Neurocomputing 407 (2020) 232–245

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Cost sensitive active learning using bidirectional gated recurrent neural networks for imbalanced fault diagnosis

Peng Peng[a], Wenjia Zhang[a], Yi Zhang[a], Yanyan Xu[a], Hongwei Wang[b,*], Heming Zhang[a,*]

[a] Department of Automation, Tsinghua University, Beijing, 100084, China
[b] ZJU-UIUC Institute, Zhejiang University, Haining, 314400, China

ARTICLE INFO

Article history:
Received 3 December 2019
Revised 14 March 2020
Accepted 12 April 2020
Available online 18 May 2020

Communicated by Hongli Dong

Keywords:
Fault diagnosis
Deep learning
Bidirectional GRU
Active learning
Class imbalance
Cost sensitive learning

ABSTRACT

Most existing fault diagnosis methods may fail in the following three scenarios: (1) serial correlations exist in the process data; (2) fault data are much less than normal data; and (3) it is impractical to obtain enough labeled data. In this paper, a novel form of the bidirectional gated recurrent unit (BGRU) is developed to underpin effective and efficient fault diagnosis using cost sensitive active learning. Specifically, BGRU is devised to consider the dynamic behavior of a complex process. In the training phase of BGRU, the idea of weighting each training example is proposed to reduce the effect of class imbalance. Besides, in order to explore the unlabeled data, cost sensitive active learning is utilized to select the candidate instances. The effectiveness of the proposed method is evaluated on the Tennessee Eastman (TE) dataset and a real plasma etching process dataset. The experiment results show that the proposed cost sensitive active learning bidirectional gated recurrent unit (CSALBGRU) method achieves better performance in both binary fault diagnosis and multi-class fault diagnosis.

1. Introduction

It is crucial to ensure the reliability and stability of industrial processes as a slight mistake could precipitate a disaster. This has resulted in the development of a wide range of fault diagnosis methods over the past few decades, e.g. the model-based approaches and the data-driven techniques. Specifically, the former has the challenge of obtaining accurate mathematical models for complex processes while the latter has become increasingly popular due to the advancement of new technologies such as 5G communication, Internet of things (IoT) and AI.

In the past decades, a number of data-driven fault diagnosis methods have been proposed, such as principal component analysis (PCA) [1], support vector machine(SVM) [2] and deep convolutional neural network (DCNN) [3], etc. Despite the effort to make the methods mentioned above more robust and reliable, they still may fail under three main circumstances. First, the process variables present auto-correlated and non-stationary behaviors. Second, faults tend to occur much less frequently so the normal classes turn to have more instances than the fault classes. Third, the data generated are very large nowadays with the rapid application of IT in industry, and as such it is often extremely hard to obtain enough labeled data.

For the first circumstance, recurrent neural network(RNN) has shown good performance in a wide range of time-series problems, in addition to the employment of deep learning techniques for learning the dynamic information in an end-to-end manner. Gated recurrent unit is a variant of recurrent units, which proves to be much simpler to compute and have better generalization performance compared with the popular long short-term memory (LSTM) [4,5] unit.

For the second circumstance, the standard supervised machine learning methods are challenged as these models pay much attention to the majority examples and distort the minority examples. Researchers from the machine learning community have developed some techniques to tackle the class imbalance problem including sampling methods, cost sensitive methods and kernel-based methods. Unfortunately, few of these methods take the class imbalance issue into consideration when they are applied to monitoring industrial processes.

For the third circumstance, if an appropriate rule is adopted to select the unlabeled instances for annotating, only a very small number of samples need to be annotated and the performance of the classification model may be improved to a large extent. In the machine learning community, this is called active learning. Common strategies used in active learning are uncertainty sampling, margin sampling and entropy sampling. As to the more efficient use of unlabeled samples, some semi-supervised approaches and active learning methods are introduced to monitor complex industrial processes. However, the class imbalance issue cannot be

* Corresponding authors.
E-mail addresses: hongweiwang@intl.zju.edu.cn (H. Wang), hmz@mail.tsinghua.edu.cn (H. Zhang).

https://doi.org/10.1016/j.neucom.2020.04.075
0925-2312/

well addressed when exploring the unlabeled data. To better take class imbalance into account when selecting the instances, cost sensitive active learning is developed by researchers from the machine learning community, e.g. the maximum expected cost and cost-weighted minimum margin strategies [6]. To address the challenges mentioned above, a cost sensitive active learning framework using the bidirectional gated neural recurrent unit is proposed for fault diagnosis of complex engineering processes. Specifically, the bidirectional gated recurrent unit (BGRU) [7] is proposed to capture dynamic information of a complex process; varied weights are assigned to samples to cope with imbalanced data classification; and cost sensitive active learning is proposed to explore the unlabeled data. The main contributions of this paper are summarized as follows:

- Bidirectional gated recurrent unit (BGRU) is proposed to learn the dynamic information from process data. Compared to the conventional fault diagnosis methods, BGRU can learn a fault diagnosis model in an end-to-end manner with the consideration of time series.
- The class imbalance issue in fault diagnosis is addressed by incorporating the sample sensitive learning strategy for imbalanced fault diagnosis. Varied weights are assigned to different classes in cost sensitive learning. For timing processes, the weights of samples at different times are naturally different while the samples that are taken shortly after a fault occurs should be given bigger weights. A detailed discussion is available in Section 3. Hence the weight of each sample is assigned in the training phase of BGRU by considering the class imbalance issue and the inherent structure of the process concerned.
- Cost sensitive active learning is adopted to explore unlabeled data. Due to class imbalance, the cost sensitive quality is also considered in active learning. Experiment results show that cost sensitive active learning can greatly improve the diagnosis performance with a small amount of sample annotation.
- Both binary fault classification and multi-class fault classification are considered. Most of the existing process monitoring methods only consider the binary fault classification. Despite being more challenging to address, multi-class fault classification is a more general form suitable for more applications. It is promising that the proposed CSALBGRU fault diagnosis framework even achieves good performance in multi-class fault classification.

The rest of this paper is organized as follows. In Section 2, the related work is discussed. In Section 3, the bidirectional gated recurrent unit and cost sensitive active learning methods are introduced, and on this basis, the evaluation index is introduced. In Section 4, the proposed cost sensitive active learning using a bidirectional gated recurrent unit (CSALBGRU) fault diagnosis framework is described in detail. In Section 5, the effectiveness of the proposed CSALBGRU framework is evaluated by its application to both the Tennessee Eastman process (TE) dataset and a real plasma etching process dataset. The main conclusions of this work are summarized in Section 5.

2. Related work

The data-driven methods can be divided into two main classes. The first major class of data-driven methods involves the multivariate statistical methods, such as principal component analysis (PCA), independent principal analysis (ICA), partial least square(PLS) and linear discriminant analysis (LDA) [1,8–10]. These methods emphasize the projection of raw data onto a lower-dimensional feature space where fault diagnosis can be performed. To address the dynamic properties of raw process data, some dynamic methods have also been proposed to use extended vectors that concatenate the current data with a certain number of past process data, e.g. the dynamic PCA(DPCA) [11], the dynamic ICA(DICA) [12], the dynamic PLS(DPLS) [13], and the dynamic LDA(DLDA) [14]. The main disadvantage of these methods is that they may aggravate the “curse of dimensionality” problem if the number of process variables is considerably large. Another major class of data-driven methods involves the use of conventional machine learning techniques, such as support vector machine(SVM) and Fisher discriminant analysis (FDA) [2,15]. The major challenge for these methods is that they may fail when dealing with a relatively large amount of data. More recent development of the field has also seen the applications of new machine learning algorithms such as deep learning, sparse auto-encoder (SAE) [16], deep convolutional neural network (DCNN) [3] and variational auto-encoder(VAE) [17,18].

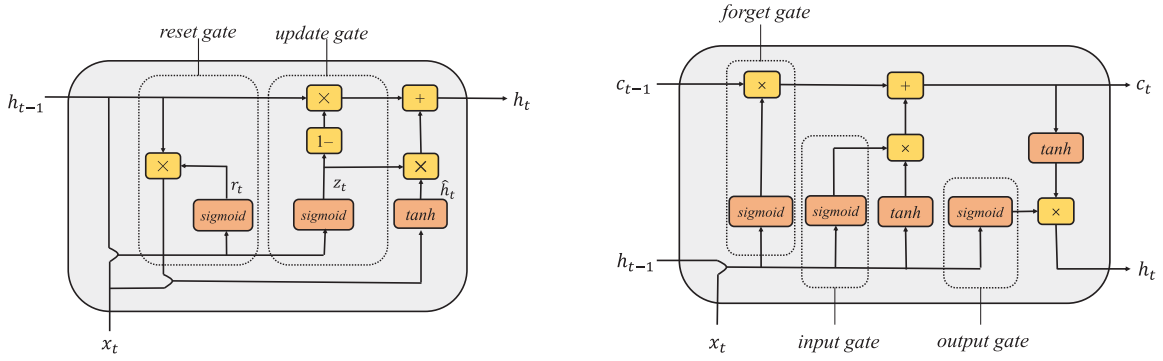
Most of the current data-driven fault diagnosis methods assume that the number of normal samples and fault sample is about the same. However, the number of fault data is generally much smaller than the number of normal data in real industrial scenarios. This means that the performance of these traditional methods tends to be very poor under the imbalanced distribution - the diagnosis result of the normal condition may be accurate while the diagnosis result of the fault condition may be inaccurate. Hence, more and more researchers start paying attention to the class imbalance problems in fault diagnosis recently. For example, Li et al. propose to use the deep transfer learning to develop an effective diagnostic method with insufficient training data as the knowledge transferred from the sufficient additional datasets can help address the issue of insufficient training data [19]. Lin et al. add the Laplacian regularization term into the original objective function of Deep Auto-encoder (DAE) to formulate the deep Laplacian Auto-encoder(DlapAE) and investigate its application in imbalanced fault diagnosis. The experiment shows that DlapAE has a better generalization performance and is more suitable for feature learning of imbalanced data [20]. Li et al. propose to use the generative neural networks(GANs) to generate the fault samples to assist model training [21,22]. However, most imbalanced fault diagnosis methods focus on using data generation techniques, while little work has been done to take advantage of cost-sensitive learning algorithms and apply these algorithms in this particular field.

Besides, the unlabeled process data can be easily collected while the acquisition of labeled data is very expensive in the industry from a practical point of view. How to select the informative samples and label them in the case of imbalanced data becomes an important problem. Most of the current literature is focused on the discussion of using active learning to fine tune the diagnosis model. For example, Jiang el at. use active learning to identify the most informative sensor data which can then be labeled for updating the learned parameters of deep neural network(DNN) [23]. Yin et al. incorporate active learning to fisher discriminant analysis (FDA) for industrial fault classification [24]. It is noteworthy that how to develop active learning under an imbalanced fault diagnosis situation is still an open problem and few research has tried to discuss this to the best of the authors' knowledge.

3. Backgroud theory

3.1. From recurrent neural network to bi-directional gated recurrent neural network

Recurrent neural networks (RNNs) [5] have been successful applied to handling sequential data in various fields. As most of the industrial processes are inherently dynamic, it is natural to consider RNN as an alternative model. Given an input sequence $x = (x_1, \ldots, x_T)$, hidden vector sequence $h = (h_1, \ldots, h_T)$ and



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Fig. 1. Illustration of (a)LSTM and (b) Gated recurrent units.

output vector sequence $y = (y_1, \ldots, y_T)$ can be derived using the following equations:

$$h_t = \Phi(Ux_t + Wh_{t-1} + b) \tag{1}$$

$$y_t = Vh_t + c \tag{2}$$

where Φ denotes the activation function, and the most popular activation function is usually an elementwise application of the sigmoid function. U denotes the input-hidden weight matrix, W denotes the hidden-hidden weight matrix, and b is the hidden bias vector. In Equation (2), V denotes the hidden-output weight matrix, and c is the output bias vector.

It is hardly possible to capture the long-term dependencies of RNNs as the gradients tend to either vanish or explode. Hence some researchers have made efforts to design a more sophisticated activation function to solve these problems. For example, the long short-term memory (LSTM) [4,5] unit is first devised to capture the long-term dependencies. And more recently, another variant of recurrent unit, gated recurrent unit (GRU) [25], is also proposed, which is much simpler to compute and has better generalization performance compared to the LSTM unit. The frameworks of the LSTM unit and the GRU unit are shown in Fig. 1. For LSTM, it uses an output gate to control the amount of memory content exposure.

$$h_t = o_t tanh(c_t) \tag{3}$$

where o_t is the output gate computed by:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t, c_t] + b_o) \tag{4}$$

where σ is the logistic function. The memory cell c_t is maintained by removing (forgetting) some existing memories and adding some new memories:

$$c_t = f_t c_{t-1} + i_t \widetilde{c}_t + b_c \tag{5}$$

the new memories $\widetilde{c}_t$ is :

$$\widetilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t]) \tag{6}$$

The extent to remove and add memories is controlled by the forget gate f_t and the input gate i_t. f_t is computed by:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t, c_{t-1}] + b_f\right) \tag{7}$$

and i_t is computed by:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t, c_{t-1}] + b_i) \tag{8}$$

where b denotes the corresponding bias vectors.

Similar to the LSTM unit, gated recurrent unit (GRU) use gates to control the flow of information inside a unit, but it has no memory cells. The hidden states h_t is a linear combination of previous hidden states h_{t-1} and new hidden states $\widetilde{h}_t$:

$$h_t = (1 - z_t)h_{t-1} + z_t \widetilde{h}_t \tag{9}$$

where z_t is the update gate which controls how much its new activation is updated. z_t is computed by:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{10}$$

The new activation $\widetilde{h}_t$ is computed by:

$$\widetilde{h}_t = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \tag{11}$$

where r_t is the forget gate, similar to the update unit in LSTM:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{12}$$

While conventional RNNs only exploit the previous information, the bidirectional RNNs (BRNNs) [26] can process data in both directions, as shown in Fig. 2. The output y of BRNN can be obtained by computing the forward hidden sequence $\overrightarrow{h}_t$ and backward sequence $\overleftarrow{h}_t$ in an iterative manner, using the following equations:

$$\overrightarrow{h}_t = \Phi\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \tag{13}$$

$$\overleftarrow{h}_t = \Phi\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \tag{14}$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \tag{15}$$

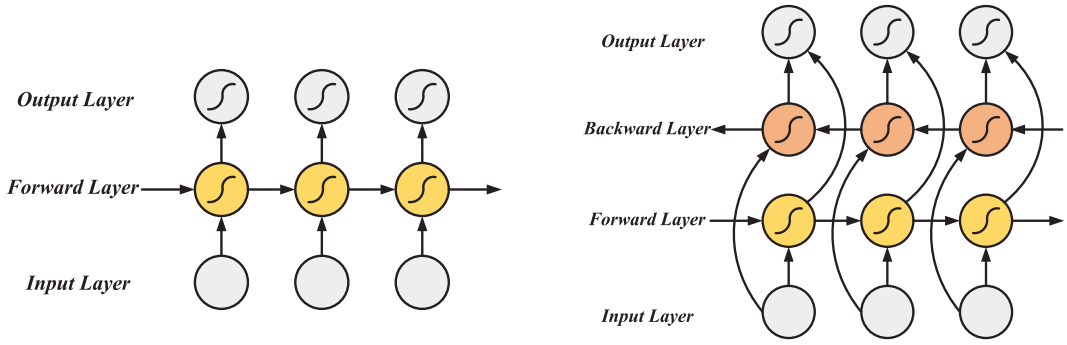
Combining BRNN with GRU gives BGRU, which can be used to access the long-term full sequential information of a given sequence in both directions. As a fault diagnosis problem can generally be viewed as a classification problem, cross entropy is adopted as the loss function. As to the sample weights, the weighted cross entropy is given by:

$$f(\theta) = -\sum_{n=1}^{N} w_n \sum_{i=1}^{M} y_i log(\hat{y}_i) \tag{16}$$

where θ denotes the neural network parameters, N denotes the number of samples, M denotes number of faults, y_i denotes the true label and $\hat{y}_i$ denotes the predicted probability.

In the training phase of BGRU, some tricks are adopted and described as follows:

- Batch Normalization [27]. Batch normalization is proposed to solve the covariate shift problem. In the training phase, data normalization is done at the intermediate layer after feeding a batch. The mean and variance of the intermediate layer will be 0 and 1 after batch normalization. The benefit of using batch normalization is that they can improve the generalization performance and speed up the training process.



(a) Conventional Recurrent Neural Networks (b) Bidirectional Recurrent Neural Networks

Fig. 2. Illustration of (a)RNN and (b) BRNN.

- Drop Out [28]. Drop out is proposed by Hinton et.al to avoid the overfitting of neural networks. During the forward propagation of neural networks, neurons will not work with a probability of *P*. Multiple sub-networks are considered to be trained when adopting dropout. This will result in a better generalization performance of the trained network.
- L1-norm Penalty [29]. For deep neural networks, smaller weights are encouraged since larger weights can lead to instability in the network. Usually, the L2-norm or L1-norm penalty in the loss function is preferred. In this paper, the L1-norm penalty is adopted to get smaller and sparser weights to avoid overfitting. Thus the loss function becomes:

$$f(\theta) = -\sum_{n=1}^{N} w_n \sum_{i=1}^{M} y_i log(\hat{y}_i) + \alpha \sum_{\theta} |\theta|_1 \tag{17}$$

where α is the trade-off parameter.

3.2. Cost sensitive active learning

Cost sensitive active learning is proposed to explore a large amount of unlabeled data while taking the class imbalance issue into consideration. Active learning (AL) mainly consists of two stages, namely query and labeling. The most commonly used query strategy for AL is uncertainty sampling while minimum confidence and minimum margin are the two common uncertainty strategies.

(1) Minimum confidence: this strategy selects the samples with the least confidence, that is to say, the strategy selects:

$$\arg\min_{D+} \sum_{x \in D+} P\left(y = f^D(x) | x, D\right) \tag{18}$$

where D denotes the labeled dataset and D_+ denotes the unlabeled dataset and

$$f^D(x) = \arg\max_{y} P(y|x, D) \tag{19}$$

denotes the most probable class of an unlabeled sample x.

(2) Minimum margin: this strategy selects the samples with the minimum difference of confidence between the most and the second most probable classes:

$$\arg\min_{D+} \sum_{x \in D+} \left(P\left(y = f^D(x) | x, D\right) - P\left(y = f^D_{second}(x) | x, D\right)\right) \tag{20}$$

$f^D(x)$ also denotes the most probable class and $f^D_{second}(x)$ denotes the second most probable class.

Taking a further look at the minimum confidence and minimum margins, the corresponding criteria for cost sensitive active learning can also be derived.

(1) Maximum expected cost reduction: the minimum confidence can be rewritten as:

$$D^+_{opt} = \arg\min_{D+} \sum_{x \in D+} \sum_{k=1}^{M} P(y = k|x) Err\left(k, f^D(x)\right) \tag{21}$$

Consider Err as the cost matrix C, we can get:

$$D^+_{opt} = \arg\max_{D+} \sum_{x \in D_+} \sum_{k=1}^{M} P(y = k|x) C(k, f^D_c(x)) \tag{22}$$

This is also called the maximum expected cost reduction, which can minimize the expected cost. The object of the cost-sensitive learning process can be represented as:

$$E_{cost}\left(f_C^{D \cup D^+}\right) = \sum_{x \in D \cup D^+} \sum_{k=1}^{M} P(y = k|x) C\left(k, f_C^{D \cup D^+}(x)\right)\Big) \tag{23}$$

(2) Cost-weighted minimum margin: this is similar to the maximum expected cost reduction. By replacing the confidence term with the expected cost, we can get the cost sensitive version of the minimum margin:

$$D^+_{opt} = \arg\min_{D+} \sum_{x \in D_+} \left(E^x_{cost}\left(f^D_{C,second}\right) - E^x_{cost}\left(f^D_C\right)\right) \tag{24}$$

The detailed derivation of the above selective criterion for cost sensitive active learning can be found in [6].

3.3. Evaluation index

The classification performance can be formulated by using a confusion matrix. For balanced classification problems, the accuracy and sensitivity are usually chosen as the key performance indexes, which are listed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{25}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{26}$$

It should be noted that the sensitivity index is also called fault detection rate(FDR) in the more general fault diagnosis field.

For imbalanced classification problem, G-mean is a popular evaluation index in general as it integrates the recalls of all categories [30], which is defined as:

$$G-mean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}} \tag{27}$$

G-mean tries to maximize the accuracy on each class while keeping these accuracy values balanced. Thus, a higher G-mean value indicates that the comprehensive performance of a classifier is better. For binary classification, G-mean is the square root of the product of sensitivity and specificity. For multi-class problems, it is a higher root of the product of sensitivity for each class.

4. The CSALBGRU framework for imbalanced fault diagnosis

In this paper, we propose a cost sensitive active learning model using a bidirectional gated neural network framework for imbalanced fault diagnosis. The offline stage mainly consists of two steps: the training of sample-sensitive bidirectional gated neural unit and cost sensitive active learning for querying unlabeled samples. Next, we first briefly introduce the training of standard BGRU, and the training of sample-sensitive BGRU is given on this basis. Then, the process of cost sensitive active learning is discussed. Finally, the proposed cost sensitive active learning model using a bidirectional gated neural unit framework is formulated.

In the training phase of BGRU, the orthogonal initialization method is adopted, and the Adam optimizer is used to train the network. In Algorithm 1 , the entire training procedure of BGRU is shown. The orthogonal initialization proposed in [31] is adopted as it can prevent vanishing and exploding gradients. When updating the network parameters, the Adam optimizer, which combines the advantages of two other popular methods, namely AdaGrad and RMSProp, is used [32]. When the performance achieved on the validation dataset cannot be improved any more, the learning rate is then delayed.

In Algorithm 2 , we show the training procedure of sample-sensitive BGRU. In this case, the sample weights need to be assigned. This can be done in the procedure below: first, we specify the class weights, and one of the recommended ways of doing so can be found in [33]; second, the sample weights are assigned according to the class weights and the effect of time. Suppose a fault occurs at time t, assuming $t=t_0$, then the weight of normal sample is:

$$w_x=\left(e^{\alpha_1(t_0-t)}+1\right)weight_{normal} \tag{28}$$

The weight of the fault sample is:

$$w_x=\left(e^{\alpha_2(t_0-t)}+1\right)weight_{fault} \tag{29}$$

where α_1 and α_2 denote the decay factors. This idea of assigning weights is mainly ascribed to the difficulty of distinguishing samples before and after a fault occurs. The closer to the time the fault occurred, the larger the weight of the sample should be. In this work, the weights of samples selected by the cost sensitive active learning are assigned the values of the class weights. A visual representation of the assigning process is shown in Fig. 3.

In order to explore the unlabelled data, the cost sensitive active learning is considered. The detailed procedure is shown in Algorithm 3 . The algorithm flow is similar to the ordinary active learning flow. First, we use the maximum expected cost reduction criterion to sample some instances from the unlabelled dataset. Then the labels of these selected will be assigned by human experts. The samples weights will be given according to their labels. In the next step, the selected samples will be added to the original training dataset to build an enhanced dataset. The enhanced dataset will then be used to train the model. The difference between our method and the classical active learning methods is in selecting the unlabelled data -in our method, class imbalance is taken into consideration. In this work, maximum expected cost reduction is adopted as the sampling strategy.

Based on the algorithms introduced above, the complete imbalanced fault diagnosis framework proposed in this work is shown in Fig. 4. The entire monitoring procedure is described as follows:

- Offline Modeling

1. Obtaining the training dataset from normal operation and fault conditions.

Algorithm 1: Learning Parameters θ of Bidirectional GRU

Input: train set (x, y), validation dataset (x_v, y_v), learning rate η, max epochs n
Output: Bidirectional GRU with learned parameters θ^*

Net ← construct _ BGRU();
θ ← initialize _ Net(Net); // Orthogonal initialization [31]
val _ err ← 1;
for $i=1; i\leqslant n$ **do** // number of epochs
  for $b=1; b\leqslant B$ **do** // number of batches
    $out_b \leftarrow$ forward-pass(x_b, Net, θ);
    $grad_\theta \leftarrow$backward-pass$(x_b, y_b, out_b, Net, \theta)$;
    $\theta^* \leftarrow$ update-NetParams$(Net, \theta, grad_\theta, \eta)$;
    $\theta \leftarrow \theta^*$;
  end
  $val\_err^* \leftarrow$ forward-pass (x_v, y_v, Net, θ);
  if $val\_err^* > val\_err$ **then**
    $\eta \leftarrow \eta * 0.01$; // Decrease step size
    $val\_err^* \leftarrow val\_err$;
  end
end

Algorithm 2: Learning Parameters W of Bidirectional GRU for Class Imbalance

Input: imbalanced train set $(x, y, sample\_weight)$, validation dataset (x_v, y_v), learning rate η, max epochs n
Output: Bidirectional GRU with learned parameters θ^*

Net ← construct _ BGRU();
θ ← initialize _ Net(Net); // Orthogonal initialization [31]
$val\_err \leftarrow 1$;
for $i=1; i\leqslant n$ **do** // number of epochs
  for $b=1; b\leqslant B$ **do** // number of batches
    $out_b \leftarrow$ forward-pass(x_b, Net, θ);
    $grad_\theta \leftarrow$ backward-pass $(x_b, y_b, out_b, Net, \theta, sample\_weight_b)$;
    $\theta^* \leftarrow$ update-NetParams $(Net, \theta, grad_\theta, \eta)$;
    $\theta \leftarrow \theta^*$;
  end
  $val\_err^* \leftarrow$ forward-pass(x_v, y_v, Net, θ);
  if $val\_err^* \leq val\_err$ **then**
    $\eta \leftarrow \eta * 0.01$; // Decrease step size
    $val\_err^* \leftarrow val\_err$;
  end
end

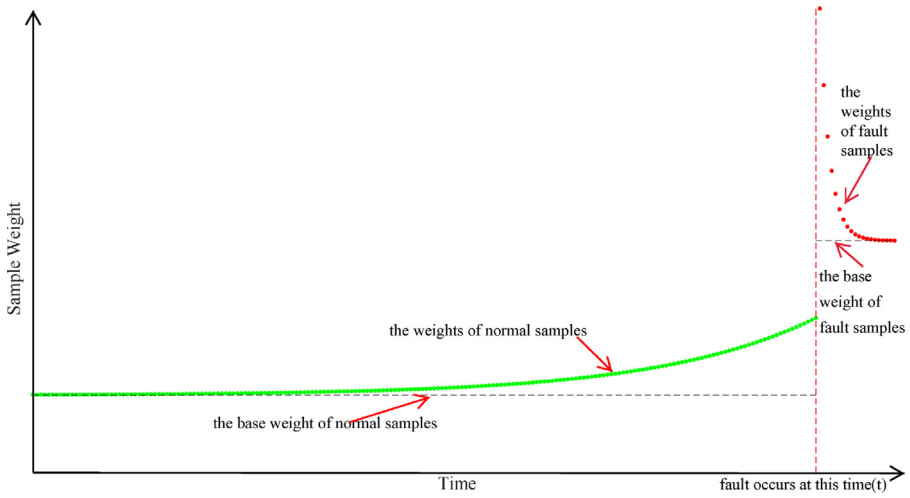


Fig. 3. Illustration of the process of assigning process weights for samples.

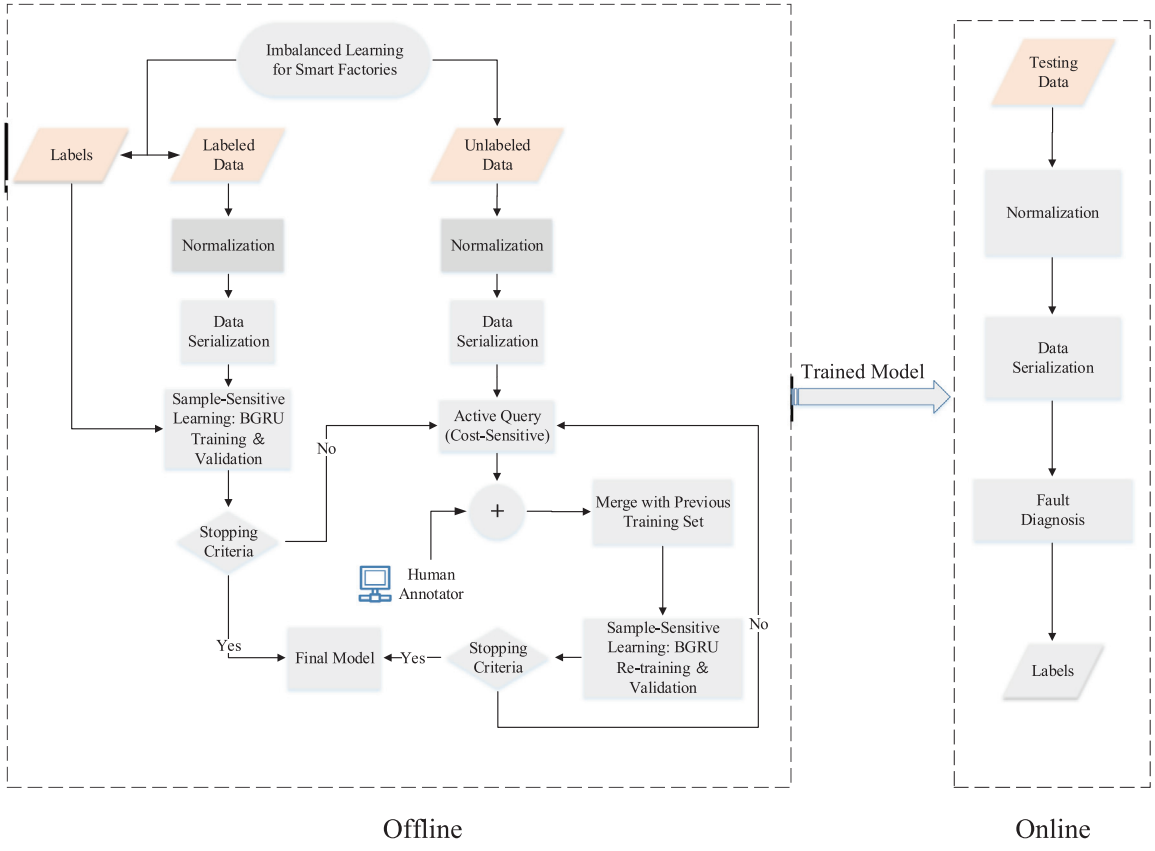


Fig. 4. Fault diagnosis procedure of the proposed CSALBGRU approach.

2. Performing normalization on the training data and combining the current data with the data from previous steps.
3. Assigning sample weights for each sample.
4. Training the sample-sensitive BGRU using the processed data.
5. Using cost sensitive active learning to select the unlabeled samples.
6. Annotating the selected samples by human experts and assigning corresponding sample weights.
7. Retraining the sample-sensitive BGRU.

- Online Fault Diagnosis

1. Processing the newly monitored samples in a manner similar to the one used in the training phase.
2. Getting the output y_t based on the trained BGRU model.
3. If $c = 0$, no faults occur; else classify the test data into the corresponding fault c, where c denotes the largest element of y_t.

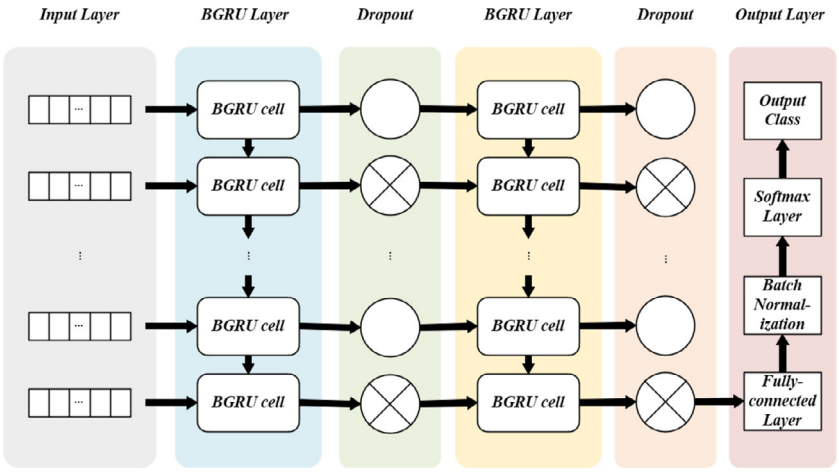


Fig. 5. Network structure.

Algorithm 3: Cost sensitive active learning using BGRU

Input: imbalanced train labelled dataset $(x_l, y_l, sample\_weight)$,imbalanced train unlabelled dataset (x_u, y_u), validation dataset (x_v, y_v)

Output: BGRU with learned parameters θ^*

$\theta^* \leftarrow$ Minimize-loss $f(\phi(x_l), y_l)$;// Imbalanced learning using Algorithm.2

for $i = 1; i \leqslant m$ **do** // number of query times

  $x^* \leftarrow$ sampling(x_u); // cost sensitive active learning using maximum expected cost reduction

  $y^* \leftarrow$ query(x^*) ;

  $x_l \leftarrow$ add(x_l, x^*) ; // add x^* to x_l

  $y_l \leftarrow$ add(y_l, y^*) ; // add y^* to y_l

  $sample\_weight \leftarrow$ add$(sample_weight, sample_weight^*)$;

  $\theta^* \leftarrow$ Minimize-loss $f(\phi(x_l), y_l)$;// Imbalanced learning using Algorithm.2

end

5. Experiment

5.1. Tennessee eastman benchmark

In this paper, the Tennessee Eastman(TE) Chemical benchmark process is firstly adopted to verify the effectiveness of the proposed method. TE is a simulation-based process using real data from a chemical engineering process [34]. The simulation model can be downloaded from the website: http://depts.washington.edu/control/LARRY/TE/download.html. A detailed explanation of the TE process can be found in [35].

5.1.1. Experiment setup

Normal data and fault data of the TE process are collected from the simulations on MATLAB 2016a. The sampling period is set to 36 seconds (i.e. 100 samples/h). For the training data, the simulator runs for 48 h in the normal state. 4800 normal training samples are then collected. In each simulation task of the 20 faults, the simulator also runs 48 h (i.e. 4800 training fault samples). It is noteworthy that the simulations of fault 6 shut down after 7 h in the fault state, hence each simulation for fault 6 only includes 7 h of fault data. For the testing data of each fault, the simulator runs for 8 h in the normal state at the beginning (i.e. 800 test normal samples). Then the corresponding fault disturbance is introduced and the simulator continues to run for 40 h. In this way, 40 h of fault data (i.e. 4000 test fault samples) are collected in each simulation. All samples will be used in a balanced scenario while some samples will be randomly selected according to the imbalance ratio and the proportion of unlabeled data for imbalanced fault diagnosis.

The experiment mainly consists of two parts. In the first part of this case study, BGRU is applied to diagnose the faults to show the necessity of considering the time series. In the second part of this case study, the proposed CSALBGRU method is applied to the diagnosis of imbalanced faults. The implementation is based the Python libraries Scikit-Learn [36], Imbalanced-Learn [37], Keras [38] and R Semi-Supervised Learning package [39]. The network structure we use is shown in Fig. 5. The first BGRU layer consists of 100 neurons and the second BGRU layer consists of 200 neurons. The network structure is determined after several structures have been tried. It can be seen that such a simple network structure can reach promising results. Adam is used for optimization with 0.001 initial learning rate and a 20-epoch training phase. The reported results are obtained after the experimental procedure is repeated for 10 times.

5.1.2. Comparison of different methods

In the first part of this case study, BGRU is applied to diagnosing the faults to show the necessity of considering time series. It should be noted that this sub-experiment is completed under the circumstance in which class balance exists. We compare BGRU with the other six methods to evaluate its effectiveness. Four widely used conventional methods for fault diagnosis, namely PCA, KPCA, one-class SVM and two-class SVM, are adopted to verify the superiority of BGRU. Meanwhile, the deep convolutional neural network(DCNN), which is a typical deep learning technique, is also chosen for the comparison. BLSTM, which shares the same architecture and training parameters with BGRU, is also used for comparison. BLSTM can further demonstrate the generalization performance of BGRU is better than BLSTM. For multi-class fault diagnosis, we compared the BGRU with SVM and DCNN. One versus one strategy is proved to be a more suitable way for practical use than the other methods [40]. It is then used to extend the two-class SVM to multi-class SVM.

In the second part of this case study, the proposed CSALBGRU method is applied to the diagnosis of imbalanced faults. In this sub-experiment, we will first set the imbalance ratio and the proportion of unlabelled data. And the training data selection procedure is as follows: (1) fault samples are randomly selected according to the imbalance ratio, i.e, the number of normal samples (4800) times imbalance ratio; (2) the corresponding proportion of

Table 1
Fault Detection Rates(%) for PCA, KPCA, One-class SVM, Two-class SVM, DCNN, BLSTM and BGRU(For Each Fault Condition, the Highest FDR is Marked in Bold).

fault	PCA [43]		KPCA [44]		One-class SVM [45]	Two-class SVM [46]	DCNN [3]	BLSTM	BGRU
	SPE	T^2	SPE	T^2					
1	99.5	99.1	**100**	99.3	99.8	99.5	98.6	99.8	99.8
2	98.4	98.5	99.0	95.3	98.6	99.4	98.5	99.2	**99.4**
3	0.6	3.6	6.8	9.0	-	-	91.7	88.2	**98.4**
4	98	21.8	100	9.5	99.6	63.4	97.6	100	**100**
5	21.7	25.7	30.3	29.9	100	90.8	91.5	99.8	**100**
6	99.9	98.9	100	81.0	100	60.1	97.5	99.3	**100**
7	99.9	99.9	100	100	100	98.9	99.9	100	**100**
8	96.8	97.4	97.9	97.4	97.5	96.0	92.2	98.6	**98.6**
9	1	3.4	6.5	5.4	-	-	58.4	85.4	**93.9**
10	15.4	36.7	52.5	48.6	87.6	81.0	96.4	99.1	**99.3**
11	63.8	41.4	77.6	51.0	69.8	80.2	98.4	98.5	**98.5**
12	92.5	98.5	98.5	98.9	**99.9**	97.8	95.6	97.8	98.8
13	95	94.3	95.2	94.3	95.5	92.5	95.7	98.4	**98.4**
14	99.9	98.8	**100**	99.6	100	91.0	98.7	99.8	99.8
15	0.7	3.5	8.8	9.3	-	-	28	56.2	**66.2**
16	13.7	17.4	48.0	33.6	**89.8**	89.4	44.2	76.4	80.1
17	90.5	78.7	95.9	81.5	**95.3**	81.6	94.5	95	95.1
18	90.1	89.3	90.6	90.0	90.0	89.5	**93.9**	89	89.5
19	5.9	11.5	43.1	8.8	83.9	85.9	**98.6**	74.9	83.4
20	42.3	34	59.8	49.1	90.0	80.5	93.3	97.2	**98**
Average	72	67.2	81.7	68.7	94	87.5	88.2	95.5	**96.6**

normal samples and fault samples is set to be unlabeled. For the test data selection, the number of fault samples is the number of normal samples(800) times imbalance ratio and these fault samples will be randomly selected.

We compare our proposed framework with four imbalanced learning methods and two semi-supervised learning methods. For imbalanced learning methods, as few researchers notice the class imbalance for fault diagnosis, two classical approaches, namely ADASYN [41]-SVM and RUSBoost [42], and two deep learning based methods, namely deep laplacian auto-encoder(DlapAE) and generative neural network(GAN) are tested. For the semi-supervised methods, as some researchers have applied the semi-fisher discriminant analysis (Semi-FDA) for fault diagnosis, it is selected for comparison as well. Another popular semi-supervised method, the transductive support vector machine(TSVM) is also employed in the simulation for comparison.

5.1.3. BGRU for balanced fault diagnosis

Table 1 shows that BGRU produces the best FDRs among these methods. The average FDR of BGRU is 96.6%, meaning an increase by 34.2%, 43.7%, 18.2%, 40.6%, 2.7%, 10.4%, 9.5% and 1.2%, compared with the SPE statistic of PCA, T^2 statistic of PCA, the SPE statistic of KPCA, T^2 statistic of KPCA, one-class SVM, two-class SVM, DCNN and BLSTM, respectively. It is also observed that the average FDR of BGRU is slightly higher than DCNN and BLSTM, so we use BGRU as the base classifier in the following discussion. Among the traditional methods, the performance of one-class SVM is the best and closest to that of BGRU. However, it is worth mentioning that the fault detection rates of fault 3, fault 9 and fault 15 for one-class SVM are not reported and it may indicate that the one-class SVM does not achieve good diagnostic results in these faults. The fault detection rates of fault 3, fault 9, fault 15 for BGRU are 98.4%, 93.9% and 66.2%, respectively.

We compare BGRU with multi-class SVM and DCNN for further analysis of their performances in multi-class faults diagnosis. In this case, fault 1, fault 2, fault 3, fault 8, fault 10, fault 11, fault 12, fault 13, fault 14, and fault 20 are selected. The two-dimensional codes for each class of BGRU is shown in Fig. 6. It is observed that BGRU produces an excellent visualization effect of the data. It is not hard to speculate that the softmax layer can get a promising classification result. The confusion matrices obtained

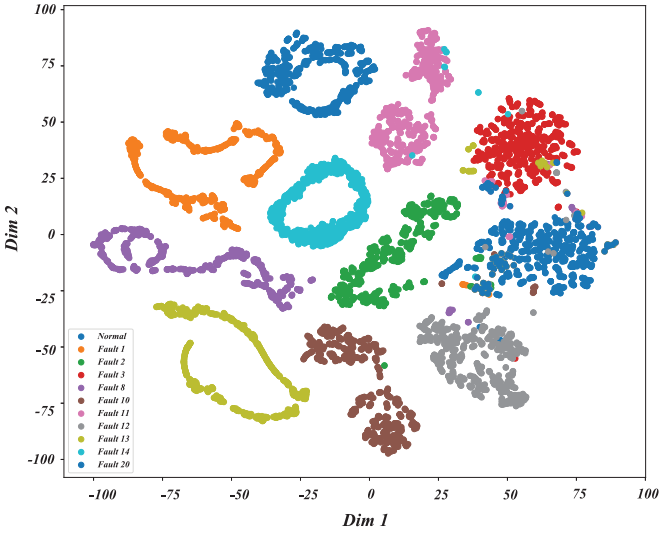
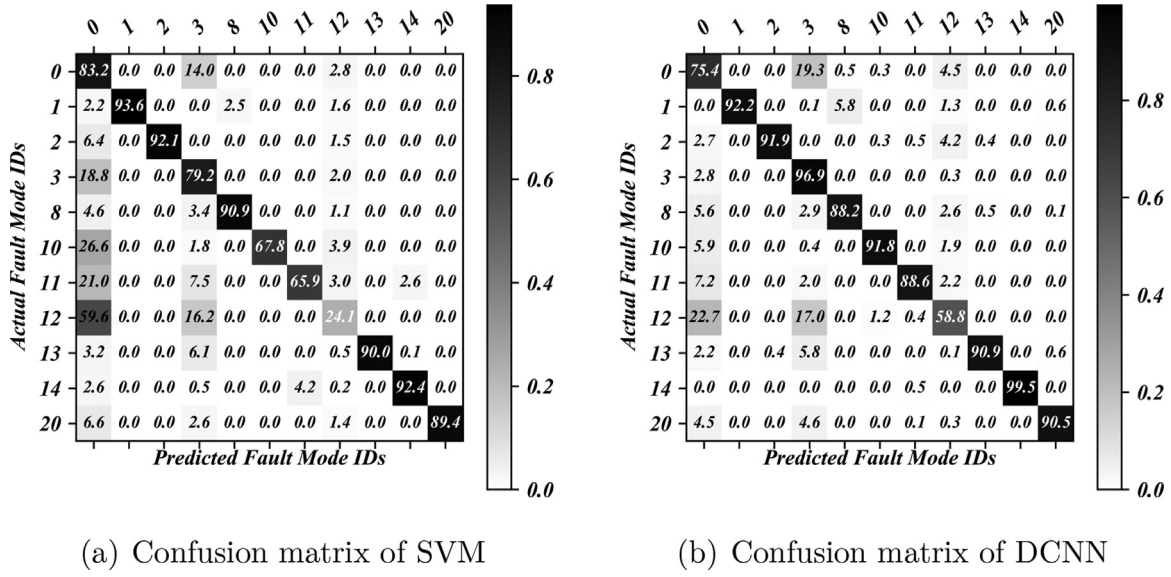


Fig. 6. BGRU-learned features. Specifically, we using t-sne reduce the dimension of the feature (input of the fully connected layer) to 2, and then plot them by class.

by SVM, DCNN and BGRU are illustrated in Figs. 7 and 8. For the confusion matrix, the rows show the predicted fault mode IDs, the columns show the actual fault mode IDs. The values on the diagonal of the matrix represent where the true fault and the predicted one match. The values on the non-matrix diagonal represent the percentage of instances where the algorithms have made mistakes. It is noticed that the accuracy value for each class of BGRU is all above 90% while the accuracy values of some classes of SVM and DCNN are much less than 90% (e.g. fault 12 of SVM, 24.12% and fault 12 of DCNN, 58.8%). The overall accuracy of BGRU is 95.5% while the overall accuracy values of SVM and DCNN are 79% and 87.7% - this shows a remarkable increase by 21% and 8.9%, respectively.

5.1.4. CSALGRU for imbalanced fault diagnosis

At the beginning of this sub-experiment, we set the imbalance ratio to be 0.5 and the proportion of unlabelled data is 0.6. We first prove the effectiveness of cost sensitive active learning using fault



(a) Confusion matrix of SVM (b) Confusion matrix of DCNN

Fig. 7. Confusion matrices of (a)SVM and (b)DCNN.

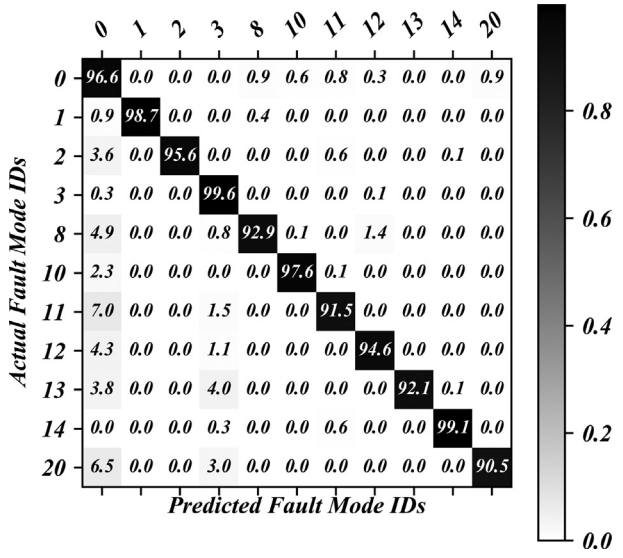


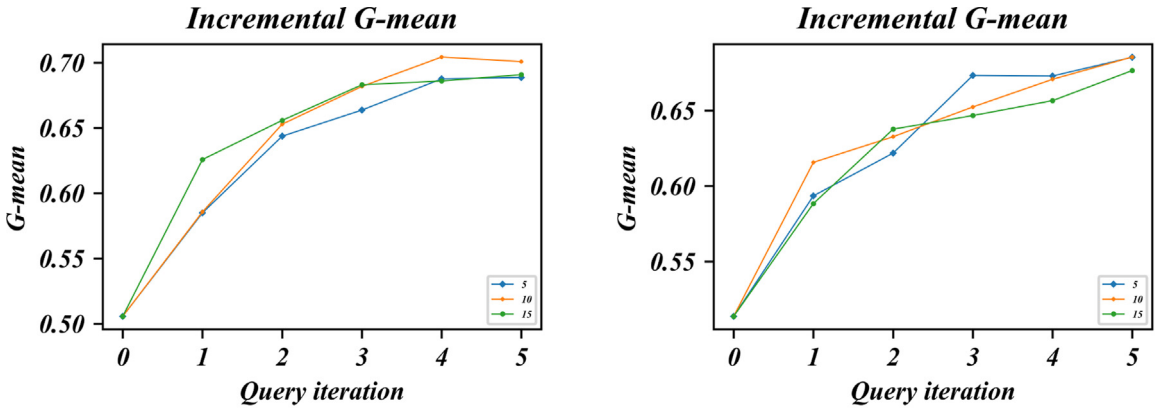
Fig. 8. Confusion matrix of BGRU.

4 and fault 15 as the test cases. We first compare the influence of the number of selected samples during each step of iteration on the experimental results. We chose 5,10 and 15 samples for one step of iteration. The detailed results are shown in Fig. 9. It can be found that the G-mean of fault 4 increases from 50% to 70% when 10 samples are selected in each query iteration while the G-mean increases from 50% to 68% when either 5 or 15 samples are selected in each iteration. This means the labeling cost can be reduced and the final performance can increase if an appropriate number is selected. Choosing to label more samples each time may grow faster in the first few iterations, but it would converge too quickly. It is also demonstrated that cost sensitive active learning can help improve the fault diagnosis performance greatly by using only a small amount of samples, i.e, a total of 50 samples in the experiment, accounting for 1% of the unlabelled data. That similar results can be found as well when fault 15 is tested.

Then the cost sensitive criterion is tested with the other three common criteria: entropy sampling, margin sampling and uncertainty sampling. We select 10 samples in each iteration. The comparison results are shown in Fig. 10. It can be seen that the proposed method obtains the highest G-mean value among all methods in each iteration. Take fault 4 as an example, the final G-mean of our method is about 70% while the other three methods are about 67%. In every query iteration, the G-mean of our method is higher than the other three methods. It is indicated that the proposed method precisely selects the most informative samples. The similar results can be found in the diagnosis of fault 15 as well.

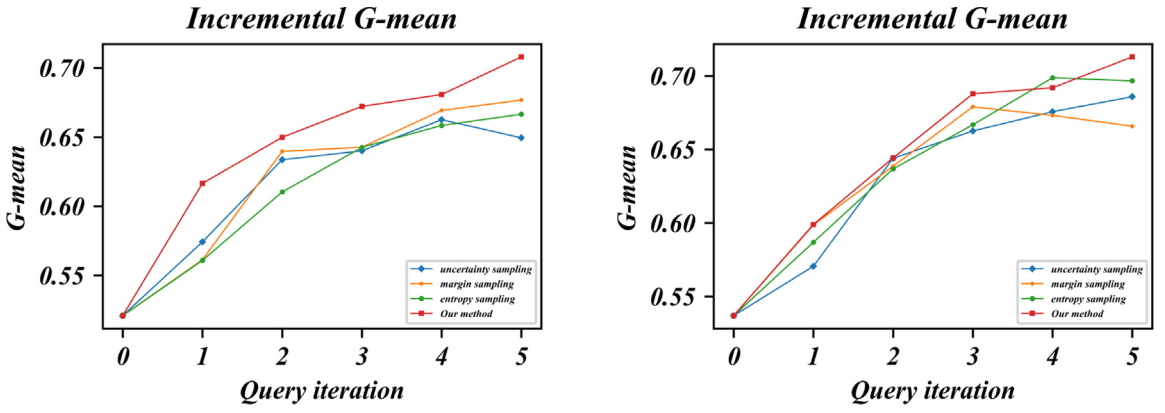
The comparison results of CSALBGRU and ADASYN-SVM, RUSboost, semi-FDA, TSVM, DlapAE and GAN are shown in Table 2. It is observed that the CSALBGRU shows remarkably improved performance in all 20 faults. It is noticed that the FDRS of ADASYN-SVM and RUSBoost are lower than CSALBGRU, and the reasons are twofold. First, the algorithms are limited to imbalanced learning. Second, the algorithms do not take advantage of unlabeled data. The FDRs of semi-FDA and TSVM are also lower than CSALBGRU and we also note that the FDRs for some faults are 0 as these methods prefer the majority classes. For the deep learning-based methods, DlapAE and GAN, their FDRs are higher than the traditional methods and lower than our proposed method in most faults. For the DlapAE based methods, they can learn a smooth manifold structure of data, but cannot solve the problem of imbalanced classes. And for GAN-based methods, the generated samples are of poor quality since the training process of GAN is not stable. These reasons can to some extent explain the reasons why the results obtained by these methods are not as good as those obtained by CSALBGRU.

We also compared the proposed techniques with ADASYN-SVM, RUSBoost, semi-FDA, TSVM, DlapAE and GAN under different imbalanced ratios. The proportion of fault 10 data to normal data is set from 0.1 to 0.6 with a step size of 0.1. The results are illustrated in Fig. 11, showing that the proposed CSALBGRU method maintains a significant advantage all the time compared with the traditional methods. It is indicated that the idea of weighting each sample and using cost sensitive active learning to explore unlabeled data is prone to improve the discrimination of minority faults. Compared with other methods, the performance of the proposed method is almost unaffected by the proportion of fault data. When the proportion of fault data is small, the semi-supervised methods fail to distinguish the minority fault completely while the imbalanced learning methods can deliver considerably good results. This further proves that the proposed method achieves the best performance. It should be noticed that the GAN based method obtains



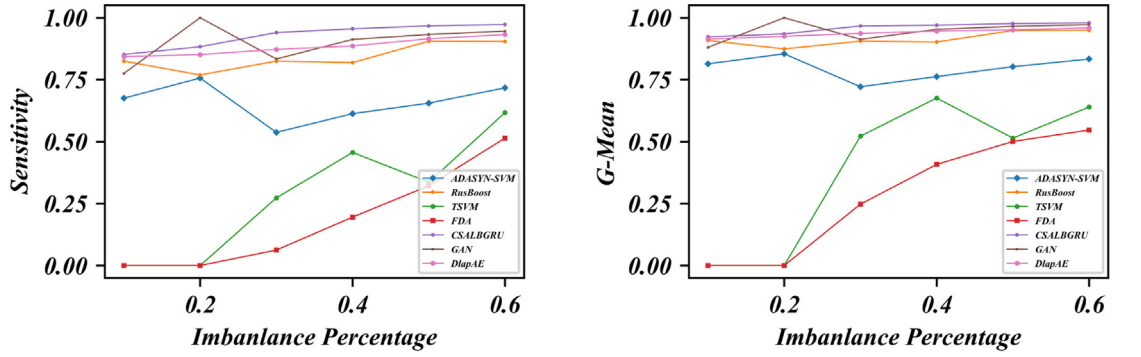
(a) The incremental G-mean of Fault 4 (b) The incremental G-mean of Fault 15

Fig. 9. The incremental G-mean of (a)Fault 4 and (b)Fault 15 when selecting different samples for each iteration.



(a) The incremental G-mean of Fault 4 (b) The incremental G-mean of Fault 15

Fig. 10. The incremental G-mean of (a)Fault 4 and (b)Fault 15 under different sampling criterion.



(a) Sensitivity in the 10th fault for different proportions (b) G-mean in the 10th fault for different proportions

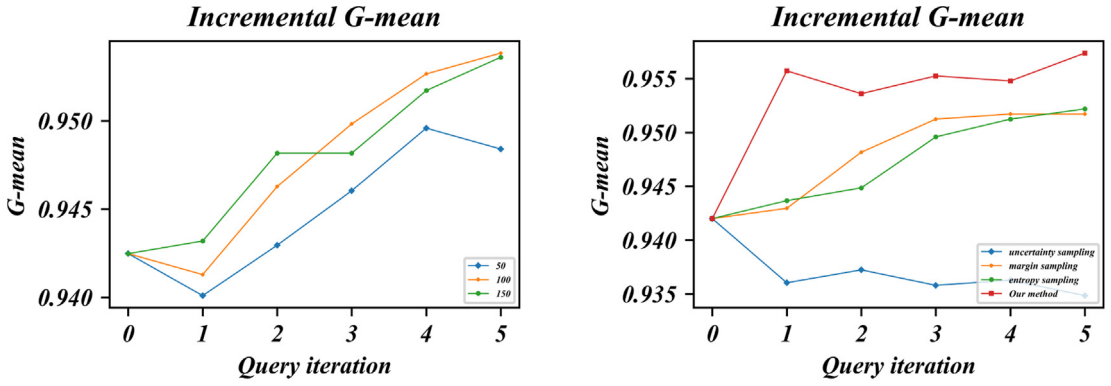
Fig. 11. Results in the 10th fault for different proportions.

the best performance when the imbanlance percentage equals to 0.2. It may indicate that the generated samples can help a lot when the training of GAN is reliable.

The proposed method is also applied to multi-class fault diagnosis. One versus one strategy is used to extend ADASYN-SVM and RUSboost for multi-class classification. The incremental G-mean of CSALBGRU is shown in Fig. 12. The results show that it is not always better to choose more samples for each iteration. A relatively large number would yield better results. In this case, selecting 100 samples in every iteration has better performance than selecting 50 or 150 samples in every iteration. The proposed method has the best performance compared to other three methods. This provides

Table 2
Fault Detection Rates(%) for ADYSYN-SVM,RUSBoost, Semi-FDA, TSVM and DLapAE, GAN and CSALBGRU(0.2)(For Each Fault Condition, the Highest FDR is Marked in Bold).

fault	ADASYN-SVM	RUSBoost	Semi-FDA [47]	TSVM	DLapAE [20]	GAN [21]	CSALBGRU
1	91.3	82.5	83.7	88.7	90.6	**100**	**100**
2	79.4	65	84	78.8	83.1	77.5	**98.9**
3	67.5	42.5	68.8	60	70.1	71.5	**88.9**
4	13.1	6.3	8	43.8	47.9	60.7	**100**
5	15	16.9	0	0	20.3	35.9	**100**
6	0	0.03	31.5	0	49.1	62.5	**70**
7	1.9	25.7	3.3	0	17.5	49.4	**100**
8	51.9	54.4	6.3	28.8	62.5	61.9	**98.1**
9	16.3	5	6.5	8.8	30.4	**82.4**	78.9
10	73.8	83.1	6.8	30	50	90	**93.8**
11	53.8	55	0	0	61.3	57.5	**95**
12	46.3	46.9	0.3	16.3	31.3	43.1	**90**
13	54.4	49.4	0	46.3	55.6	62.5	**98.4**
14	94.4	98.8	0.5	33.8	**100**	98.1	**100**
15	22.5	19.4	1.8	0	41.2	55.4	**56.3**
16	13.7	16.9	0.3	10	25.1	36.4	**87.5**
17	2.5	56.9	42	72.5	81.3	92.4	**95.7**
18	0	2.5	24	58.8	66.5	89.5	**91**
19	14.4	11.9	0.5	8.8	35.2	**52.6**	50
20	46.3	50	63.8	71.3	56.9	51.9	**97.5**



(a) The incremental G-mean when selecting different numbers of samples in each iteration

(b) The incremental G-mean under different sampling criterion

Fig. 12. The incremental micro G-mean.

further evidence of the effectiveness of the proposed method in terms of both binary fault diagnosis and multiclass fault diagnosis. The proposed method is further compared with ADASYN-SVM and RUSBoost. The confusion matrices are shown in Fig. 13 and 14, respectively. It can be seen that the classification accuracy of most faults is above 90% for CSALBGRU while almost no fault classification accuracy reaches above 90% for ADASYN-SVM and RUSBoost. The overall G-mean of CSALBGRU is 95.1%, meaning an increase by 29.5%, 21.3%, 11.4% and 17.3% compared with the ADASYN-SVM, RUSBOOST, DlapAE and GAN, respectively. It is worth mentioning that the proposed method shows better performance again. Hence the stability and robustness of the proposed method are further demonstrated.

5.2. A case study

In this section, a practical case study is used to further illustrate the effectiveness of the proposed method. The data are obtained from a plasma etching process in a real-world semiconductor packaging production line. A detailed description of the plasma etching process can be found at [48]. During the plasma etching process, a fault called the micro-arcing fault may occur. Its occurrence is mostly unnoticed and can seriously affect product quality. Therefore, it is necessary to detect faults for this process in a timely manner.

Table 3
The list of the main variables.

No.	Name	No.	Name
1	Gas flow	7	Voltage
2	APC pressure	8	Current
3	APC position	9	Pumping time
4	Source Power	10	Temperature
5	Source Power Reflect	11	Vacuum capacitor
6	RF bias power	12	Vacuum capacitor position

APC = Advanced Process Control

5.2.1. Dataset description

The process involves 12 key variables, as listed in Table 3. The dataset used in this case study contains 22,605 labeled samples, of which 19,762 are normal samples and 2843 are fault samples. In addition, there are also 2516 unlabeled samples, meaning that this is a typical imbalanced fault diagnosis problem. After

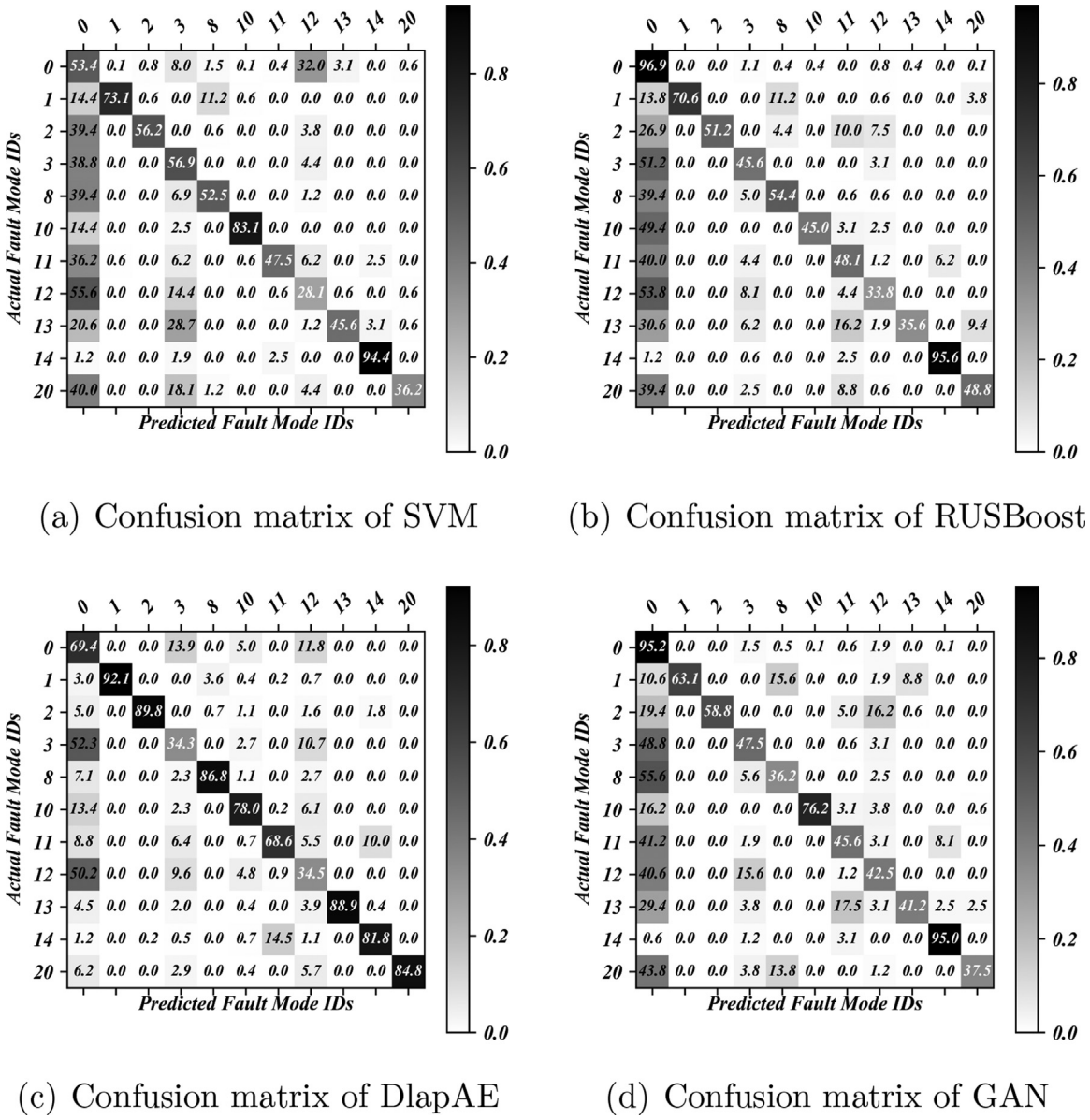


Fig. 13. Confusion matrices of (a)SVM (b)RUSBoost (c)DlapAE (d)GAN.

Table 4
Practical Results(%) for ADYSYN-SVM,RUSBoost, Semi-FDA, TSVM and DLapAE, GAN and CSALBGRU(The Highest Index is Marked in Bold).

		ADASYN-SVM	RUSBoost	Semi-FDA	TSVM	DLapAE	GAN	CSALBGRU
Train	Sensitivity	77.3	93.7	50.1	55.2	90.6	**97.9**	94.9
	G-mean	75.6	89.7	42.6	78.8	51.8	98.8	**99.3**
Test	Sensitivity	66.8	63.4	35.7	40.5	90.6	75.3	**94.8**
	G-mean	47.7	78.8	84	32.9	41.6	86.7	**98.7**

normalization, we also randomly choose 80% of the labeled samples and all unlabelled samples as the training samples for imbalanced fault diagnosis. The last 20% of the labeled samples are used for testing.

5.2.2. Experimental analysis

We compare our proposed method with six methods, namely ADASYN-SVM, RUSBoost, Semi-FDA, TSVM, DlapAE and GAN. The network structure is the same as described in Fig. 5. The results of comparison are shown in Table 4. It is observed that the proposed CSALBGRU achieves the best performance among these methods in both the training dataset and test dataset. It is worth mentioning the fault diagnosis performance of the GAN based method on the test set is not as good as that on the training set. This indicates that the GAN based methods may suffer from the overfitting problem.

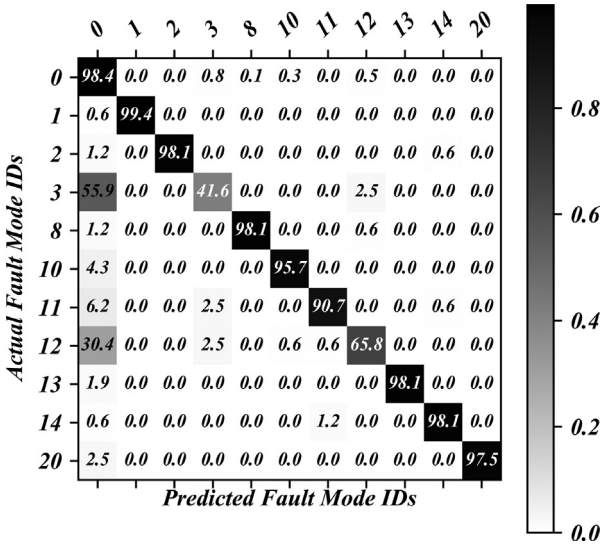


Fig. 14. Confusion matrix of CSALBGRU.

6. Conclusions and future work

In this paper, a cost sensitive active learning using bidirectional gated recurrent neural network is proposed for imbalanced fault diagnosis. The dynamic behaviour of the process is first tackled using a bidirectional gated recurrent neural network. Then sample sensitive learning is developed to reduce the influence of class imbalance. Finally, cost sensitive active learning is utilized to explore the unlabeled data so that waste of data can be eliminated.

The effectiveness of the proposed technique is evaluated in a set of computational experiments using the Tennessee Eastman (TE) dataset and a practical dataset. It is demonstrated in the experiments that the proposed method possesses a significant improvement compared to the existing methods. The authors' future work will be focused on two main areas. First, deep learning techniques for semi-supervised learning will be studied for using the unlabeled data more effectively. Second, some researchers have begun to pay attention to the cases where fault samples are missing [49,50]. For example, Hu et al. proposed a novel incremental imbalance modified deep neural network (incremental-IMDNN) to address the continuous arrival of new fault modes [50]. Xu et al. proposed the renewable fusion fault diagnosis network (RFFDN) to handle the extreme case. These two methods have the common idea of sharing weights, meaning that transfer learning based methods may achieve promising performance in the case. This will be a particular area that the authors will explore as well.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Hongwei Wang: Writing - review & editing, Funding acquisition. **Heming Zhang:** Funding acquisition, Supervision.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1701602, in part by the State Key Laboratory of Intelligent Manufacturing Systems Technology under Grant QYYE1601, and in part by the National Natural Science Foundation of China under Grant 61374163. This work was also supported in part by the Zhejiang University/University of Illinois at Urbana-Champaign Institute, and was led by Principal Supervisor Prof. Hongwei Wang.

References

[1] H. Kodamana, R. Raveendran, B. Huang, Mixtures of probabilistic PCA with common structure latent bases for process monitoring, IEEE Trans. Control Syst. Technol. (99) (2017) 1–9.
[2] L.H. Chiang, M.E. Kotanchek, A.K. Kordon, Fault diagnosis based on fisher discriminant analysis and support vector machines, Comput. Chem. Eng. 28 (8) (2004) 1389–1401.
[3] H. Wu, J. Zhao, Deep convolutional neural network model based chemical process fault diagnosis, Comput. Chem. Eng. 115 (2018) 185–197.
[4] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[5] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 6645–6649.
[6] P.-L. Chen, H.-T. Lin, Active learning for multiclass cost-sensitive classification using probabilistic models, in: Proceedings of the Conference on Technologies and Applications of Artificial Intelligence, IEEE, 2013, pp. 13–18.
[7] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv:1412.3555 (2014).
[8] C. Tong, T. Lan, X. Shi, Ensemble modified independent component analysis for enhanced non-gaussian process monitoring, Control Eng. Pract. 58 (2017) 34–41.
[9] F. Harrou, Y. Sun, M. Madakyaru, B. Bouyedou, An improved multivariate chart using partial least squares with continuous ranked probability score, IEEE Sens. J. 18 (16) (2018) 6715–6726.
[10] L. Xiao, H. Sun, L. Zhang, F. Niu, L. Yu, X. Ren, Applications of a strong track filter and LDA for on-line identification of a switched reluctance machine stator inter-turn shorted-circuit fault, Energies 12 (1) (2019) 134.
[11] Y. Dong, S.J. Qin, A novel dynamic pca algorithm for dynamic data modeling and process monitoring, J. Process Control 67 (2018) 1–11.
[12] Y. Xu, X. Deng, Fault detection of multimode non-gaussian dynamic process using dynamic bayesian independent component analysis, Neurocomputing 200 (2016) 70–79.
[13] X. Xu, Q. Liu, J. Ding, A modified dynamic pls for quality related monitoring of fractionation processes, IFAC-PapersOnLine 51 (18) (2018) 315–320.
[14] G. Rong, S.-Y. Liu, J.-D. Shao, Dynamic fault diagnosis using extended matrix and tensor locality preserving discriminant analysis, Chemometr. Intell. Laborat. Syst. 116 (2012) 41–46.
[15] I. Yélamos, G. Escudero, M. Graells, L. Puigjaner, Performance assessment of a novel fault diagnosis system based on support vector machines, Comput. Chem. Eng. 33 (1) (2009) 244–255.
[16] L. Jiang, Z. Ge, Z. Song, Semi-supervised fault classification based on dynamic sparse stacked auto-encoders model, Chemomet. Intell. Laborat. Syst. 168 (2017) 72–83.
[17] S. Lee, M. Kwak, K.-L. Tsui, S.B. Kim, Process monitoring using variational autoencoder for high-dimensional nonlinear processes, Eng. Appl. Artif. Intell. 83 (2019) 13–27.
[18] F. Cheng, Q.P. He, J. Zhao, A novel process monitoring approach based on variational recurrent autoencoder, Comput. Chem. Eng. 129 (2019) 106515.
[19] X. Li, W. Zhang, Q. Ding, X. Li, Diagnosing rotating machines with weakly supervised data using deep transfer learning, IEEE Trans. Ind. Inf. (2019).
[20] X. Zhao, M. Jia, M. Lin, Deep laplacian auto-encoder and its application into imbalanced fault diagnosis of rotating machinery, Measurement 152 (2020) 107320.
[21] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, X. Li, Machinery fault diagnosis with imbalanced data using deep generative adversarial networks, Measurement 152 (2020) 107377.
[22] F. Zhou, S. Yang, H. Fujita, D. Chen, C. Wen, Deep learning fault diagnosis method based on global optimization gan for unbalanced data, Knowl. Based Syst. 187 (2020) 104837.
[23] P. Jiang, Z. Hu, J. Liu, S. Yu, F. Wu, Fault diagnosis based on chemical sensor data with an active deep neural network, Sensors 16 (10) (2016) 1695.
[24] L. Yin, H. Wang, W. Fan, L. Kou, T. Lin, Y. Xiao, Incorporate active learning to semi-supervised industrial fault classification, J. Process Control 78 (2019) 88–97.
[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078(2014).
[26] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.
[27] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.
[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[29] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.

[30] S. Barua, M.M. Islam, X. Yao, K. Murase, Mwmote–majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans. Knowl. Data Eng. 26 (2) (2012) 405–425.
[31] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, ICLR 2015: International Conference on Learning Representations 2015, 2015. Sourced from Microsoft Academic https://academic.microsoft.com/paper/2964121744.
[32] A.M. Saxe, J.L. McClelland, S. Ganguli, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, ICLR 2014: International Conference on Learning Representations (ICLR) 2014, 2014 Sourced from Microsoft Academic https://academic.microsoft.com/paper/2963504252.
[33] G. King, L. Zeng, Logistic regression in rare events data, Political Anal. 9 (2) (2001) 137–163.
[34] P.R. Lyman, C. Georgakis, Plant-wide control of the tennessee eastman problem, Comput. Chem. Eng. 19 (3) (1995) 321–331.
[35] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (3) (1993) 245–255.
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.
[37] G. Lemaître, F. Nogueira, C.K. Aridas, Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning, J. Mach. Learn. Res. 18 (1) (2017) 559–563.
[38] F. Chollet, et al., Keras, 2015, (????).
[39] J.H. Krijthe, Rssl: Semi-supervised learning in r, in: Proceedings of the International Workshop on Reproducible Research in Pattern Recognition, Springer, 2016, pp. 104–115.
[40] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[41] H. He, Y. Bai, E.A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.
[42] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: a hybrid approach to alleviating class imbalance, IEEE Trans. Syst. Man Cybern. Part A Syst. Humans 40 (1) (2009) 185–197.
[43] T.J. Rato, M.S. Reis, Fault detection in the tennessee eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (dpca-dr), Chemomet. Intell. Labor. Syst. 125 (2013) 101–108.
[44] J. Fan, Y. Wang, Fault detection and diagnosis of non-linear non-gaussian dynamic processes using kernel dynamic independent component analysis, Inf. Sci. (Ny) 259 (2014) 369–379.
[45] S. Mahadevan, S.L. Shah, Fault detection and diagnosis in process data using one-class support vector machines, J. Process Control 19 (2009) 16277-1639.
[46] S. Yin, X. Gao, H.R. Karimi, X. Zhu, Study on support vector machine based fault detection in tennessee eastman process, in: Proceedings of the Abstract and Applied Analysis, Hindawi, 2014.
[47] S. Zhong, Q. Wen, Z. Ge, Semi-supervised fisher discriminant analysis model for fault classification in industrial processes, Chemomet. Intell. Labor. Syst. 138 (2014) 203–211.
[48] K.A. Reinhardt, R.F. Reidy, Handbook for Cleaning for Semiconductor Manufacturing: Fundamentals and Applications, 67, John Wiley & Sons, 2011.
[49] K. Xu, S. Li, X. Jiang, Z. An, J. Wang, T. Yu, A renewable fusion fault diagnosis network for the variable speed conditions under unbalanced samples, Neurocomputing 379 (2020) 12–29.
[50] Z. Hu, P. Jiang, An imbalance modified deep neural network with dynamical incremental learning for chemical fault diagnosis, IEEE Trans. Ind. Electron. 66 (1) (2018) 540–550.

Peng Peng is currently a doctoral student at the National Engineering Research Centre of Computer Integrated Manufacturing System (CIMS-ERC) in Tsinghua University, Beijing, China. He received his Bachelor degree at Department of Automation from Northeastern University in 2016. His research interests are process monitoring and prognostic and health management.

Wenjia Zhang is currently a doctoral student at the National Engineering Research Centre of Computer Integrated Manufacturing System (CIMS-ERC) in Tsinghua University, Beijing, China.Â She received her Bachelor degree at Department of Automation from Beihang University in 2018. Her research interests are process monitoring and prognostic and health management.

Yi Zhang is currently a doctoral student at the National Engineering Research Centre of Computer Integrated Manufacturing System (CIMS-ERC) in Tsinghua University, Beijing, China. He received his Bachelor degree at Department of Automation from Naval Research Institute in 2010. His research interests are process monitoring and quality prediction.

Yanyan Xu is currently a Master Degree Candidate at the National Engineering Research Centre of Computer Integrated Manufacturing System (CIMS-ERC) in Tsinghua University, Beijing, China. She received her bachelor of engineering degree from school of information engineering, wuhan university of technology in 2008. Her research focuses on product platform construction, product lifecycle research, and text mining.

Hongwei Wang received the B.S. degree in information technology and instrumentation from Zhejiang University, China, in 2004, the M.S. degree in control science and engineering from Tsinghua University, China, in 2007, and the Ph.D. degree in design knowledge retrieval from the University of Cambridge. From 2011 to 2018, he was a Lecturer and then, a Senior Lecturer in engineering design with the University of Portsmouth. He is currently a Tenured Associate Professor with Zhejiang University and the University of Illinois Urbana-Champaign Institute. His research interests include the application of intelligent and computing technologies to address specific issues in engineering systems design, manufacture and maintenance, for example, knowledge and information management, collaborative product development, collaborative modeling, and simulation and fault diagnosis. His research work, in these areas, has led to the over 100 peer-reviewed publications.

Heming Zhang is currently a professor at the National Engineering Research Center of Computer Integrated Manufacturing System (CIMS-ERC) in Tsinghua University, China. He received his Ph.D. degree in Mechanical Engineering from Zhejiang University, China in 1995. He joined the faculty of Department of Automation in Tsinghua University after completing a two-year Postdoctoral Fellowship in CIMS-ERC. His research interests include multidisciplinary modeling and collaborative simulation, service-oriented modeling and simulation, concurrent and collaborative design.