | 0.359 | 0.331 | 0.358 |

### *B. Analysis of Feature Extraction Effect*

When converting discrete features from SM fault data into one-hot coding, the feature space expands, necessitating the use of efficient feature extraction methods to reduce redundancy. To validate the effectiveness of the DESAE feature extraction method proposed herein, the feature extraction process outlined in Section III is incorporated before classification. The results for each classifier are listed in Table VII. Compared with Table IV, the proposed feature extraction method enhances the generalization ability of the model, resulting in an improved classification accuracy. However, the improvement in the accuracy for the minority class is not significant. This is attributed to the limitations of the DESAE in learning the key features of the minority class when insufficient samples are available. In addition, the CNN inherently possesses a feature extraction function. Therefore, the classification accuracy does not improve significantly after implementing the proposed DESAE feature extraction method. This suggests that excessive feature extraction may result in the loss of crucial information.

TABLE VII
FAULT DIAGNOSIS ACCURACY OF EACH CLASSIFIER AFTER DESAE FEATURE EXTRACTION

| Category | Accuracy | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | RF | XGBoost | CNN | KNN | LightGBM | Stacking | SVM | CatBoost | DT | NB | AdaBoost |
| 1 | 0.384 | 0.546 | 0.581 | 0.413 | 0.515 | 0.569 | 0.512 | 0.453 | 0.415 | 0.198 | 0.372 |
| 2 | 0.722 | 0.781 | 0.629 | 0.766 | 0.793 | 0.814 | 0.739 | 0.740 | 0.717 | 0.527 | 0.731 |
| 3 | 0.563 | 0.760 | 0.607 | 0.650 | 0.768 | 0.749 | 0.661 | 0.705 | 0.590 | 0.517 | 0.658 |
| 4 | 0.387 | 0.434 | 0.268 | 0.392 | 0.354 | 0.449 | 0.232 | 0.488 | 0.386 | 0.298 | 0.316 |
| 5 | 0.423 | 0.616 | 0.652 | 0.557 | 0.560 | 0.672 | 0.657 | 0.722 | 0.415 | 0.257 | 0.582 |
| 6 | 0.713 | 0.698 | 0.628 | 0.673 | 0.703 | 0.754 | 0.621 | 0.623 | 0.701 | 0.422 | 0.513 |
| 7 | 0.782 | 0.795 | 0.687 | 0.782 | 0.784 | 0.829 | 0.643 | 0.722 | 0.774 | 0.580 | 0.673 |
| 8 | 0.529 | 0.671 | 0.361 | 0.598 | 0.624 | 0.639 | 0.294 | 0.435 | 0.431 | 0.225 | 0.537 |
| 9 | 0.908 | 0.902 | 0.809 | 0.868 | 0.901 | 0.923 | 0.748 | 0.827 | 0.917 | 0.616 | 0.825 |
| 10 | 0.455 | 0.541 | 0.407 | 0.561 | 0.524 | 0.574 | 0.338 | 0.543 | 0.446 | 0.245 | 0.473 |
| 11 | 0.423 | 0.582 | 0.546 | 0.499 | 0.542 | 0.581 | 0.426 | 0.562 | 0.338 | 0.499 | 0.549 |
| 12 | 0.276 | 0.240 | 0.132 | 0.236 | 0.241 | 0.357 | 0.129 | 0.399 | 0.282 | 0.127 | 0.262 |

To demonstrate the superiority of the proposed feature extraction method, DESAE is compared with AE, sparse autoencoder (SAE), and principal component analysis (PCA), and the fault diagnosis accuracy with different feature extraction methods is illustrated in Fig. 4. The DESAE effectively leverages the information gained from category labels and learns more discriminative feature information than the other feature extraction methods. Additionally, the feature extraction performance of the SAE surpasses that of the AE, indicating that incorporating sparsity constraints assists the AE in obtaining higher-quality and reduced-dimensional feature representations. Note that PCA is a linear transformation method that captures primarily linear relationships in the data. However, it struggles to achieve effective dimensionality reduction on SM data that contain nonlinear features. Consequently, not all feature extraction methods are effective in improving the accuracy of fault diagnosis. Therefore, efficient feature extraction methods tailored to specific problem domains need to be developed.

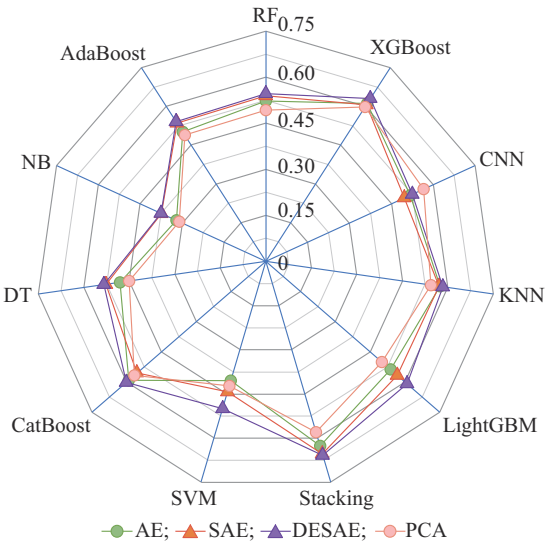


Fig. 4. Fault diagnosis accuracy with different feature extraction methods.

## C. Analysis of Data Augmentation Results

Data augmentation is a common practice in the field of fault diagnosis. However, directly increasing the number of minority classes of samples to match the majority classes of samples may lead to unreliable results. This is because synthesizing a large number of samples can introduce noise, which affects the recognition accuracy of the majority classes of samples. Based on the previous experiments, we conduct further investigations to determine the optimal amount of data augmentation based on the macro-F1-scores. The variation of SM fault diagnosis accuracy with different numbers of synthesized samples is shown in Fig. 5. A continuous decrease in the F1-score indicates that the recognition accuracy of the minority classes of samples is not significantly improved, which further affects the recognition accuracy of the majority classes of sample. From Fig. 5, it is apparent that the highest average classification accuracy for each model is achieved when the number of synthesized minority classes of samples is 4400. Therefore, in this experiment, we augment each minority classes of samples with 4400 samples using the GS-ACGAN. Notably, if the importance of faults is considered, different weights can be assigned to the accuracy of each class when exploring the optimal number of data augmentations. This method helps improve the recognition accuracy of specific targeted faults.

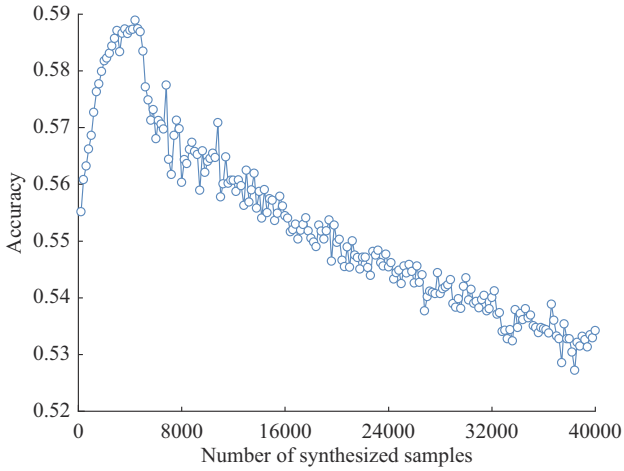


Fig. 5. Variation of SM fault diagnosis accuracy with different numbers of synthesized samples.

The effectiveness of the proposed data augmentation method is validated using the Stacking ensemble classifier developed in this study, as presented in Table VIII.

TABLE VIII
COMPARISON OF SM FAULT DIAGNOSIS ACCURACY WITH DIFFERENT DATA AUGMENTATION METHODS

| Category | Accuracy | | | |
|---|---|---|---|---|
| | GS-ACGAN | SMOTE-NC | ROS | ACGAN |
| 1 | 0.599 | 0.586 | 0.580 | 0.573 |
| 2 | 0.813 | 0.793 | 0.774 | 0.784 |
| 3 | 0.754 | 0.749 | 0.751 | 0.746 |
| 4 | 0.529 | 0.492 | 0.463 | 0.481 |
| 5 | 0.702 | 0.681 | 0.683 | 0.688 |
| 6 | 0.752 | 0.746 | 0.733 | 0.734 |
| 7 | 0.786 | 0.779 | 0.759 | 0.763 |
| 8 | 0.655 | 0.642 | 0.633 | 0.629 |
| 9 | 0.919 | 0.904 | 0.885 | 0.913 |
| 10 | 0.634 | 0.606 | 0.592 | 0.597 |
| 11 | 0.631 | 0.619 | 0.617 | 0.622 |
| 12 | 0.501 | 0.460 | 0.439 | 0.428 |

The comparison includes the methods applicable to both continuous and discrete variables, namely, the SMOTE-nominally continuous (SMOTE-NC) and ROS. In addition, we directly apply ACGAN to data augmentation for comparison and forcibly constrain the generated discrete features to integers. Among these methods, ROS is proven to be the least effective because of the repetitive sampling of minority classes of samples, leading to an increase in redundant information and a decrease in the generalization ability of the model. In contrast, the data augmentation method based on GS-

ACGAN improves the classification accuracy of the minority classes of samples by an average of 5.86%, outperforming the other two methods. Moreover, the recognition accuracy of the majority classes of samples is minimally affected, indicating that the method synthesizes fewer noisy samples. This indicates that GS-ACGAN, which incorporates GS, enables synthesized samples to align more closely with the actual sample distribution, resulting in more effective data augmentation.

### D. Extended Experiment

To verify the effect of the proposed external archive, we compare the search performance when introducing and not introducing an external archive in the iterative process of the lower-layer optimization, as shown in Fig. 6. The number of individuals in the WOA is set to be 30, and the number of iterations is 50. As the upper-layer optimization feeds different Stacking configurations, the lower-layer takes different decision variables for the optimization, resulting in slower optimization for the traditional scheme with random initialization. The external archive is endowed with a heuristic memory function that can provide near-optimal initial hyperparameters for the lower-layer heuristic optimization, effectively reducing the number of searches, which fully verifies the effectiveness of the proposed method.

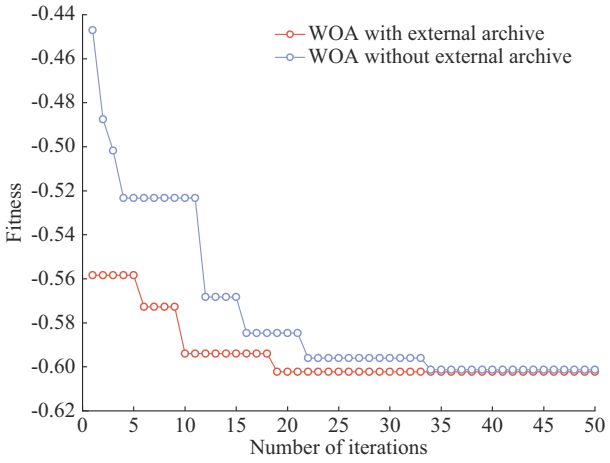


Fig. 6. Variation of fitness with different numbers of iterations.

## VII. Conclusion

In this study, we present an SM fault diagnosis method based on a two-layer Stacking ensemble optimization and data augmentation method. We develop feature engineering, data enhancement, and fault classification modules to investigate the characteristics of the SM data. The proposed method is evaluated using fault samples collected from Zhejiang Province. The results demonstrate the effectiveness of the proposed method. In the feature engineering module, the DESAE feature extraction method enhances the macro-F1-score by up to 1.72%. Through the GS-ACGAN, we explore the optimal number of synthesized samples, considering the importance of fault categories. This provides valuable insights for data augmentation in scenarios, where the importance of the fault category is considered. By employing the proposed method, we could adaptively select the optimal combination of base and meta-classifiers. This enables the combination method to improve the performance of the prediction model effectively. Furthermore, a comparison with other state-of-the-art classifiers such as SVM, RF, CNN, and LightGBM demonstrates the superiority of the proposed method in terms of accuracy, making it suitable for industrial applications.

This study provides a robust solution to the stable operation of power systems and offers insights for fault diagnosis studies in other device domains. For example, the proposed two-layer Stacking ensemble optimization and data augmentation can obtain an optimal heterogeneous model integration solution for any task. However, the lack of training data makes it difficult to verify more complex SMs and limits the effectiveness of fault identification. We plan to collect additional SM fault samples to validate and improve the proposed method further in the future.

## References

[1] Y. Wang, Q. Chen, T. Hong *et al.*, "Review of smart meter data analytics: applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125-3148, May 2019.

[2] Global and China Smart Meters Industry. (2023, May). Global and China smart meters industry report, 2022-2027. [Online]. Available: https://www.reportlinker.com/p06286403/?utm_source=GNW

[3] P. Tao, H. Shen, Y. Zhang *et al.*, "Status forecast and fault classification of smart meters using LightGBM algorithm improved by random forest," *Wireless Communications & Mobile Computing*, vol. 2022, p. 3846637, May 2022.

[4] L. Ge, T. Du, C. Li *et al.*, "Virtual collection for distributed photovoltaic data: challenges, methodologies, and applications," *Energies*, vol. 15, no. 23, p. 8783, Dec. 2022.

[5] P. Ray, S. S. Reddy, and T. Banerjee, "Various dimension reduction techniques for high dimensional data analysis: a review," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3473-3515, Jun. 2021.

[6] L. Ma, Y. Cheng, Y. Ding *et al.*, "Binomial adversarial representation learning for machinery fault feature extraction and diagnosis," *Applied Soft Computing*, vol. 131, p.109772, Dec. 2022.

[7] M. Cui, Y. Wang, X. Lin *et al.*, "Fault diagnosis of rolling bearings based on an improved stack autoencoder and support vector machine," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4927-4937, Feb. 2021.

[8] F. Jia, Y. Lei, L. Guo *et al.*, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619-628, Jan. 2018.

[9] K. Chen, J. Hu, and J. He, "A framework for automatically extracting overvoltage features based on sparse autoencoder," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 594-604, Mar. 2018.

[10] Z. Pu, C. Li, S. Zhang *et al.*, "Fault diagnosis for wind turbine gearboxes by using deep enhanced fusion network," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, p. 2501811, Sept. 2021.

[11] W. Li, Z. Shang, M. Gao *et al.*, "A novel deep autoencoder and hyperparametric adaptive learning for imbalance intelligent fault diagnosis of rotating machinery," *Engineering Applications of Artificial Intelligence*, vol. 102, p. 104279, Jun. 2021.

[12] W. W. Y. Ng, J. Hu, D. S. Yeung *et al.*, "Diversified sensitivity-based undersampling for imbalance classification problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2402-2412, Nov. 2015.

[13] L. Chen, S. Wan, and L. Dou, "Improving diagnostic performance of high-voltage circuit breakers on imbalanced data using an oversampling method," *IEEE Transactions on Power Delivery*, vol. 37, no. 4, pp. 2704-2716, Aug. 2022.

[14] L. Wang, M. Han, X. Li *et al.*, "Review of classification methods on unbalanced data sets," *IEEE Access*, vol. 9, pp. 64606-64628, Apr. 2021.

[15] F. Zhou, S. Yang, H. Fujita *et al.*, "Deep learning fault diagnosis method based on global optimization GAN for unbalanced data," *Knowledge-Based Systems*, vol. 187, p. 104837, Jan. 2020.

[16] W. Li, X. Zhong, H. Shao *et al.*, "Multi-mode data augmentation and fault diagnosis of rotating machinery using modified ACGAN designed with new framework," *Advanced Engineering Informatics*, vol.

52, p. 101552, Apr. 2022.
[17] Z. Meng, Q. Li, D. Sun *et al*., "An intelligent fault diagnosis method of small sample bearing based on improved auxiliary classification generative adversarial network," *IEEE Sensors Journal*, vol. 22, no. 20, pp. 19543-19555, Oct. 2022.
[18] X. Dong, Z. Yu, W. Cao *et al*., "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241-258, Apr. 2020.
[19] S. Tian, J. Li, J. Zhang *et al*., "STLRF-Stack: a fault prediction model for pure electric vehicles based on a high dimensional imbalanced dataset," *IET Intelligent Transport Systems*, vol. 17, no. 2, pp. 400-417, Feb. 2023.
[20] I. U. Khan, N. Javeid, C. J. Taylor *et al*., "A stacked machine and deep learning-based approach for analysing electricity theft in smart grids," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1633-1644, Mar. 2022.
[21] P. W. Khan and Y. C. Byun, "Multi-fault detection and classification of wind turbines using stacking classifier," *Sensors-Basel*, vol. 22, no. 18, p. 6955, Sept. 2022.
[22] P. Radhakrishnan, K. Ramaiyan, A. Vinayagam *et al*., "A stacking ensemble classification model for detection and classification of power quality disturbances in PV integrated power network," *Measurement*, vol. 175, p. 109025, Apr. 2021.
[23] Z. Sahri, R. Yusof, and J. Watada, "FINNIm: iterative imputation of missing values in dissolved gas analysis dataset," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2093-2102, Nov. 2014.
[24] X. Miao, Y. Gao, G. Chen *et al*., "Processing incomplete *k* nearest neighbor search," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1349-1363, Dec. 2016.
[25] T. Kuo and K. Wang, "A hybrid *k*-prototypes clustering approach with improved sine-cosine algorithm for mixed-data classification," *Computers & Industrial Engineering*, vol. 169, p.108164, Jul. 2022.
[26] S. Zhang, Z. Han, Y. Lai *et al*., "Stylistic scene enhancement GAN: mixed stylistic enhancement generation for 3D indoor scenes," *Visual Computer*, vol. 35, no. 6, pp. 1157-1169, Jun. 2019.
[27] Y. Xia, B. Gou, and Y. Xu, "A new ensemble-based classifier for IGBT open-circuit fault diagnosis in three-phase PWM converter," *Protection and Control of Modern Power Systems*, vol. 3, no. 1, p. 33, Nov. 2018.
[28] S. Talatahari and M. Azizi, "Chaos game optimization: a novel metaheuristic algorithm," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 917-1004, Feb. 2021.
[29] L. Ge, Y. Li, J. Yan *et al*., "Smart distribution network situation awareness for high-quality operation and maintenance: a brief review," *Energies*, vol. 15, no. 3, pp. 828, Feb. 2022.
[30] L. Ge, H. Liu, J. Yan *et al*., "A virtual data collection model of distributed PVs considering spatio-temporal coupling and affine optimization reference," *IEEE Transactions on Power Systems*, vol. 38, no. 4, pp. 3939-3951, Jul. 2023.
[31] H. Cai, Q. Chen, Z. Guan *et al*., "Day-ahead optimal charging/discharging scheduling for electric vehicles in microgrids," *Protection and Control of Modern Power Systems*, vol. 3, no. 1, p. 9, Apr. 2018.
[32] M. Amroune, T. Bouktir, and I. Musirin, "Power system voltage instability risk mitigation via emergency demand response-based whale optimization algorithm," *Protection and Control of Modern Power Systems*, vol. 4, no. 1, p. 25, Nov. 2019.
[33] L. Ge, Y. Li, J. Yan *et al*., "Multivariate two-stage adaptive-stacking prediction of regional integrated energy system," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 5, pp. 1462-1479, Sept. 2023.
[34] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, May 2016.
[35] M. J. Islam, X. Li, and Y. Mei, "A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO," *Applied Soft Computing*, vol. 59, pp. 182-196, Oct. 2017.

**Leijiao Ge** received the Ph.D. degree in electrical engineering from Tianjin University, Tianjin, China, in 2016. He is currently an Associate Professor in the School of Electrical and Information Engineering in Tianjin University. His main research interests include smart distribution network, cloud computing, and big data.

**Tianshuo Du** is a graduate student at Tianjin University, Tianjin, China. His research interests include new energy grid-connected optimization control technology.

**Zhengyang Xu** received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Information Engineering in Tianjin University, Tianjin, China. Currently, he is a Lecturer with the School of Electrical and Information Engineering, Tianjin University. His research interests include distribution system evaluation and planning.

**Luyang Hou** received the B.Eng. degree in mechanical design, manufacturing, and automation from Henan Polytechnic University, Jiaozuo, China, in 2013, the M.S. degree in micro-electromechanical engineering from Dalian University of Technology, Dalian, China, in 2016, and the Ph.D. degree in information systems engineering from Concordia University, Montreal, Canada, in 2020. He is currently an Associate Researcher at the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include political, technical, and economic planning, and operation for the smart green infrastructures.

**Jun Yan** received the B.Eng. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2011, and the M.S. and Ph.D. (hons.) degrees in electrical engineering from The University of Rhode Island, Rhode Island, USA, in 2013 and 2017, respectively. He is currently an Assistant Professor with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada. His research interests include computational intelligence and cyber-physical security in smart critical infrastructures.

**Yuanliang Li** received the B.E. degree in electrical engineering and automation from China University of Petroleum, Qingdao, China, in 2016, and the M.S. degree in electrical engineering from Tianjin University, Tianjin, China, in 2020. He is currently working at the Concordia School of Information Systems Engineering, Concordia University, Montreal, Canada. His research interests include situational awareness of smart distribution networks and big data.