

Research Article

Center Loss Guided Prototypical Networks for Unbalance Few-Shot Industrial Fault Diagnosis

Tong Yu ¹, Haobin Guo ², and Yiyi Zhu ³

¹College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350000, China

²School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou 510000, China

³School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310000, China

Correspondence should be addressed to Tong Yu; firstplace2021@163.com

Received 1 June 2022; Revised 12 July 2022; Accepted 17 September 2022; Published 9 October 2022

Academic Editor: Alessandro Bazzi

Copyright © 2022 Tong Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The success of deep learning is based on a large number of tagged data, which is challenging to satisfy on many occasions. Especially in industry fault diagnosis, considering the cost of data collection, the fault data are few and severely unbalanced. Therefore, it is not enough to support a reliable data-driven deep learning model. Few-shot learning effectively solves the few sample problems, but traditional methods pay little attention to the impact of unbalanced data. However, imbalanced data exists in large quantities. At the same time, unbalanced data often causes decision boundaries to be biased towards categories with larger sample sizes, resulting in lower accuracy. This study proposes a prototype network incorporating center loss for diagnosing industrial faults with few-shot samples. Based on the prototypical networks, by adding center loss at the loss level, the mapping points of the samples in the feature space play the role of intraclass contraction and interclass separation, thereby improving the classification effect. The experiment takes the TE process industrial data set as an example. Comparing various current few-shot learning methods reflects the superiority of the method proposed in the few-shot imbalanced scenario.

1. Introduction

Industrial Internet plays a significant role in society, and a variety of sensors are embedded in the actual industrial process to collect experimental data, and the data in this study are based on this. With the evolution of industrial intelligence skills and the emergence of Industry 4.0 [1, 2], the identification and diagnosis of industrial engineering faults are particularly critical in maintaining mechanical equipment safety and availability. Industrial failures will lead to a shorter life of components, mechanical damage, and even casualties. Meanwhile, the need for experts to diagnose through specific technical means often consumes a lot of human resources and property and has a certain chance of missing the golden opportunity to deal with the fault. Therefore, accurate prediction and diagnosis of various failures in real industrial scenarios are significant. The fault diagnosis in the real world has a problem, that is, the same fault varies significantly under different working conditions,

so it is often quite challenging to acquire a sufficient sample. For example, labeled data are expensive. This situation can arise for several following reasons: First of all, the industry system does not allow some failures that can bring huge losses to occur frequently. Second, most electromechanical failures occur as a slow process and follow a degradation path, making the system's failure degradation a lengthy process, which brings substantial time costs to collecting relevant data sets and is challenging to obtain. Third, the operating conditions of mechanical systems are very complex and constantly change from generation to generation depending on manufacturing requirements. Collecting and labeling sufficient training samples is impractical [2]. Therefore, using a data-driven fault inspection and analysis method based on few-shot learning is very relevant. In FSL methods, three categories can be distinguished as follows: incremental learning, metalearning, and metric learning.

Data augmentation [3, 4] is a method of synthesizing new sample data by mining existing data information. Data

augmentation methods can be classified into two levels, namely, data level and feature level. The data level takes the picture data as an example to achieve the purpose of synthesizing new data samples by simply rotating, flipping, clipping, and adding a little noise to the picture. However, this method cannot bring helpful information for the training of the model based on existing data. It may even lead to a decrease in inaccuracy due to the addition of noise. The feature level is to generate useful information to synthesize new data samples by fully mining the characteristics of existing data. Nowadays, the popular feature-level data augmentation methods include feature trajectory transfer (FFT) [5] and attribute-guided augmentation (AGA) [6]. Through the transfer of feature trajectories, FFT can obtain the feature trajectories by learning one and transferring the feature trajectories to other categories with fewer samples to enhance the feature-level data [5]. However, this method needs to have a fine-grained and continuous description, which is a prohibitive cost for data preparation. AGA trains an encoder-decoder network with the ability to synthesize another comprehensive feature and obtain its mapping relationship by using the sample input features, to synthesize the missing features with the help of the existing features of the sample, to realize data augmentation. However, this method needs to have side information [6]. In summary, the data augmentation method needs to fully mine the feature information provided by the existing data samples, which often requires side information. Therefore, mining side information becomes a difficult problem in data augmentation.

Metalearning [7] is seeking to directly optimize a fast-learning algorithm by using a dataset of tasks [8] as a new and efficient cross-task learning strategy. Metalearning has an essential role in few-shot learning. Snell et al. [9] proposed measure prototypical nets by solving the overfitting problem caused by less training data. The model-agnostic metalearning (MAML) proposed by Finn et al. [10] can be combined with any task based on gradient update to maximize the accuracy of the model under iteration by learning its initialization parameters. Mishra et al. [11] proposed a simple neural attentive metalearner (SNAIL), which applied the new combination of sequential convolution and causal attention mechanism to achieve a commendable prediction effect in new samples. However, metalearning has some limitations. The similarity between tasks should not be too high. Otherwise, it will degenerate into supervised learning and fail to achieve memorization.

Metric learning [12–14], the main genre of the minority learning field, categorizes query samples by learning the feature extraction agents on the underlying classes, extracts sample features from novice classes in the testing process, and gauges the separation or likeness between labeled supporting samples and unaffected query samples. The two most representative classes of metric learning are twin networks and prototype networks, respectively. Unlike standard classifiers, conjoined networks [15, 16] can perform the classification of samples from a new class without any retraining of the new category. Training is initially performed offline on many sample pairs which belong to the

same category or different categories. After the Siamese network is downloaded, the data to be categorized is matched against marked proxy examples of each category. It is referred to as prototypes in the remainder of this work. Incoming samples do not need to belong to the class seen during training. The category that wins is the one that corresponds to the maximum likeness between the sample of interest and the preserved prototype. The design of the prototypical network [17] presumes the presence of an embedded space, in which the sample projects of each class are clustered around a single prototype (or centroid). Classification is then performed by counting the separation from the prototype representation of each category in the embedding space. By doing this, the generic adaptation is to use one prototype, which represents the distribution of each class and matches the prototype of each class in the embedding space learned on the data from different domains. However, most existing probability learning methods [9, 18–21] focus on the relevance among the supporting and query samples and do not take sufficient advantage of the information in the foundation classes, resulting in sample imbalance that is not well addressed.

The above approaches can achieve promising results in solving the few-shot-learning problem but do not involve the class imbalance problem. The category imbalance problem is very widespread in the industry, and the data sets of different categories of industrial failures are often not uniformly distributed ideally. The data will show a “long-tail distribution” [21] when sorted by the frequency of different categories of data from high to low. It is supposed that the unbalanced samples are given to the model for training. In that case, the model will learn the prior information of the sample proportions in the training set to minimize the value of the loss function, resulting in the actual prediction will be concentrated on the primary class and the generalization ability of the second class will be poor, which will affect the robustness of the model learning.

For classical machine learning models [22–25], it can be divided into the sample level and the model algorithm level. At the sample level, it can be subdivided into under sampling [26] to decrease the number of majority classes, over-sampling [27] to increase the number of minority classes, and data augmentation methods. These aspects aim to equalize the gradient contribution of the sample to the model learning, eliminate the bias of the model towards different classes, and learn more essential features. At the model algorithm level, they can be subdivided into the use of classification models that are insensitive to imbalance, weighing penalties for small class misclassifications, and processing at the reconfigured classifier level. In this case, the learning algorithm is modified. For example, at the classifier level, the misclassification of examples from different classes is performed by introducing different weights [28] or by explicitly adjusting the prior class probabilities [29]. However, none of these methods is used based on a few-shot-learning framework. Since fault diagnosis is very critical in industry, industrial fault samples are often characterized by unbalanced categories, a tiny number of some samples, and small differences between similar categories.

At present, in industrial faults, the diagnosis of faults is very important. Deep learning has been widely used in the detection of industrial faults, such as adding schedulers to communication infrastructure to solve crashes [30]. At the same time, artificial intelligence models have also been applied to automated decision-making, such as explainable artificial intelligence (XAI) systems in the role of the health care field [31, 32], although these methods have good results for fault diagnosis, then for some faults, due to the relative difficulty, the fault diagnosis of small sample learning is also very important.

This study is motivated by the fact that on the one hand, the problem of insufficient sample data is fully considered and small sample learning is taken into account; on the other hand, the problem of sample imbalance is further considered and the prototype network ideas are combined to effectively address the decision bias caused by sample imbalance and make industrial fault classification more accurate.

Considering the problem of insufficient samples, we adopt the idea of the prototype. It is a metric-based approach to modeling the distance across samples. However, the loss function is used in the literature, which does not consider the scaling strategy for the distance and the class imbalance. In order to better measure the similarity between a query image and a sample image, this study combines discriminant loss function and prototypical network, which are suitable for intraclass compression and interclass separation to solve the problem of small sample imbalance. Specifically, this work designs a novel prototypical network for industrial fault diagnosis and tests it on a benchmark task with an industrial dataset. The findings are indicated that the current approach is superior to traditional approaches concerning FSL. The contribution points of this study are concluded as follows:

- (i) An improved prototypical network model is designed combined with center loss. By compacting the intraclass samples and separating the interclass samples, the proposed method can resolve the problem of class imbalance in few-shot learning.
- (ii) The presented method was applied to an industrial process in which several fault class imbalance cases were designed to validate the methodology. In contrast to several different learning methods, our method yields the most excellent results.

The remainder of the study is organized as follows: in the second subsection, we present the preparatory knowledge, in the third subsection, we describe in detail the experimental method, and in the next sections we present the experimental method, the results, and the summary analysis.

2. Preliminaries

In this study, we briefly introduce the concept of few-shot learning and the fundamental approach of the prototypical network.

2.1. Few-Shot Learning. Few-shot learning seeks to resolve machine learning tasks using a limited amount of data. FSL can split the data into three sections, namely, the training set,

the support set, and the query set. The training set is a category with a significant number of instances so that the model can learn a model that can extract features from that category. During the training phase, categories are randomly selected, with samples from each category (shared samples) being chosen as the supporting set from the training set. The remaining data from the categories samples are then selected as the query set for the model. When the support set includes categories and each category has a category, this is termed a c -way k -shot problem.

$$\begin{aligned} S &= \{(X_i, Y_i)\}_{i=1}^{c*k}, \\ Q &= \{(X'_i, Y'_i)\}_{i=1}^{k'.} \end{aligned} \quad (1)$$

2.2. The Prototypical Network. The prototypical network would be a kind of metric learning [9]. It learns a mapping that can extract pattern features from the training set to realize the mapping from input to embedded space. The metric chooses the space of one distance function $d: \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, +\infty)$, which computes the M -dimensional representation or prototype of each class through embedding functions with learning parameters. Typical distance functions are Euclidean distance and triangular chord distance, and in this study, the Euclidean distance is chosen. Moreover, each prototype is the average of the vector of embedding support points in its class. The prototypical representation of this class is obtained after averaging the samples. Finally, the same mapping operation is performed for the query set, and classification can be done by counting the separation from the prototype representing every class. The smaller the distance, the higher the probability that the sample belongs to the class, and the final classification result is the class with the highest probability. Then, a loss function is chosen to optimize the parameters in the mapping function. The specific formula is as follows:

$$p_\varphi(y = k|x) = \frac{\exp(-d(f_\varphi(x), c_k))}{\sum_{k'} \exp(-d(f_\varphi(x), c_{k'}))}. \quad (2)$$

In this formula, the input data are the sample attribute variable x and any class label y , and the output data are the probability that the sample belongs to this class. The formula represents the feature vector of the prototype representation of the corresponding category, represents the feature vector of the corresponding sample in the feature space, and is the mapping function of the attribute vector of the sample to the feature vector. It represents the Euclidean distance between the sample feature vector and the prototype of the category representing the feature vector. It can be seen from this formula that the closer the distance between the sample and the prototype representation of a certain category is, the greater the probability that the sample belongs to this category is.

3. Method

3.1. Necessity of the Research. In FSL, sample imbalance will have a great impact. The FSL method based on a prototypical network is taken as an example to explain in detail. Each

class has four data when there are five classes to solve a classification problem, which is called the 5-way 4-shot FSL task. The data belonging to the same class are grouped, and these supporting data are cast into the feature space using a featured network. Then, the prototypes c_1, c_2, c_3, c_4, c_5 are calculated as the average of the embedded supporting data for each class. The mean of the eigenvectors generated by the network c_1, c_2, c_3, c_4, c_5 is taken. Then, it takes the average of the centers of the five classes to get x . The fresh query picture is cast onto the characteristic or insertion space and compared with these prototypes using the Euclidean method distances to assign it to one of these classes. If x is closest to the class-1 prototype in balanced data classification, it belongs to class-1 (Figure 1).

However, due to the imbalance of data classification in industrial fault diagnosis, if the original 4 samples are reduced to 2 in the class-3 prototype, the range of judgment area will change. The x is close to the class-5 prototype during classification and may be misjudged as c_5 , resulting in classification errors. If we take the average method for the samples with few categories, classification errors are easy to occur, and the samples with few categories are not representative, resulting in poor robustness of the model. In order to maximize the value of each sample in the case of an imbalance few-shot sample, the improvement of FSL based on the prototypical network in this study is much essential.

3.2. Center Loss. The key to improving model classification accuracy is to reduce the minimization of space between classes and keep the space among classes, so a center loss is added in this study. Center loss requires similar features to be closer to their center points, thus directly constraining sample features, as shown in equation (2).

$$\mathcal{L}_c = \frac{1}{2} \sum_{i=1}^m \|x_i - p_\varphi(y = k|x)\|_2^2, \quad (3)$$

where x_i represents the feature extracted from the i sample, c_{y_i} represents the average feature of the i sample, and m stands for sample count.

The representation c_{y_i} and the gradient of \mathcal{L}_c are in formulas (3) and (4), where $\delta(x)$ is the function that represents when x is true, returning 1 and 0 otherwise. The "1" in the denominator is to prevent the exception of dividing by 0 because there are no j samples in the min-batch.

$$\begin{aligned} \frac{\partial \mathcal{L}_c}{\partial x_i} &= x_i - c_{y_i}, \\ \Delta c_j &= \frac{\sum_{i=1}^m \delta(y_i = j) \cdot (c_j - x_i)}{1 + \sum_{i=1}^m \delta(y_i = j)}, \end{aligned} \quad (4)$$

when we update the characteristic centroid c_{y_i} of class y_i , if the category y_i is not the same as the class corresponding to the feature center. That is, the feature of a specific class is only responsible for updating its corresponding class center c_{y_i} . Algorithm 1 shows the specific algorithm about the center loss.

3.3. Improving Prototypical Networks. Due to the instability of the prototypical representation of a few classes in the class imbalance problem, the classification region in the mapping space is chaotic. Therefore, the center loss can be added based on the SoftMax loss to shorten the distance within a category and increase the distance between categories so that the regional distribution of different classes becomes clear.

Given below is (4) for SoftMax loss plus center loss [33]:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_S + \lambda \mathcal{L}_c = - \sum_{i=1}^m \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_{y_i}^T x_i + b_{y_i}}} \\ &+ \frac{\lambda}{2} \sum_{i=1}^m \|x_i - p_\varphi(y = k|x)\|_2^2. \end{aligned} \quad (5)$$

In the improved prototype network proposed in this study, the main principle is reflected in the improvement of the loss function of the prototype network. In the loss function, it consists of SoftMax loss and center loss. In the SoftMax loss, $W_{y_i}^T x_i + b_{y_i}$ represents the function composed of the linear function and activation function of the determined parameters trained by the neural network. Its independent variable is the attribute vector of the sample, and its dependent variable is a scalar to measure the degree of membership of each category. It is normalized in SoftMax to make its value range [0, 1]. In center loss, x_i represents the feature vector corresponding to the attribute vector of each sample in the feature space, and $p_\varphi(y = k|x)$ represents the feature vector represented by the prototype of the corresponding category. λ measures the weight between SoftMax loss and center loss.

SoftMax activation function can get a number between 0 and 1, which is generally regarded as a probability belonging to this class. In SoftMax loss, the inverse of SoftMax is taken. If the label has been given, the greater the probability of belonging to this category obtained by SoftMax, the smaller the loss function value. Combined with the back propagation of neural network parameters correction, a classification effect is played. The Euclidean distance is used in the center loss function, which is obtained by summing the distances between all samples and the prototypes of the corresponding categories. If the distance sum is smaller, the loss function value is smaller. Combined with the correction of neural network parameters by back propagation, the intraclass distance is reduced and the interclass distance is expanded.

Training plots are shaped by forming a random choice of a category subsection in the training set, then selecting a subsection of instances in every class as a support set and the others as the query points. The prototypical network computes the M-dimensional expression $c_k \in \mathbb{R}^M$ or prototype for every class via an embedded function $f_\varphi: \mathbb{R}^D \rightarrow \mathbb{R}^M$ with respect to the learning-ready parameters φ . The cluster center of each class is as follows:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\varphi(x_i). \quad (6)$$

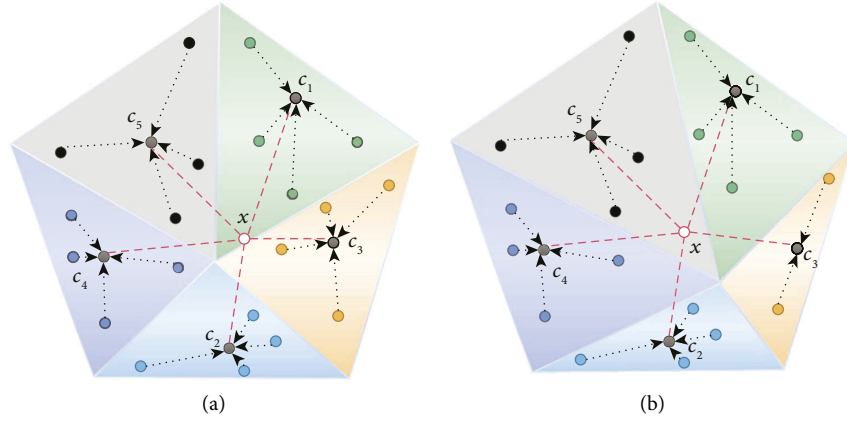


FIGURE 1: (a) Balanced data classification based on a prototypical network; (b) imbalanced data classification based on a prototypical network.

Step 1: Given a distance function $d: \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, +\infty)$, the prototypical network characterizes the sample x as belonging to a certain class based on SoftMax over the distance from the prototype in the embedding dimension, which can be shown in equation (2).

Step 2: After knowing the cluster center of each class of samples, we can characterize which class a sample x belongs to, represented by the distance and Softmax functions. At the same time, find the objective function J used by the parameter φ of the network $f_\varphi(x)$, as shown in equation (2).

Step 3: After the loss function J is obtained, the parameter φ of the embedding function is updated by stochastic gradient descent. The pseudocode to calculate the center loss is provided in Algorithm 1.

$$\begin{aligned}
 J(\varphi) &= \frac{1}{2} \sum_{i=1}^m \|x_i - p_\varphi(y = k|x)\|_2^2, \\
 &= d(f_\varphi(x), c_k) + \frac{1}{2} \sum_k \exp(-d(f_\varphi(x), c_k)).
 \end{aligned} \tag{7}$$

The training set damage calculations for prototypical networks. K is the count of classes in the training set, N is the count of examples in the trained set, $N_C \leq K$ is the count of classes per episode, N_S is the count of supported examples per class, N_Q is the count of query examples per class. $\text{RANDOMSAMPLE}(S, N)$ denotes the set of N elements selected evenly and stochastic without replacement from the set S . Algorithm 2 is about the algorithm of prototypical networks.

For the problem of small samples with imbalanced samples, this study combined the central loss function to improve the prototypical network, which realized the excellent characteristics of intraclass contraction and interclass separation, improved the distinguishing degree of feature categories, and effectively reduced the problems of classification deviation and misjudgment.

The diagram contains three modules, including the input layer, an operation layer, and output layer, and the working principle can be understood as follows:

- (1) The input layer represents the input of the training set and validation set data of different categories
- (2) The operation layer encodes the data, convolution, and other operations and processes the data at the loss level
- (3) The output layer performs a neural network on the data, extracts features, and performs predictions to obtain results

As illustrated in Figure 2, this figure contains three modules, including the embedded module, the induction module, and the output module. The embedded module represents the input of the different categories of data from the training and validation sets and projects the presented samples. The induction module performs operations such as coding convolution on the data and generates the category prototype G , while processing the missing level data. The output module performs neural network processing on the data, extracts the features, and performs predictive classification to obtain the final result. The final classification results are shown that the distance between blue and orange classes decreases, and the distance between classes increases, which makes the region division more obvious.

4. Experiment

4.1. Data Description. This experiment adopts the classic industrial data set of the Tennessee-Eastman (TE) process presented by Downs and Vogel [34]. The Tennessee-Eastman process simulated an actual chemical process for anomaly detection and process tuning. Moreover, the entire process comprises five operator units, namely, reactor, condenser, gas-liquid separator, circulation compressor, and product stripper. In the TE process, a single deterministic reservoir model fits several inputs and multiple export signals, mapping the signal space to the model space. TE

Input: Training data $\{x_i\}$. Initialization parameters φ_C in convolution layers. In loss layer the parameters W and $\{c_j | j = 1, 2, \dots, n\}$. Hyperparameter λ, α and learning rate μ^t . $T \leftarrow 0$.

Output: The parameters φ_C .

- (1) **while** not converge **do**
- (2) $t \leftarrow t + 1$.
- (3) Compute loss by \mathcal{L}_c .
- (4) Compute back spread fault $\partial \mathcal{L}_c / \partial x_i^t$ for each i .
- (5) Update W by $W^{t+1} = W^t - \mu^t \partial \mathcal{L}_c / \partial W^t$
- (6) Update c_j for every j by $c_j^{t+1} = c_j^t - \alpha \Delta c_j^t$.
- (7) Update φ_C by $\varphi_C^{t+1} = \varphi_C^t - \mu^t \sum_i^m (\partial \mathcal{L}_c / \partial x_i^t), (\partial x_i^t / \partial \theta_C^t)$
- (8) **end while**

ALGORITHM 1: The center loss algorithm.

Input: Training set $D = \{(x_1, y_1), \dots, (x_N, y_N)\}, y_i \in \{1, \dots, K\}$. D_k represents the subset of D containing all elements (x_i, y_i) .

Output: The loss J for a randomly generated the training set.

$V \leftarrow \text{RANDOMSAMPLE}(\{1, \dots, K\}, \text{NC})$.

for k in $\{1, \dots, \text{NC}\}$ **do**

$S_k \leftarrow \text{RANDOMSAMPLE}(D_{V_k}, N_S)$.

$\mathcal{L}_c \leftarrow (1/\text{NC}) \sum_{(x_i, y_i) \in S_k} f_\varphi(x_i)$.

end for

$J \leftarrow 0$.

$Q_k \leftarrow \text{RANDOMSAMPLE}(D_{V_k} \setminus S_k, N_Q)$

for k in $\{1, \dots, \text{NC}\}$ **do**

for (x, y) in Q_k **do**

$\mathcal{L} \rightarrow J + (1/N_C N_B) [d(f_\varphi(x), t, n\mathcal{L}_c) + (1/2) \sum_{k'} \exp(-d(f_\varphi(x), t, n\mathcal{L}_c))]$

end for

ALGORITHM 2: Prototypical networks [9].

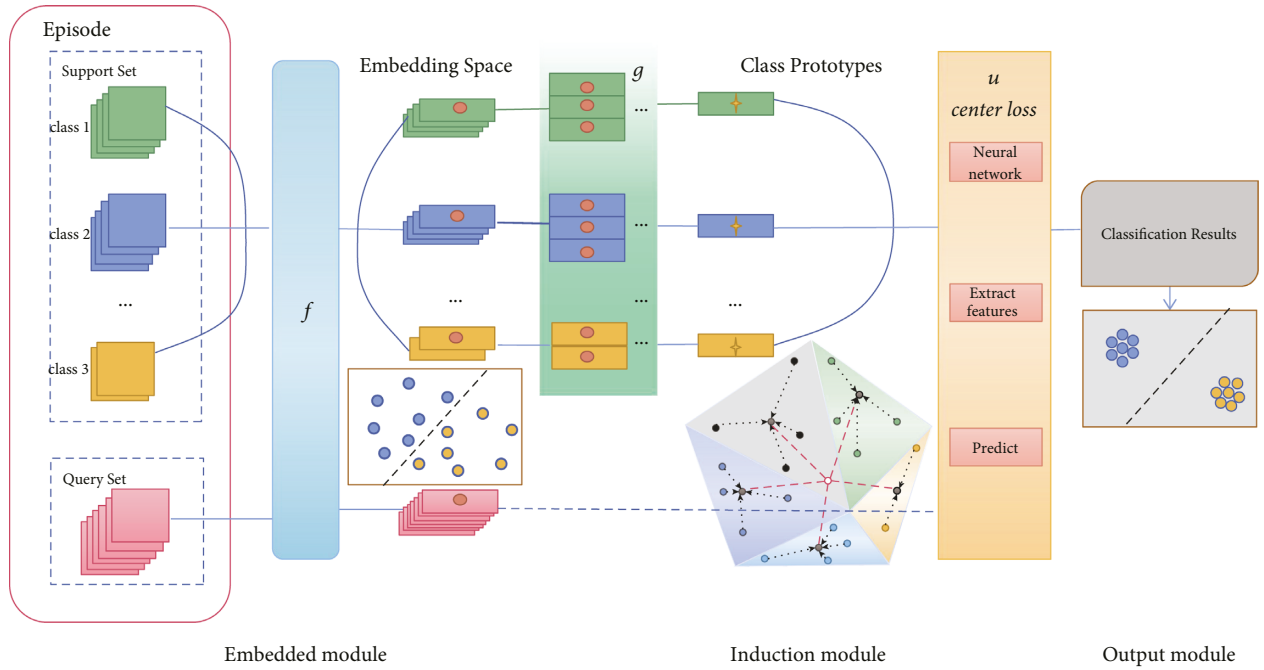


FIGURE 2: The entire infrastructure of the new module.

process data, as experimental data, has certain rigor and authority. TE process data has also been used for verification in previous literature [35–37].

TE process consists of 11 operational variables and 41 measurement variables. These 52 variables were used as input and analysis using a high-dimensional vector of fault data, with the specific meanings of the variables listed in Table 1 and Table 2. For the combination of few-shot learning and unbalanced data, the five faults shown in Table 3 were used in this experiment to demonstrate the superior capabilities of the model on the TE process dataset.

4.2. Experimental Details and Evaluation Indicators. A prototypical network combined with center loss was used for modeling the TE process data set. The batch size for the initialization parameters was 5, and the initial learning rate was set to 0.1.

In this experiment, accuracy rate and F1 were adopted in evaluation Indicators. Accuracy is one of the most common classification evaluation indexes to measure the classification accuracy of a classifier, which indicates the percentage of correctly classified samples with the overall sample. The accuracy rate is a good indicator of the model's ability to discriminate between negative samples. The more the accuracy rate, the better the model separates negative samples.

The recall is a good indicator of the model's capacity to discriminate between positive samples. The higher the recall rate is, the more capable the model is of distinguishing positive samples. In the reconciled average of accuracy and recall, the two are a pair of contradictory quantities. As one metric becomes better, it is often accompanied by another metric becoming worse. Therefore, to better evaluate the performance of the classifier, F1 and accuracy are used as evaluation criteria to measure the comprehensive performance of the classifier. Detailed information is shown in the following equation:

$$\begin{aligned} \text{Accuracy (ACC)} &= \frac{TP + TN}{TP + FP + TN + FN}, \\ \text{Recall}_i &= \frac{TP_i}{TP_i + FN_i}, \\ \text{Precision}_i &= \frac{TP_i}{TP_i + FP_i}, \\ \text{F1} &= \sum_{i=1}^C \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{(\text{Precision}_i + \text{Recall}_i) \cdot C}, \end{aligned} \quad (8)$$

where TP, FN, FP, and TN, respectively, express the number of successfully confirmed positive samples, positive samples wrongly perceived as negative, negative samples wrongly perceived as positive, and successfully confirmed negative samples.

4.3. Experimental Results. To prove the superiority of the new model algorithm in industrial fault detection under the condition of small samples and unbalanced samples, the

TABLE 1: TE procedure operation variables.

Variable no.	Description
1	D feed flow with stream
2	E feed flow with stream 3
3	A feed flow with stream 1
4	Total feed flow with stream 4
5	Compressor recycle valve
6	Purge valve with stream 9
7	Separator pot liquid flow with stream 10
8	Stripper liquid product flow
9	Stripper steam valve
10	Reactor cooling water flow
11	Condenser cooling water flow

TABLE 2: Measured (output) variables of the TE process.

Continuous variable number	Description
1	A feed with stream 1
2	D feed with stream 2
3	E feed with stream 3
4	The total feed with stream 4
5	The recycle flow with stream 8
6	The reactor feed rate with stream 6
7	The pressure in reactor
8	The level in reactor
9	The temp of separator
10	The purge rate with stream 9
11	The temp of separator
12	The pressure in separator
13	The pressure in separator
14	The separator underflow with stream 10
15	The level in stripper
16	The pressure in stripper
17	The stripper underflow with stream 11
18	Stripper temp
19	Stripper steam flow
20	The compressor work
21	Reactor cooling water export temp
22	Condenser cooling water export temp
23	A composition with stream 6
24	B composition with stream 6
25	C composition with stream 6
26	D composition with stream 6
27	E composition with stream 6
28	F composition with stream 6
29	A composition with stream 9
30	B composition with stream 9
31	C composition with stream 9
32	D composition with stream 9
33	E composition with stream 9
34	F composition with stream 9
35	G composition with stream 9
36	H composition with stream 9
37	D composition with stream 11
38	E composition with stream 11
39	F composition with stream 11
40	G composition with stream 11
41	H composition with stream 11

TABLE 3: Five fault types of the TE process.

Fault ID (IDV)	Disturbance
1	Stream 4: the A/C feed ratio, composition constant of B
2	Stream 4: the composition of B, the A/C ratio constant
3	Stream 2: feed temperature of D
4	The reactor cooling water intake temp
5	The condenser cooling water intake temp

experiment classifies the TE process data set and uses the appropriate training set and verification set (no overlap) to conduct comparative experiments on the model. Six hundred samples of each fault were selected for testing for the validation sets. This work chose metalearning and prototypical networks as the main contrast objects. Prototype-based on network coding is studied to extract the features of each sample; the sample taken on the sample average coding method, based on the classification results by the minimum distance, showed the use of Euclidean distance calculation to determine which classified query sample belongs to which category, obtaining the distance between the categories and using the Softmax converts from probability form.

The algorithm performance of each model is compared under different experimental conditions, namely, balance data sample (Plan 1) and unbalanced data sample (Plan 2, Plan 3, and Plan 4). The experiment set has different proportions for five fault types of the TE process. The number of training sets has been described in Table 4. Whether and what loss functions should be added to the model-agnostic metalearning (MAML [10]). Moreover, prototypical network models are quantitatively analyzed and calculated. Experimental results were further compared according to different groups, as shown in Table 5. For example, in the unbalanced case of Plan 4, the ACC of the MAML model increases from 38.9% to 49.2% after adding center loss, and F1 also increases from 41.4% to 56.3%. Compared with the simple application of the prototype network, the proposed new method of combining center loss with the prototype network improved ACC by 37.2% and F1 by 30.7%.

This work, Plan 1, is a balanced fault sample case. However, for Plan 2, the fault sample is unbalanced. Compared to other methods, IDV 4 and IDV 5 data plummeted from the original 10 to 2, leading to the unbalanced nature of the experimental samples as shown in Figure 3. The effect extracted by the prototypical network alone is more biased towards the three types of fault samples with a higher number in front, and less attention is paid to IDV 4 and IDV 5. The common prototypical network and similar methods do not consider the weight distribution of data ratios, which leads to the classification of faults belonging to IDV 4 and IDV 5 into IDV 1-IDV 3 with more fault samples, and the accuracy of fault classification results is reduced.

This study fully accounts for the few-shot imbalance problem by giving different attention to the different numbers of datasets to obtain weighted class prototypes.

Moreover, favorable classification results are obtained. Then, the results of different allocation ratios are compared with confusion matrix analysis, and it is obvious that the improved strategy has improved the classification performance. A confusion matrix compared the prototypical network with or without the addition of center loss. As shown in Table 6, for Plan 2, this experiment uses a total of 3000 test samples, 600 for each fault sample. For fault IDV1, our method improves the model accuracy by 24.2% compared to the simple prototype network, and for fault IDV5, it improves from the original 14.9% to 80.3%. In the left figure of the prototypical networks confusion matrix in Figure 3, it can be visualized that for IDV1, IDV2, and IDV3, the proportion of samples successfully classified was good, with accuracies reaching 0.65, 0.60, and 0.60, respectively.

However, for IDV4 and IDV5, the range of classification discriminations was relatively small due to their unevenly trained fault samples, and the prototypes were not very representative, resulting in a classification accuracy of only 0.16 for IDV4, which was more often discriminated as IDV5 during testing. The accuracy of misclassification as IDV5 reached 0.39. In IDV5, the accuracy of misclassification as IDV4 reached 0.38, while the accuracy of correctly classifying as IDV5 was only 0.15.

It is suggested that unbalanced samples have an impact on the accuracy of fault classification. With the inclusion of intraclass shrinkage and interclass separation center loss, the accuracy of successful classification for the fault classification samples IDV1, IDV2, and IDV3 exceeded 0.80 with 0.89, 0.85, and 0.93, respectively. For the unbalanced samples IDV4 and IDV5, the accuracy of successful classification reached 0.76 and 0.80. The accuracy of successful discrimination increased significantly, reflecting the model fault discrimination’s superior performance.

For Plan 3, the failure sample is the same is unbalanced. Compared with Plan 2, the two imbalance numbers become IDV 1 and IDV 2, and the data plummet from the original 10 to 2 and 1, respectively, with an extreme imbalance of the experimental samples. The effect of extraction with the prototypical network alone is more biased toward the three types of fault samples with a higher number of samples behind, and less attention is paid to IDV 1 and IDV 2; the fault classification originally belonging to IDV 1 and IDV 2 is incorrectly classified as IDV 3-IDV 5, which with a high number of fault samples causes the accuracy of the classification results decreases. While using prototypical networks + center loss, the accuracy is greatly improved with a successful classification accuracy of 89.8% for IDV 3, and the accuracy improved from 11.8% to 60.7% for IDV 2. For IDV 4, the accuracy also improved from 68.1% to 86.4%.

Also, for Plan 3, the experiment uses a total of 3000 test samples (Table 7). As can be visualized in the prototypical network confusion matrix in Figure 4(a), for IDV3, IDV4, and IDV5, correct classification accuracy reached 0.65, 0.68, and 0.70, respectively. However, for the unbalanced samples IDV1 and IDV2, the correct classification accuracy for IDV1 was only 0.22, and the accuracy of incorrect classification for IDV2 reached 0.51. For IDV2, the accuracy of correct classification was only 0.11, and the accuracy of incorrect

TABLE 4: Training set sample ratio setting.

Plan	The number of different fault samples				
	IDV1	IDV2	IDV3	IDV 4	IDV 5
1	10	10	10	10	10
2	10	10	10	2	2
3	2	1	10	10	10
4	2	3	1	10	10

TABLE 5: Quantitative comparison of models in different groups.

Methods	Plan 1		Plan 2		Plan 3		Plan 4	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
MAML [10]	33.3	46.2	43.4	41.6	44.6	50.3	38.9	41.4
MAML + center loss	53.9	66.6	56.3	61.7	52.9	68.2	49.2	56.3
Prototypical networks [38]	46.1	51.7	43.2	41.5	47.3	47.7	42.7	52.6
Prototypical networks + focal loss	80.3	81.6	71.5	70.2	70.2	71.6	81.7	82.8
Prototypical networks + center loss	91.7	97.2	84.6	83.9	78.1	78.1	79.9	83.3

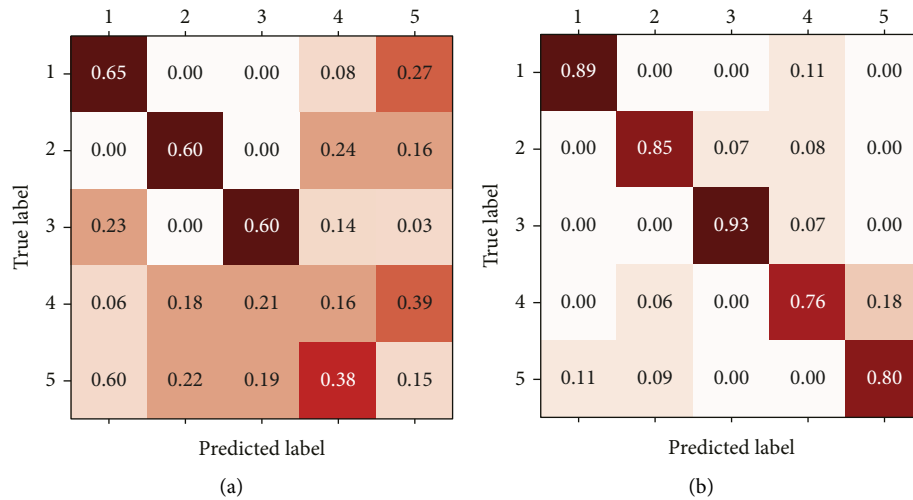


FIGURE 3: Results of the confusion matrices for prototypical networks (a) and prototypical networks + center loss (b) under unbalance sample (Plan 2).

TABLE 6: Comparison of model accuracy under the test set for Plan 2.

Methods	IDV 1 600	IDV 2 600	IDV 3 600	IDV 4 600	IDV 5 600
Prototypical networks (ACC %)	65.1	60.2	59.8	16.1	14.9
Prototypical networks + center loss (ACC %)	89.3	85.2	92.6	75.8	80.3

TABLE 7: Comparison of model accuracy under the test set for Plan 3.

Methods	IDV 1 600	IDV 2 600	IDV 3 600	IDV 4 600	IDV 5 600
Prototypical networks (ACC %)	21.7	11.8	64.9	68.1	70.2
Prototypical networks + center loss (ACC %)	69.4	60.7	89.8	86.4	84.3

classification for IDV1 was 0.49. It is indicated that few unbalanced shot samples have a considerable influence on fault classification accuracy. However, as shown on the

immediate right side of Figure 4, after the inclusion of intraclass shrinkage and loss of center of separation between classes. The accuracy of successful classification to IDV1

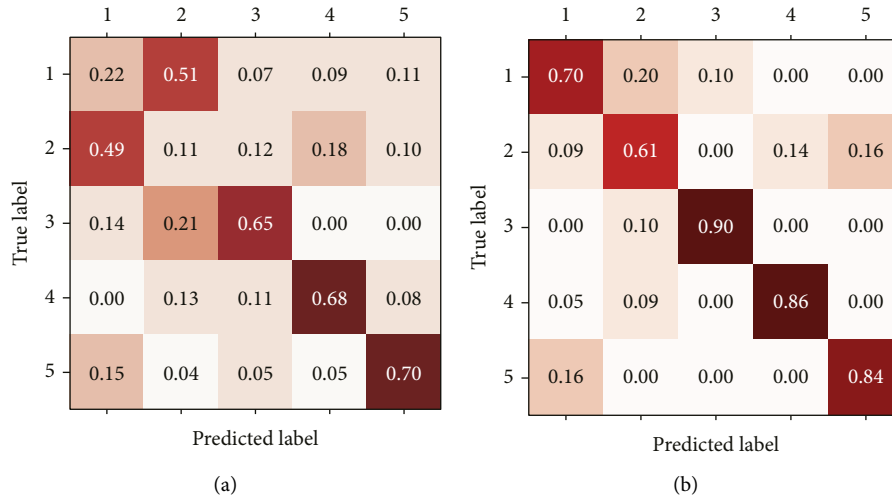


FIGURE 4: Results of the confusion matrices for prototypical networks (a) and prototypical networks + center loss (b) under unbalance sample (Plan 3).

TABLE 8: Comparison of model accuracy under the test set for Plan 4.

Methods	IDV 1 600	IDV 2 600	IDV 3 600	IDV 4 600	IDV 5 600
Prototypical networks (ACC %)	31.9	25.3	18.1	64.6	73.8
Prototypical networks + center loss (ACC %)	70.3	79.5	68.7	91.4	89.5

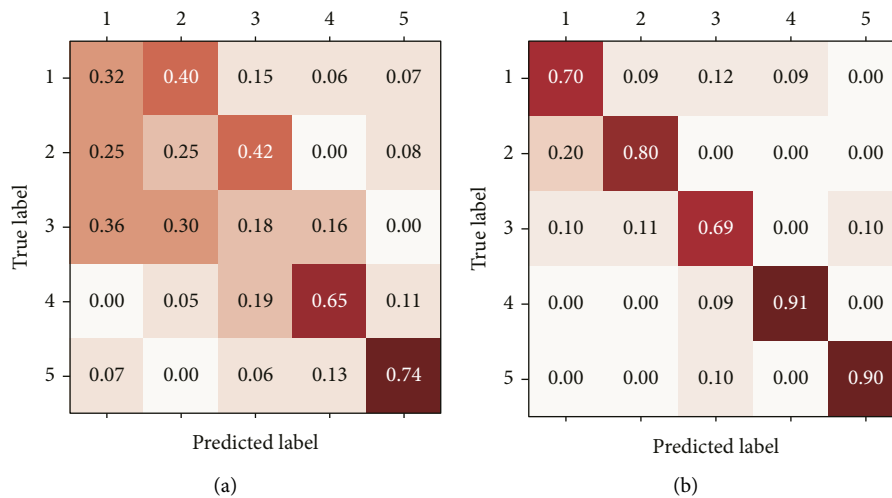


FIGURE 5: Results of the confusion matrices for Prototypical Networks (a) and prototypical networks + center loss (b) under unbalance sample (Plan 4).

reached 0.70, and the accuracy of successful classification to IDV2 reached 0.61. The accuracy of successful discrimination was significantly improved. The ACC for accurate classification of IDV3, IDV4, and IDV5 was also maintained at high levels of 0.90, 0.86, and 0.84, respectively.

The same is true for Plan 4, where the number of training samples for IDV1, IDV2, and IDV3 plummeted from 10 to 2, 3, and 1 (Table 8), respectively. Compared to the fault

classification using the prototype network alone, the accuracy of our method improved from 31.9% to 70.3% for IDV1, from 25.3% to 79.5% for IDV2, and 50.6%, 26.8%, and 15.7% for IDV3, IDV4, and IDV5, respectively. For IDV4 and IDV5, the results were only 0.65 and 0.74 for the prototypical network alone, but with the addition of central loss, the accuracy was increased to 0.91 and 0.90, a major breakthrough. For IDV3 with only 1 training sample, the accuracy

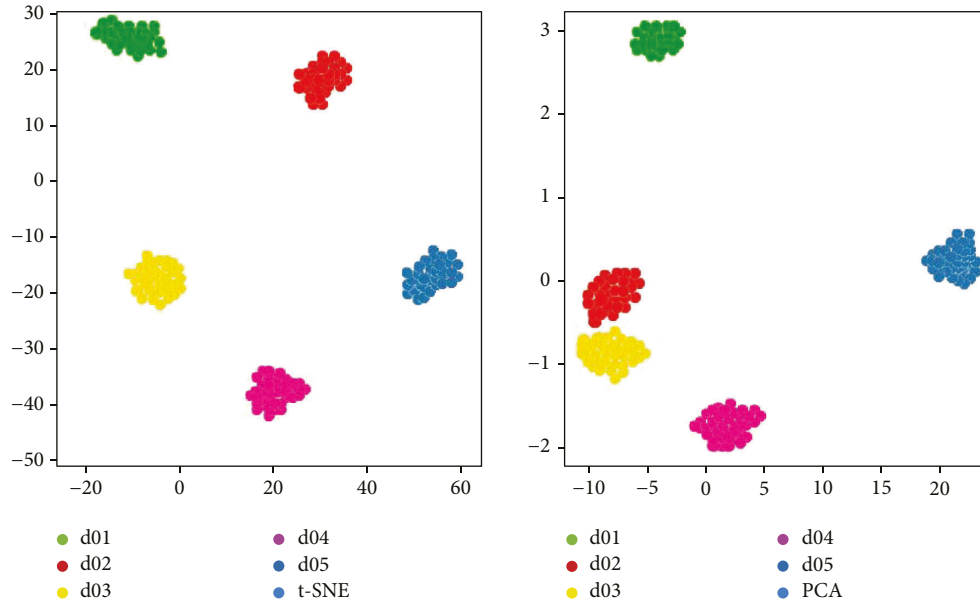


FIGURE 6: Data visualization for Plan 1 (sample balance).

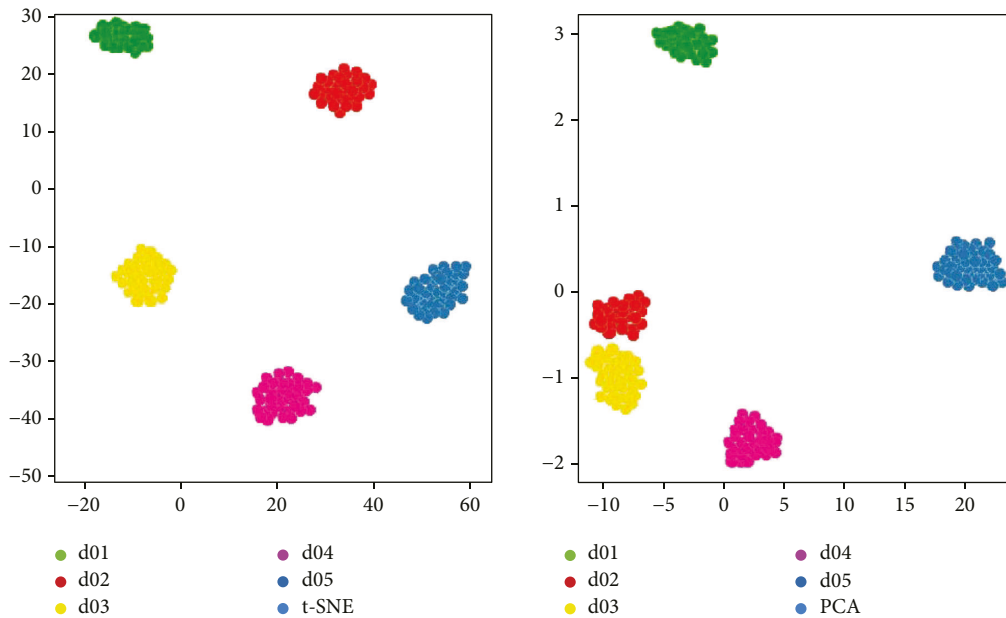


FIGURE 7: Data visualization for Plan 2 (sample unbalance).

increased from 0.18 to 0.69. The four unbalanced confusion matrices results show that our method achieves optimal classification results (Figure 5).

4.4. Analysis. The experimental outcomes show that the model presented in this study demonstrates superior performance under different equilibrium states. The addition of the loss function also brings about a significant improvement in model performance. There is a particular gap

between metalearning and the classification effect of prototypical networks. Compared with adding focal loss and center loss to prototypical networks, it is evident that the addition of central function is conducive to the contraction within classes and separation between classes, which further improves the accuracy of our classification. This study proposed a new model that performs well in the case of sample equilibrium (Plan 1), significantly outperforming other methods. In the case of unbalanced samples (Plan 2, Plan 3, and Plan 4), the effect is also better than the general

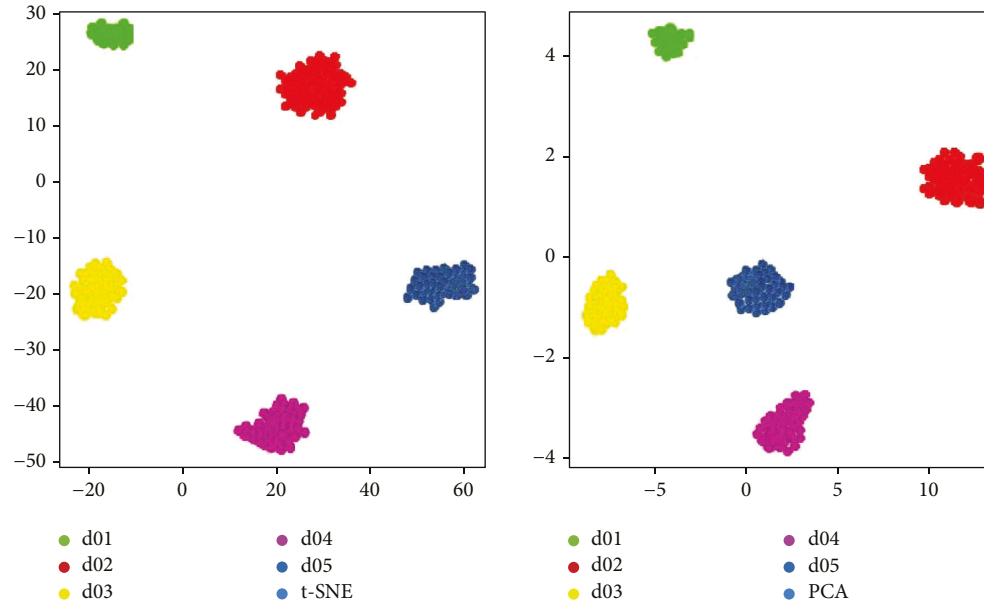


FIGURE 8: Data visualization for Plan 3 (sample unbalance).

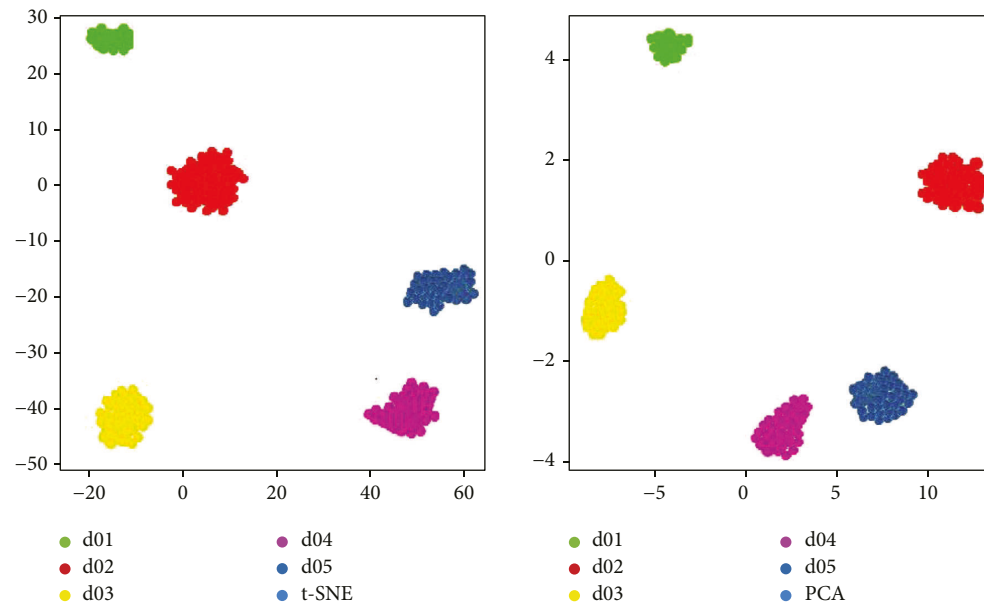


FIGURE 9: Data visualization for Plan 4 (sample unbalance).

results. The novel model algorithm introduced in this study has great potential for application in the industry where samples are difficult to obtain and where samples are unbalanced.

The experiment performed t-distributed stochastic neighbor embedding [39] (t-SNE) and principal component analysis [40] (PCA) dimensionality reduction operations on the results. Here, t-SNE transforms the distances between data points at high latitudes into Gaussian distribution probabilities, which is a nonlinear dimensionality reduction approach. PCA reduces the

number of feature dimensions used to train the model by constructing the so-called principal components from multiple features. The proposed model algorithm increases the precision of the fault categories and enables intraclass shrinkage and interclass classification, and the visualized results are shown in Figures 6–9. It can be seen that whether the samples are balanced or unbalanced, or even with very few training samples, our method can achieve the performance of intraclass tightening and interclass separation with clear classification and less confusion misclassification.

5. Conclusion

In this study, industrial failure diagnostics under complex operating situations with restricted data is considered an unbalanced classification problem for few-shot learning, and a prototypical network improvement model incorporating central loss is proposed. The characteristics of these training models are investigated using a learning method in model space to achieve intra-class contraction and intra-class separation, which can effectively recognize and segregate faults. This study investigates the TE process based on odds learning and imbalanced data problems for the first time. Through numerous experiments and simulations with other methods, the results show that the model obtains the best performance under different c -way k -shots. We will continue to investigate intelligent fault diagnosis based on prototype networks in the future. One aspect is to investigate the optimization of the hyperparameters of the prototype network, such as selecting the learning rate and meta-batch size in a learnable manner. The other area is the extension of supervised learning to semisupervised learning. [41, 42].

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interests.

References

- [1] M. S. Reis and G. Gins, "Industrial process monitoring in the big data/industry 4.0 era: from detection, to diagnosis, to prognosis," *Processes*, vol. 5, no. 4, p. 35, 2017.
- [2] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.
- [3] X. Li, W. Zhang, Q. Ding, and J. Q. Sun, "Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation," *Journal of Intelligent Manufacturing*, vol. 31, no. 2, pp. 433–452, 2020.
- [4] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol. 8, no. 1, pp. 101–134, 2021.
- [5] R. Kwitt, S. Hegenbart, and M. Niethammer, "One-shot learning of scene locations via feature trajectory transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 78–86, Vegas, NV, USA, December 2016.
- [6] M. Dixit, R. Kwitt, M. Niethammer, and N. Vasconcelos, "Aga: attribute-guided augmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7455–7463, Honolulu, HI, USA, July 2017.
- [7] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: a survey," 2020, <https://arxiv.org/abs/2004.05439>.
- [8] J. Lu, P. Gong, J. Ye, and C. Zhang, "Learning from very few samples: a survey," 2020, <https://arxiv.org/abs/2009.02653>.
- [9] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [10] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1126–1135, PMLR, New York City, NY, USA, 2017, July.
- [11] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," 2017, <https://arxiv.org/abs/1707.03141>.
- [12] C. G. Atkeson, A. W. Moore, and S. Schaal, *Locally Weighted Learning*, pp. 11–73, Lazy learning, 1997.
- [13] J. Goldberger, G. E. Hinton, S. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," *Advances in Neural Information Processing Systems*, vol. 17, 2004.
- [14] S. H. Chopra, R. R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 539–546, Diego, CA, USA, June 2005.
- [15] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *Advances in Neural Information Processing Systems*, vol. 6, 1993.
- [16] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition ICML deep learning workshop," vol. 2, no. 0, 2015, July.
- [17] Z. Zhan, J. Zhou, and B. Xu, "Fabric defect classification using prototypical network of few-shot learning algorithm," *Computers in Industry*, vol. 138, Article ID 103628, 2022.
- [18] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [19] B. Oreshkin, L. Iguéz, Opez, and Alexandre. Lacoste, *Tadam: Task Dependent Adaptive Metric for, Improved Few-Shot Learning*NIPS, Kolkata, 2018.
- [20] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," pp. 7260–7268, 2019, <https://arxiv.org/abs/1903.12290>.
- [21] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in *Proceedings of the European Conference on Computer Vision, Springer, Cham, Germany*, pp. 167–178, 2020.
- [22] N. V. Chawla, *Data Mining for Imbalanced Datasets: An Overview. Data Mining and Knowledge Discovery Handbook*, pp. 875–886, 2009.
- [23] N. Japkowicz and S. Stephen, "The class imbalance problem: a systematic study1," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.
- [24] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," *ICML-2003 workshop on learning from imbalanced data sets II*, vol. 2, pp. 2–1, 2003, August.
- [25] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, "Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance," *Neural Networks*, vol. 21, no. 2-3, pp. 427–436, 2008.

- [26] I. Cordon, S. Garcia, A. Fernandez, and F. Herrera, "Imbalance: oversampling algorithms for imbalanced classification in R," *Knowledge-Based Systems*, vol. 161, pp. 329–341, 2018.
- [27] S. Pouyanfar, Y. Tao, A. Mohan et al., "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 112–117, IEEE, Manhattan, NY, USA, April 2018.
- [28] Z. H. Zhou and X. Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.
- [29] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, and C. L. Giles, "Neural network classification and prior class probabilities," in *Neural Networks: Tricks of the Trade*, pp. 299–313, Springer, Berlin, Heidelberg, 1998.
- [30] M. Jemmali, M. Denden, W. Boulila, G. Srivastava, R. H. Jhaveri, and T. R. Gadekallu, "A novel model based on window-pass preferences for data-emergency-aware scheduling in computer networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7880–7888, 2022.
- [31] P. N. Srinivasu, A. K. Bhoi, R. H. Jhaveri, G. T. Reddy, and M. Bilal, "Probabilistic deep Q network for real-time path planning in censorious robotic procedures using force sensors," *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1773–1785, 2021.
- [32] P. N. Srinivasu, N. Sandhya, R. H. Jhaveri, and R. Raut, "From blackbox to explainable ai in healthcare: existing tools and case studies," *Mobile Information Systems*, vol. 2022, 2022.
- [33] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the European Conference on Computer Vision*, pp. 499–515, Springer, Cham, 2016, October.
- [34] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.
- [35] A. Bathelt, N. L. Ricker, and M. Jelali, "Revision of the Tennessee Eastman process model," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 309–314, 2015.
- [36] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.
- [37] X. Gao and J. Hou, "An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process," *Neurocomputing*, vol. 174, pp. 906–911, 2016.
- [38] S. Laenen and L. Bertinetto, "On episodes, prototypical networks, and few-shot learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [39] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.
- [40] J. Shlens, "A tutorial on principal component analysis," 2014, <https://arxiv.org/abs/1404.1100>.
- [41] F. Sung, Y. Yongxin, Z. Li, X. Tao, H. S. T. Philip, and M. H. Timothy, "Learning to compare: relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [42] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, "Research progress on few-shot learning for remote sensing image interpretation," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2387–2402, 2021.