IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2017, accepted February 5, 2018, date of publication February 19, 2018, date of current version March 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2807121

An Integrated Ensemble Learning Model for Imbalanced Fault Diagnostics and Prognostics

ZHENYU WU[1], WENFANG LIN[2], AND YANG JI[2]
[1]Engineering Research Center of Information Network, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Zhenyu Wu (shower0512@bupt.edu.cn)

ABSTRACT With the development of artificial intelligence technology, data-driven fault diagnostics and prognostics in industrial systems have been a hot research area since the large volume of industrial data is being collected from the industrial process. However, imbalanced distributions exist pervasively between faulty and normal samples, which leads to imprecise fault diagnostics and prognostics. In this paper, an effective imbalance learning algorithm Easy-SMT is proposed. Easy-SMT is an integrated ensemble-based method, which comprises synthetic minority oversampling technique (SMOTE)-based oversampling policy to augment minority faulty classes and EasyEnsemble to transfer an imbalanced class learning problem into an ensemble-based balanced learning subproblem. We validate the feasibility and effectiveness of the proposed method in a real wind turbine failure forecast challenge, and our solution has won the third place among hundreds of teams. Moreover, we also evaluate the method on prognostics and health management 2015 challenge datasets, and the results show that the model could also achieve good performance on multiclass imbalance learning task compared with baseline classifiers.

INDEX TERMS Industrial prognostics, class-imbalance learning, machine learning, ensemble learning.

I. INTRODUCTION

Fault diagnostics and prognostics plays a critical role in industrial system as more and more sensor data are being collected from the industrial process, and it has attracted increasing attention in both academic and industrial fields. A robust and accurate fault diagnostic and prognostic system helps preventing fatal accident, saves costs and increases manufacturing efficiency [1]. However, the complexity of modern industrial systems sets obstacles for devising a practical model and faults are difficult to be diagnosed without any expert's knowledge of how the failures happen. Thus data-driven methods tend to solve the problem of failure prognostics in industrial field by learning the failing patterns of machines from observed data [2].

Several open challenges have been organized to call for fault diagnostic and prognostic solutions. For instance, wind turbine's blade suffers from freezing problem in wind electricity generation system, which has been a global challenge in industrial field. Predicting the early stage of the freezing events precisely is valuable for wind turbine maintenance, which decrease the failure risks and save the maintaining cost dramatically. The challenge held by China Academy of Information and Communication Technology (CAICT) aims at predicting the freezing durations according to the SCADA data of a certain wind farm management. The objective is similar in 2015 PHM Challenge, which is to predict 6 different faults at plant level according to a volume of operational data. The key challenge of analyzing these data is that different degrees of imbalance exists between failure and normal data pervasively, which has a serious influence on the performance of predicting models [3]. The abnormal data of each fault are extremely imbalanced with high imbalance ratio compared to normal data. Therefore, it is very difficult for us to correctly classify the abnormal and normal states of the industrial system. According to these challenges, there are some notable work for failure prognostics in industrial system and class-imbalance learning models.

Isermann [4] uses traditional model-based methods to detect fault and diagnosis the status by the related experience and expertise. S. Yin *et al.* [5] design a data-driven methods of robust fault detection system for wind turbines. Many machine learning algorithms used in pattern classification are now being utilized in fault detection. For example, Bayesian network [6] and fuzzy-logic [7] are two powerful methods that have been used to detect faults in railway traction device and mechanical systems. Fisher's discriminant analysis [8] and artificial neural networks [9] are also widely applied to detect failure. Meanwhile, random forest and gradient boosted tree [10] are used to fault prognostics in aircraft systems. Moreover, several approaches for classification using imbalanced data have been researched. Undersampling and oversampling [11] can reduce the level of imbalance and both sampling methods are helpful in imbalanced data problems [12], [13]. In addition, Synthetic Minority Oversampling TEchnique (SMOTE) [14] is a synthetic technique, which can added new minority class examples. Chan and Stolfo [15] introduce an approach to explore majority class examples, they split the majority class into several non-overlapping subsets, and finally ensemble classifiers using stacking. Liu *et al.* [16] propose EasyEnsemble and BalanceCascade algorithms to overcome the class-imbalance learning.

Consequently, this paper proposes an integrated ensemble-based imbalance learning model for industrial prognostics. The model is aimed at predicting failure in industrial system, and using an ensemble algorithm to overcome the classification of imbalanced data. The contributions of this paper are summarized below:

1) We propose the Easy-SMT ensemble algorithm based on synthesizing SMOTE-based data augmentation policy and EasyEnsemble algorithm, which can overcome both binary and multi-class imbalance problems.
2) We evaluate our model in real cases from a wind turbine freezing failure forecast contest and PHM 15 challenge. The result shows that the model could predict the *start_time* and *end_time* of wind turbine freezing failure with a relative high performance and achieve third place among 830 teams. Meanwhile, the model could also achieve good performance of both binary and multi-class imbalance classifications.

The remainder of the paper is organized as follows: In Section II, we formally define the problem to be solved. And our integrated method for industrial prognostics is proposed in Section III. In section IV, we use the real data sets to validate the feasibility and effectiveness of our model and algorithm. Finally, conclusions are drawn and future work are presented in section V.

II. PROBLEM STATEMENT

A. CLASS-IMBALANCE LEARNING FOR TIME SERIES

Many of the industrial processes usually operate in the normal state. Thus, it is very common for the diagnostic systems to collect a large number of samples of the normal state in the batch of data, while only a very few faulty samples could be collected in practice. Various diagnostic schemes have been applied to industrial processes, however, detecting faults under the class-imbalance condition is a challenging task with growing attention from both academia and industry.

An imbalanced dataset can be described as a set of samples, in which the proportion of the representative samples of one class is significantly larger than other class. The amount of this proportion brings up the definition of the ?imbalance ratio?, which is an important factor in selecting a proper classification technique. The imbalance ratio indicates the collected data are highly imbalance, moderate or low. The major class in an imbalance dataset referred to a class with more number of samples, while the minor class is often the class of interest and should be detected with high accuracy. In industrial process, the faulty samples are usually minor classes compared with normal samples with different imbalance ratios.

To deal with class-imbalance learning problem, two main strategies are usually used: data-level and algorithm-level methods. **Data-level** method [17], [18] is to change the class distribution of imbalanced data by sampling policies. Under-sampling and over-sampling [19] are two common methods. Mani and Zhang [20] indicate that the random under-sampling strategy usually outperformed some other complicated under-sampling strategies. In addition, SMOTE [18] is a synthetic over-sampling technique, which can added new minority class examples. Han *et al.* [21] proposed the borderline-SMOTE to over-sample the minority class near the borderline. Xie and Qiu [22] showed that over-sampling usually perform better than under-sampling. Estabrooks *et al.* [23] and Barandela *et al.* [24] both suggested that a combination of over-sampling and under-sampling might be more effective to solve the class imbalance problem. However, it is argued that sampling method leads to overfitting or drop some useful features of majority classes. **Algorithm-level** method [17] is to adjust the classifier to imbalance data. Bagging and Boosting ensemble-based method have been widely used. Seiffert *et al.* [25] conducted a comprehensive study comparing sampling methods with boosting for improving the performance of decision trees model built for identifying the software defective modules. Their results showed that sampling methods were effective in improving the performance of such models while boosting outperformed even the best data sampling methods. Chawla *et al.* [26] proposed a novel approach SMOTEBoost for learning from imbalanced datasets on the basis of the SMOTE algorithm and the boosting procedure. Seiffert *et al.* [27] presented a different hybrid ensemble methods named RUSBoost, which combined the random under-sampling strategy with the boosting procedure. Liu *et al.* [16] proposed the EasyEnsemble method which change the imbalance learning problem into several balance classification tasks with ensemble strategies. The idea is based on a two fold of ensembles that under-samples the majority class without information loss. Adaboost [28] is used to train the weak classifiers.

Consequently, the fault diagnostic and prognostic task for industrial system could be modeled as class-imbalance classification problem. While the raw data are sequential time series, which are able to be segmented into time windows by sliding window mechanism, and each time window could be labeled as normal or different types of faults. Commonly, the faulty time windows are minor class, while the normal time windows are major class. Predicting the faults is mainly

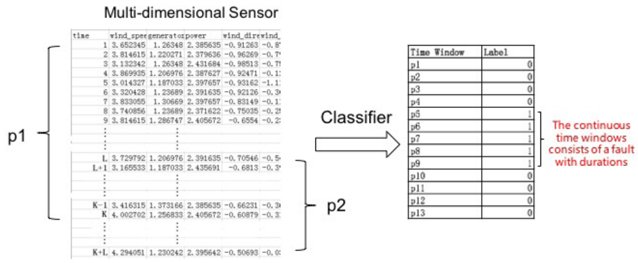


FIGURE 1. Problem formulation: imbalance-class classification for time windows.

based on classifying the faulty time windows from the normal ones. The diagram of problem formulation is introduced in Figure 1, where $p1$ and $p2$ are the time windows segmented by sliding window mechanism.

B. DESIGN CRITERIA

To better understand the data with its features, an initial statistic analysis is supposed to be given. For example, the duration of the faults tend to be analyzed in training datasets and it will give a reference to determine the length of the time windows that covers the features of the faults. Moreover, the imbalance ratio tend to be calculated as a reference to the determination of sampling policies and baseline classifiers. To evaluate the performance on class-imbalance classification, accuracy is not an appropriate evaluation metric when the datasets are class imbalance. In this paper, we use Recall, Precision, F-measure, ROC and AUC as performance evaluation metrics. We take the minority class (failure samples) as positive class and majority class (normal samples) as negative class.

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Recall is the percentage of true fault events which are labelled. Precision is the percentage of predicted fault events which are correctly labelled. F-measure is the weighted harmonic mean of recall and precision, which is the function of confusion matrix. ROC (Receiver operating characteristic curve) is another tool for measuring the imbalanced data in classification, which is a comprehensive index reflecting the continuous variables of sensitively and specificity. One of the indicators for comparing different ROC curves is the area under the curve, and AUC shows the average performance of the classifier for imbalanced and cost-sensitive problems.

Generally speaking, following criteria will guide us to design the learning framework:

1) Features extracted from raw sensor data should represent the faulty data as many as possible;
2) Baseline classifier should be suitable for imbalance data and optimization tend to be designed based on the baseline model;

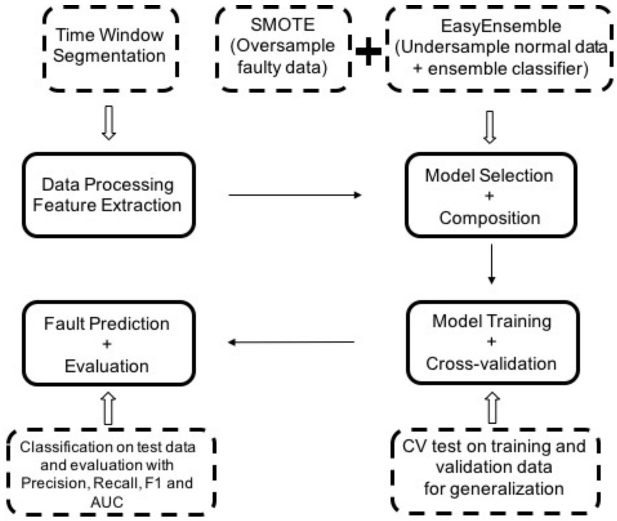
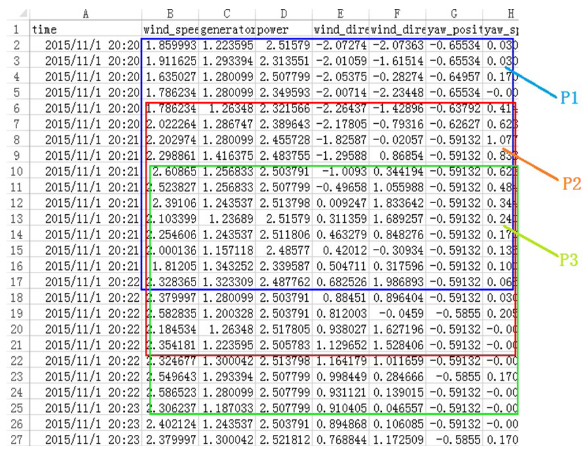


FIGURE 2. Imbalance learning framework for fault diagnostics and prognostics.



	A	B	C	D	E	F	G	H
1	time	wind_spee	generator	power	wind_dire	wind_dire	yaw_posit	yaw_sp
2	2015/11/1 20:20	1.859993	1.223595	2.51579	-2.07274	-2.07363	-0.65534	0.030
3	2015/11/1 20:20	1.911625	1.293394	2.313551	-2.01059	-1.61514	-0.65534	0.030
4	2015/11/1 20:20	1.635027	1.280099	2.507799	-2.05375	-0.28274	-0.64957	0.170
5	2015/11/1 20:20	1.786234	1.280099	2.349593	-2.00714	-2.23448	-0.65534	-0.00
6	2015/11/1 20:20	1.786234	1.26348	2.321566	-2.26437	-1.42896	-0.63792	0.41
7	2015/11/1 20:20	2.022264	1.286747	2.389643	-2.17805	-0.79316	-0.62627	0.62
8	2015/11/1 20:21	2.202974	1.280099	2.455728	-1.82587	-0.02057	-0.59132	1.07
9	2015/11/1 20:21	2.298861	1.416375	2.483755	-1.29588	0.86854	-0.59132	0.83
10	2015/11/1 20:21	2.60865	1.256833	2.503791	-1.0093	0.344194	-0.59132	0.62
11	2015/11/1 20:21	2.523827	1.256833	2.507799	-0.49658	1.055988	-0.59132	0.48
12	2015/11/1 20:21	2.39106	1.243537	2.513798	0.009247	1.833642	-0.59132	0.34
13	2015/11/1 20:21	2.103399	1.23689	2.51579	0.311359	1.689257	-0.59132	0.24
14	2015/11/1 20:21	2.254606	1.243537	2.511806	0.463279	0.848276	-0.59132	0.17
15	2015/11/1 20:21	2.000136	1.157118	2.48577	0.42012	-0.30934	-0.59132	0.13
16	2015/11/1 20:21	1.81205	1.343252	2.339587	0.504711	0.317596	-0.59132	0.10
17	2015/11/1 20:22	2.328365	1.323309	2.487762	0.682526	1.986893	-0.59132	0.06
18	2015/11/1 20:22	2.379997	1.280099	2.503791	0.88451	0.896404	-0.59132	0.030
19	2015/11/1 20:22	2.582835	1.200328	2.503791	0.812003	-0.0459	-0.5855	0.20
20	2015/11/1 20:22	2.184534	1.26348	2.517805	0.938027	1.627196	-0.59132	-0.00
21	2015/11/1 20:22	2.354181	1.223595	2.505783	1.129652	1.528406	-0.59132	-0.00
22	2015/11/1 20:22	2.324677	1.300042	2.513798	1.164179	1.011659	-0.59132	-0.00
23	2015/11/1 20:22	2.549643	1.293394	2.507799	0.998449	0.284666	-0.5855	0.170
24	2015/11/1 20:22	2.586523	1.280099	2.507799	0.931121	0.139015	-0.59132	-0.00
25	2015/11/1 20:23	2.306237	1.187033	2.507799	0.910405	0.046557	-0.59132	-0.00
26	2015/11/1 20:23	2.402124	1.243537	2.503791	0.894868	0.106085	-0.59132	-0.00
27	2015/11/1 20:23	2.379997	1.300042	2.521812	0.768844	1.172509	-0.5855	0.170

FIGURE 3. Feature extraction from multi-dimensional time series.

3) Evaluation criteria of the model should be suitable for imbalance learning task.

III. METHODOLOGY

The overall learning and prediction pipeline consists of four parts as Figure 2 shown: 1) Data preprocessing and feature extraction; 2) Model selection and composition; 3) Model training and cross-validation; 4) Fault prediction and evaluation.

A. DATA PREPROCESSING AND FEATURE EXTRACTION

Before putting the data into the training model, raw data should be cleaned and feature be extracted. The N-dimensional time series are segmented into time windows by sliding window mechanism. The length of time window is denoted as K and sliding step as L. Thus, each time window is a vector $\boldsymbol{E} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_K}\}$ and $\boldsymbol{x_t}^{\mathrm{T}} = \{x_1^{(t)}, x_2^{(t)}, \ldots, x_N^{(t)}\}$ $(1 \leq t \leq K)$. The feature at time t denotes as $\boldsymbol{f_t} = \boldsymbol{x_t}^{\mathrm{T}}$ and the feature of the time window $\boldsymbol{E}$ represents as $\boldsymbol{f_E} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \ldots, \boldsymbol{f_K}\}$, the detailed description about feature extraction is shown in Figure 3.

B. EASY-SMT ALGORITHM

Undersampling is a popular strategy to deal with the class-imbalance problems, which uses a subset of the majority class to improve performance. However, it also leads to loss of some useful features for majority class. Liu *et al.* [16] propose EasyEnsemble algorithm to fully exploit features from majority class by sampling several subsets from the majority class, training a learner using each of them, and combining the outputs of those learners. Thus, EasyEnsemble overcomes the deficiency of undersampling method. While at the other hand, it is also necessary to generate samples for minority class artificially to make the training data as balanced as possible. SMOTE [14] is an efficient method for class-imbalance problem by adding new synthetic minority class examples according to original data's distributions, which is an effective method to balance the samples before training. Therefore, we propose the Easy-SMT ensemble algorithm based on EasyEnsemble and SMOTE to deal with the class-imbalance problems in industrial system.

Given the minority training set $\mathcal{P}$ and the majority training set $\mathcal{N}$, we use undersampling method randomly sample several subsets $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_{T_1}$ from $\mathcal{N}$. For each subset $\mathcal{N}_i$ $(1 < i < T_1)$, SMOTE algorithm is used to add several new synthetic $\mathcal{P}'_1, \mathcal{P}'_2, \ldots, \mathcal{P}'_{T_1}$ from $\mathcal{P}$, $|\mathcal{N}_i| = T_2\,|\mathcal{P}|$, and $\left|\mathcal{P} + \mathcal{P}'\right| = |\mathcal{N}_i|$. So the number of $\mathcal{P}'_i$ is $T_2 - 1$. Then, a classifier H_i is trained using $\mathcal{N}_i$ and $\mathcal{P}''_i$, $\mathcal{P}''_i = \mathcal{P} + \mathcal{P}'_i$. All generated classifiers are combined for the final decision. Ensemble classifier is used to train the classifier H_i. The pseudocode for Easy-SMT ensemble algorithm is shown in Algorithm 1.

Algorithm 1 The Easy-SMT Algorithm

Input: A set of minority class examples $\mathcal{P}$, a set of majority class examples $\mathcal{N}$, $|\mathcal{P}| < |\mathcal{N}|$, the number of subsets T_1 to sample from $\mathcal{N}$, T_2 is the ratio of $\mathcal{N}_i$ to $\mathcal{P}$, and s_i the number of iterations to train an ensemble H_i

1: $i \Leftarrow 0$
2: **repeat**
3: $\quad i \Leftarrow i + 1$
4: $\quad$ Randomly sample a subset $\mathcal{N}_i$ from $\mathcal{N}$, $|\mathcal{N}_i| = T_2\,|\mathcal{P}|$
5: $\quad \mathcal{N}' = \textbf{\textit{SMOTE}}\,(\mathcal{P}, 100(T_2 - 1), k)$
6: $\quad \mathcal{P}'' = \mathcal{P} + \mathcal{P}'$
7: $\quad$ Learn H_i using $\mathcal{P}''$ and $\mathcal{N}_i$. H_i is an ensemble with s_i weak classifiers $h_{i,j}$ and corresponding weights $\alpha_{i,j}$. The ensembleąŕs threshold is θ_i, i.e.,

$$H_i(x) = sgn(\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \theta_i)$$

8: **until** $i = T_1$

Output: An ensemble

$$H(x) = sgn(\sum_{i=1}^{T_1} \sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \sum_{i=1}^{T_1} \theta_i)$$

Easy-SMT algorithm generate T_1 numbers of balanced subproblems. Classifier H_i for each subproblem is trained, and each classifier H_i is an ensemble with s_i weak classifiers $h_{i,j}$, and corresponding weights is $\alpha_{i,j}$. H_i is simply a logistic regression model for classification. Finally, instead of counting votes from the H_i, we collect all the $h_{i,j}$ and form an ensemble classifier from them. In our proposed algorithm, we add the SMOTE after the step 4 of EasyEnsemble algorithm, which increases the number of minority class samples effectively and solves the problem of imbalanced classification. It can not only ensure that each majority class can be learned, but also generate new samples of minority class so that the features of the minority class samples are learned. To improve the generalization of the model, a K-fold cross validation is used, and the best classifier is chosen as the final model. Furthermore, according to the design criteria, the basic classifier H_i for each sub-training task should be suitable for learning features from class-imbalance data.

IV. EXPERIMENTS AND EVALUATIONS

A. DATA DESCRIPTION

To validate the performance of classifying the fault from normal data with proposed model, two different datasets are used. One is wind turbine data which could be modeled as a binary classification problem, while the other is a plant's operational data which could be modeled as a multi-class classification problem.

1) DATASET #1: WIND TURBINE FREEZING FAULT

The data[1] is provided by a wind turbine manufacturer for PHM competition[2] held by Chinese government (Ministry of Industry and Information Technology, MIIT), which is generated from SCADA of a wind electricity generation system. The data contains 28 dimensional continuous time series, including working conditions, environment parameters and state parameters.

Two wind turbine's data (#15 and #21) are available and labeled with normal and freezing durations (start_time, end_time). It is noted that the unlabeled data in training dataset are ineffective data which are not used for training in our experiment. Each wind turbine contains three files:

1) *_data.csv: original data from SCADA, including time stamp, 26 sensors&actuator, and group information;
2) *_normalinfo.csv: the labeled normal durations, including start_time and end_time;
3) *_failureinfo.csv: the labeled freezing durations, including start_time and end_time;

2) DATASET #2: INDUSTRIAL PLANT FAULT (PHM15 CHALLENGE)

The dataset is provided by PHM 2015 Challenge,[3] which records the actual working conditions of several industrial

[1] https://github.com/minelabwot/Imbalance_Learning
[2] http://www.industrial-bigdata.com/
[3] https://www.phmsociety.org/events/conference/phm/15/datachallenge

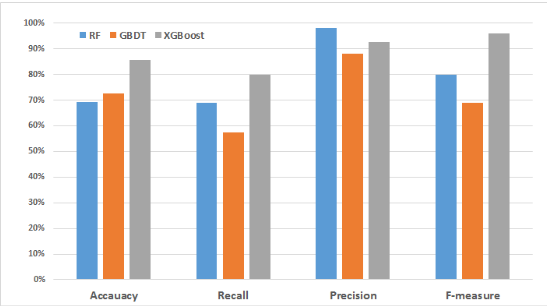
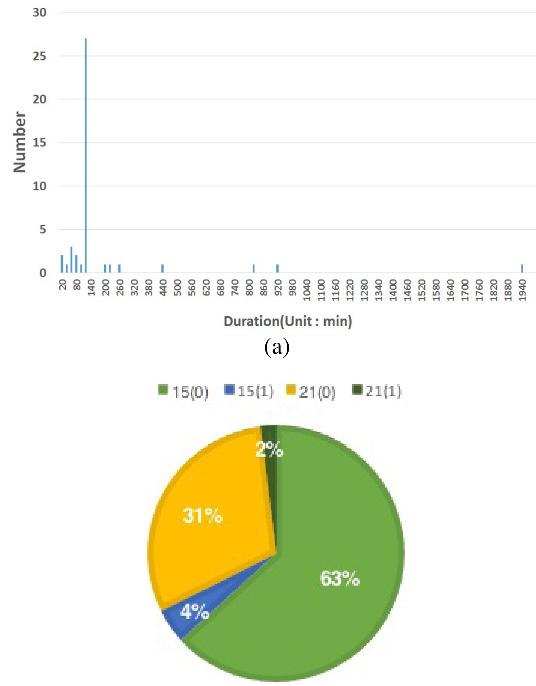


FIGURE 4. Accuracy, Recall, Precision and F-measure of RF, GBDT, XGBoost algorithms.



(a)

(b)

FIGURE 5. Initial statistics on original sensor data. (a) Freezing failure durations. (b) Imbalance class ratio.

plants, including six kinds of faults as well as normal events. The datasets consist of following three parts:

1) time series of sensor measurements and control reference signals for each of a number of control components of the plant (e.g. 6 components);
2) time series data representing additional measurements of a fixed number of plant zones over the same period of time (e.g. 3 zones), where a zone may cover one or more plant components;;
3) plant fault events, each characterized by a start time, an end time, and a failure code.

Only faults of type 1-5 are of interest, while code 6 represents all other faults not in focus. The frequency of measurements is approximately one sample every 15 minutes, and the time series data spans a period of approximately three to four years. The goal is to predict the beginning time and end time of failure events of types 1-5. The dataset can be downloaded from NASAAmes Prognostics Data Repository [(J. Rosca, 2015)].

B. DATA PREPARATION AND EXPERIMENTAL SETTINGS

In this section, we set experiment to evaluate the classifiers. Two sets of experiments are organized based on dataset #1 and #2 respectively.

Task 1 (Wind Turbine's Freezing Fault Forecast): We select data from the 9000th to 149999th timestamp of 21 wind turbine for testing, while the rest are used as training set. RF (Random Forest), GBDT (Gradient Boosting Decision Tree) and XGBoost (eXtreme Gradient Boosting) [29] are used for baseline classifiers. Accuracy, recall, precision and F-measure of three algorithms are show in Figure 4. XGBoost shows competitive performance in recall and accuracy, that is, XGBoost can identify more fault events than other algorithms, meanwhile, F-measure reflect the XGBoost algorithm distinguishes fault and normal events well. Consequently, XGBoost is chosen as baseline classifier to built our model.

According to the initial statistics as figure 5 shown, the shortest failure event happen in 12 minutes, which gives a reference to determine the length of the time window. The shortest failure event lasts for 106 time points, in order to cover the features of all the freezing features, the maximum value of K is 106. We try to use different K and L for the feature extraction, including $K = 106, K = 80$, $K = 50, L = 20, L = 10, L = 5$ and so on. The features of failure and normal event can not be learned well with a small K, and the training time will be longer with a smaller L as well. Consequently, as a compromise, the parameter configuration with $K = 106$, and $L = 20$ is determined. Then, we perform a 5-fold cross validation based on XGBoost algorithm to verify the generalization performance of our model. Each fold of training set is used as a validation dataset, the whole cross-validation process is repeated for five times, and final values from this method are the best of these five cross-validation runs.

We compared the performance of 5 methods in validation sets and testing sets as follows.

1) **XGBoost** (abbreviated as XGB): It uses the entire data set ($\mathcal{P}$ and $\mathcal{N}$) to train an ensemble classifier. The number of iterations is 100.
2) **Undersampling+Oversampling+XGBoost** (abbreviated as UO-XGB): A new minority training set $\mathcal{P}'$ is sampled randomly from the original minority class, a subset $\mathcal{N}'$ is sampled randomly from the original majority class. The ratio between $\mathcal{P}'$ and $\mathcal{N}'$ is $1 : 1.4$. Then, XGBoost is used to train a classifier using $\mathcal{P}'$ and $\mathcal{N}'$. The number of iteration is 100.
3) **SMOTE+XGBoost** (abbreviated as SMT): In our experiments, we first generate $\mathcal{P}'$ using SMOTE, a set of synthetic minority class examples with $\left|\mathcal{P}'\right| = 14\ |\mathcal{P}|$. Then, XGBoost is used to train a classifier using $\mathcal{P}+\mathcal{P}'$ and $\mathcal{N}'$. the number of iteration is 100.

TABLE 1. The ratio of faults and normal events in plant #1.

Event Type	PF1	PF2	PF3	PF4	PF5	PF6	PN
Ratio	4.83%	3.75%	3.39%	0.06%	0.65%	19.26%	68.06%

TABLE 2. The Precision, Recall, F-measure and AUC of compared methods. XGB, UO-XGB, SMT and Easy are evaluated on test datasets with cross validations; Easy-SMT is evaluated on validation and test datasets with cross validation, as well as on test datasets without cross validation.

Method	XGB	UO-XGB	SMT	Easy	Easy-SMT (validation)	Easy-SMT (no CV)	Easy-SMT (CV)
Recall	34.97%	46.15%	81.12%	88.82%	94.93%	95.10%	**96.50%**
Precision	100.00%	95.31%	88.33%	75.45%	78.49%	86.33%	**86.82%**
F1	0.5182	0.6219	0.8457	0.8159	0.8593	0.9050	**0.9140**
AUC	0.2998	0.4173	0.6658	0.6236	0.7411	0.7458	**0.7564**

4) **EasyEnsemble+XGBoost** (abbreviated as Easy): Number of subsets $T_1 = 15$, The number of iteration is 100.
5) **EasyEnsemble+SMOTE+XGBoost** (abbreviated as Easy-SMT): Number of subsets $T_1 = 5$, for each subsets $\mathcal{N}_i$, we generate $\mathcal{P}'$ using SMOTE, a set of synthetic minority class examples with $|\mathcal{P}'| = 2\ |\mathcal{P}|$. Then, XGBoost is used to train a classifier using $\mathcal{P} + \mathcal{P}'$ and $\mathcal{N}_i$. The number of iteration is 100.

Task 2 (Plant's Fault Prediction): To better evaluate the results, plant #1's training datasets with labels are used to validate the performance of our model on multi-class imbalance classification. The original labeled data are randomly divided into training dataset and test dataset with 9:1 proportion. Moreover, the original data are segmented into time windows with 48 dimensions (6 machines with 8 dimensions for each). An initial statistics of the imbalance ratio are given in Table 1, and the results show that each type of fault is at a different high imbalance ratio. XGBoost is also chosen as baseline multi-class classifier to evaluate the model.

We compare several baseline methods as follows.

1) **XGBoost** (abbreviated as XGB): It uses the entire data set ($\mathcal{P}$ and $\mathcal{N}$) to train an ensemble classifier. The number of iterations is 5000.
2) **SMOTE+XGBoost** (abbreviated as SMT): In our experiments, we first generate $\mathcal{P}'$ using SMOTE, a set of synthetic minority class examples with $|\mathcal{P}'| = |\mathcal{N}|$-$|\mathcal{P}|$. Then, XGBoost is used to train a classifier using $\mathcal{P} + \mathcal{P}'$ and $\mathcal{N}'$. the number of iteration is 5000.
3) **SMOTE+Undersampling+XGBoost** (abbreviated as SMT-U): A subset $\mathcal{N}'$ is sampled (without replacement) from $\mathcal{N}$. Then, we generate $\mathcal{P}'$ using SMOTE, a set of synthetic minority class examples with $|\mathcal{P}'| = |\mathcal{N}'|$-$|\mathcal{P}|$. Then, XGBoost is used to train a classifier using $\mathcal{P} + \mathcal{P}'$ and $\mathcal{N}'$, iteration is 5000.
4) **EasyEnsemble+XGBoost** (abbreviated as Easy): Number of subsets $T_1 = 1000$, The number of iteration is 500.
5) **EasyEnsemble+SMOTE+XGBoost** (abbreviated as Easy-SMT): Number of subsets $T_1 = 4$, for each subsets $\mathcal{N}_i$, we generate $\mathcal{P}'$ using SMOTE, a set of synthetic minority class examples with $|\mathcal{P}'| = |\mathcal{N}_i|$-$|\mathcal{P}|$. Then, XGBoost is used to train a classifier using $\mathcal{P} + \mathcal{P}'$ and $\mathcal{N}_i$. The number of iteration is 5000.

C. RESULTS AND EVALUATION

Firstly. we evaluate the model on the test data of dataset #1 with cross validations. The overall results show that our proposed method outperforms other four methods on validation sets. As Table 2 shown, Easy-SMT has not only the highest recall and F-measure, but also a better precision. Meanwhile, our proposed method has a better classification effect on imbalanced data according to the AUC value.

Specifically, XGB is not designed for the imbalanced data, and obviously, this method has a good classification ability for the majority class samples, but a lot of minority class samples are not recognized because the features of the minority class samples are too small to learn. UO-XGB method use under-sampling to the majority class samples and oversampling to the minority class samples [30], which solve the problem of imbalance to some extent. However, oversampling is easy to cause overfitting, and under-sampling may delete some valuable samples. SMT method is an efficient method for class-imbalance problem by adding new synthetic minority class examples, but generally the data is with a large imbalance level. It can alleviate the overfitting problem of oversampling by generating large minority class samples. However, SMT ignores adjacent instance of other classes when the synthesis of new samples, which results in a large number of samples overlap in different classes. Easy method effectively solves the imbalance problem by exploring the majority class examples ignored by under-sampling. But it is difficulty to learn the characteristics of minority due to the scarcity of minority class samples. Easy-SMT method add a small amount of artificially generated minority class samples to each classifier in Easy method, which increases the recall. Recall of Easy-SMT shows that it can predict more fault events than other methods, meanwhile, F-measure value indicates that Easy-SMT can distinguish the normal events and fault events precisely. As Figure 6 shown, we can see that ROC of Easy-SMT can completely wrap the ROC curve of

TABLE 3. The Precision, Recall, F-measure and AUC values of SMT, Easy and Easy-SMT compared with corresponding results on validation datasets.

Method	SMT (no CV)	SMT (CV)	Easy (no CV)	Easy (CV)	Easy-SMT (no CV)	Easy-SMT (CV)
Recall	-0.14%	**+0.56%**	-0.25%	-1.64%	+0.17%	**+1.57%**
Precision	+12.68%	**+14.67%**	+0.81%	-3.44%	+7.843%	**+8.33%**
F1	+0.0631	**+0.0761**	+0.0033	-0.0268	+0.0457	**+0.0547**
AUC	-0.0819	**+0.0291**	-0.0807	-0.0951	+0.0047	**+0.0153**

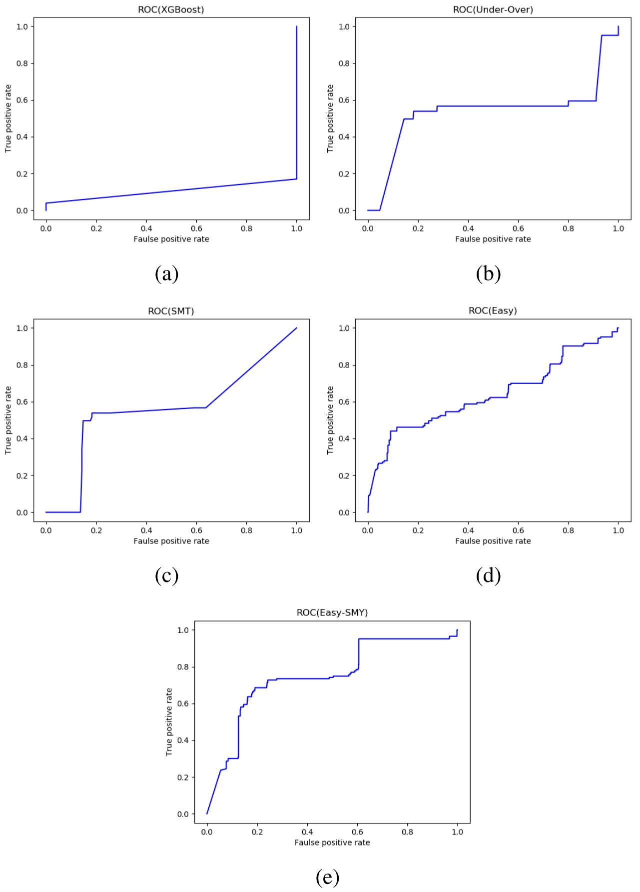


FIGURE 6. The ROC curve comparisons between Easy-SMT and other baseline methods. (a) The ROC curve of XGB. (b) The ROC curve of UO-XGB. (c) The ROC curve of SMT. (d) The ROC curve of Easy. (e) The ROC curve of Easy-SMT.

other methods, and the performance of this method is the best among the five methods. Meanwhile, Easy-SMT has a highest AUC value, showing it is a good method for imbalanced data in classification.

Secondly, we compare the generalization performance of different models by evaluating on the validation data and test data of dataset #1. The values in Table 3 indicate the relative improved or deteriorated performance on test datasets compared with which on validation datasets. The results illustrate the generalization performance of different methods. As Table 3 shown, Easy-SMT performs well on both validation datasets and test datasets even without cross validation. If cross validation is added to the model, the performance is improved more from validation to test datasets than without cross validation. Table 2 shows that SMT and Easy-SMT with cross validations gain relative higher improvements than Easy, and the results illustrate that SMOTE-based oversampling method with cross validation has a positive influence on the generalization of the classifier; while EasyEnsemble without SMOTE has an obvious decreased results with cross validation, which illustrates that EasyEnsemble performs overfitting even with cross validation. Compared with the results of SMT, Easy and Easy-SMT without cross validation, it could be found that SMT and Easy perform not well on recall and AUC, which illustrates that Easy-SMT gains more generalized performance on the imbalanced learning problem. Moreover, Easy-SMT with cross validation improves more than Easy-SMT without cross validation. Consequently, Easy-SMT with cross validation is the best imbalanced classifier on the failure prediction task.

TABLE 4. The score of predicting the freezing failure of wind turbine in different methods.

Method	FN	FP	Score
XGB	419	2034	44.16
UO-XGB	13567	679	79.81
SMT	10145	646	81.10
Easy	14438	372	88.13
Easy-SMT	**10126**	**150**	**94.71**

According to the evaluation standard of wind turbine challenge, the score metric for fault prediction is defined as:

$$Score = (1 - \alpha \times \frac{FN}{N_{normal}} - \beta \times \frac{FP}{N_{fault}}) \times 100\%$$

α and β are weight coefficient, given by the number of positive and negative samples, $\alpha + \beta = 1$. FN (false negative) is the number of normal events which are labelled falsely. FP (fault positive) is the number of fault events which are labelled falsely. According to the data set, $N_{normal} = 52564$, $N_{fault} = 3423$.

The scores in Table 4 reflect that the performance of predicting fault events in the industrial filed, the results of all methods have been adjusted in noise elimination and fault merging. We eliminate the failure event which only last a window time, and we also merge the adjacent failure events into a failure event when they have a short interval. Easy-SMT

TABLE 5. The Precision, Recall and F-measure of XGB, SMT, SMT-U, Easy and Easy-SMT compared with corresponding results on test data of dataset #2.

Method	XGB	SMT	SMT-U	Easy	Easy-SMT
Recall	59.81%	80.7%	83.26%	**86.88%**	**84.38%**
Precision	53.57%	84.03%	74.41%	67.55%	**84.19%**
F1	56.02%	82.11%	77.67%	72.99%	**84.0%**

TABLE 6. The Precision, Recall and F-measure of faulty and normal classes of dataset #2 based on Easy-SMT.

Method	PF1	PF2	PF3	PF4	PF5	PF6	PN
Recall	83.64%	92.25%	92.41%	85.71%	61.29%	92.83%	96.35%
Precision	79.86%	75.72%	83.25%	100%	52.78%	88.01%	95.89%
F1	81.71%	83.17%	87.59%	92.31%	56.72%	90.36%	96.12%

has a highest score, which can identify more fault events and normal events correctly than other methods.

Finally, we compare the performance of different models on test data of dataset #2 to illustrate the effectiveness and efficiency on multi-faults imbalance learning problem. As Table 5 shown, Easy-SMT also performs well on imbalanced multi-faults recognition, which illustrates that the integrated solution with data augmentation, under-sampling and ensemble learning policies could learn the minor and major classes' features better than other solutions with partial policies. Compared SMT, SMT-U and Easy-SMT with XGB and Easy, the results illustrate that SMOTE as data augmentation policies enhances the baseline classifier's performance significantly. While, compared Easy-SMT and SMT-U with SMT, the results illustrates that random under-sampling and EasyEnsemble as a kind of under-sampling policy enhance the performance. And moreover, EasyEnsemble works better than random under-sampling since EasyEnsemble has fully utilized the data of majority class by dividing the imbalance learning problem into several balance learning subproblems. Compared Easy with Easy-SMT, the results show that the recall and F1 have been significantly improved by Easy-SMT, while the precision is slightly decreased. It could be explained that Easy-SMT could better recognize the minority faulty classes since precision and F1 are high though recall is a little bit lower. The results also illustrate that SMOTE as a data augmentation policy benefits in improving representations of the minority classes' features, and EasyEnsemble as a kind of under-sampling policy does not abandon useful features of majority classes.

In addition, the classification performances on each type of faulty and normal classes are shown In Table 6. The results indicate that the performance on different faulty classes with different imbalance ratios are similar except for PF4, which illustrate that Easy-SMT performs quite well on relative low imbalance ratio fault class (PF6) and also achieves a good performance on medium imbalance ratio fault classes (PF1-PF3). However, the performance fluctuates a lot on extremely imbalanced fault classes, where PF4 achieves relatively good results and PF5 achieves unsatisfactory results. The phenomenon illustrates that there still exists improvement spaces for Easy-SMT on extremely imbalanced fault diagnosis.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a framework for failure prognostics of industrial systems, which can transform the prediction issue into a classification problem and predict the failure by real-time data without any prior knowledge. In our model, we propose Easy-SMT ensemble algorithm to overcome the challenge about imbalance learning. Furthermore, this model can be used to predict the freezing failure of wind turbine and machine's faults at plant levels based on two representative PHM competitions. The experiments show that the Easy-SMT achieve better performance than other baseline models on binary and multi-faults classifications. To improve the effectiveness and efficiency of the prognostic model for industrial area, a feature selection mechanism with data augmentation policies could be further studied. In addition, a hybrid new method based on data-level and algorithm-level model could be studied in the future.

REFERENCES

[1] C. Hu, B. D. Youn, P. Wang, and J. T. Yoon, "Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life," *Rel. Eng. Syst. Safety*, vol. 103, pp. 120–135, Jul. 2012.

[2] H. Kim, T. Hwang, J. Park, H. Oh, and B. D. Youn, "Risk prediction of engineering assets: An ensemble of part lifespan calculation and usage classification methods," *Int. J. Prognostics Health Manage.*, vol. 5, no. 2, 2014.

[3] N. V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 1–6, Jun. 2004.

[4] R. Isermann, "Model-based fault-detection and diagnosis—Status and applications," *Annu. Rev. Control*, vol. 29, no. 1, pp. 71–85, 2005.

[5] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, no. 4, pp. 298–306, 2014.

[6] Y. Wu, B. Jiang, N. Lu, and Y. Zhou, "Bayesian network based fault prognosis via Bond graph modeling of high-speed railway traction device," *Math. Problems Eng.*, vol. 2015, May 2015, Art. no. 321872.

[7] J. Korbicz, J. M. Kościelny, Z. Kowalczuk, and W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Springer, 2012.

[8] 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," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[9] L. Wang and J. Yu, "Fault feature selection based on modified binary PSO with mutation and its application in chemical process fault diagnosis," in *Advances in Natural Computation*. Springer, 2005, pp. 832–840.

[10] S. Lee, W. Park, and S. Jung, "Fault detection of aircraft system with random forest algorithm and similarity measure," *Sci. World J.*, vol. 2014, Jun. 2014, Art. no. 727359.

[11] C. Drummond and R. C. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. Workshop Notes ICML Workshop Learn. Imbalanced Data Sets*, Washington, DC, USA, 2003, pp. 1–8.

[12] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 7–19, Jun. 2004.

[13] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, Jan. 2006.

[14] L. Demidova and I. Klyueva, "SVM classification: Optimization with the SMOTE algorithm for the class imbalance problem," in *Proc. 6th Medit. Conf. Embedded Comput. (MECO)*, Bar, Montenegro, Jul. 2017, pp. 1–4.

[15] P. K. Chan and S. J. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection," in *Proc. 4th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 1998, pp. 164–168.

[16] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[17] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[19] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell.*, 2000, pp. 111–117.

[20] I. Mani and I. Zhang, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. ICML Workshop Learn. Imbalanced Datasets*, 2003, pp. 42–48

[21] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Adv. Intell. Comput.*, 2005, pp. 878–887.

[22] J. Xie and Z. Qiu, "The effect of imbalanced data sets on LDA: A theoretical and empirical analysis," *Pattern Recognit*, vol. 40, no. 2, pp. 557–562, 2007.

[23] A. Estabrooks, T. H. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, 2004.

[24] R. Barandela *et al.*, "The imbalanced training sample problem: Under or over sampling?" in *Proc. Joint IAPR Int. Workshops Struct., Syntactic, Statist. Pattern Recognit. (SSPR SPR)*, Lisbon, Portugal, Aug. 2004, p. 806.

[25] C. Seiffert, T. M. Khoshgoftaar, and J. Van Hulse, "Improving software-quality predictions with data sampling and boosting," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 6, pp. 1283–1294, Nov. 2009.

[26] N. V. Chawla *et al.*, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases—PKDD* (Lecture Notes in Computer Science), vol. 2838, 2003, pp. 107–119.

[27] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[28] R. E. Schapire, "A brief introduction to boosting," in *Proc. 16th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, 1999, pp. 1401–1406.

[29] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2016, pp. 785–794.

[30] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory under-sampling for class-imbalance learning," in *Proc. 6th IEEE Int. Conf. Data Mining*, Hong Kong, Dec. 2006, pp. 965–969.

ZHENYU WU received the B.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2008 and 2013, respectively. He is currently an Assistant Professor with the School of Information and Communication Engineering, BUPT. His research interests include Internet of Things, machine learning, prognostics and health management technology, and industrial big data analysis.

WENFANG LIN received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2016, where she is currently pursuing the M.S. degree with the School of Information and Communication Engineering. Her research interests include Internet of Things, machine learning, prognostics and health management technology, and industrial big data.

YANG JI received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2002. He is currently a Full Professor with the School of Information and Communication Engineering, BUPT, where he is also the Director of the Mobile Life and New Media Laboratory. His research interests include Internet of Things, machine learning, smart manufacturing, and smart education.