Article citation info:
CHEN X, ZHANG Z, ZHANG Z. Real-time equipment condition assessment for a class-imbalanced dataset based on heterogeneous ensemble learning. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2019; 21 (1): 68–80, http://dx.doi.org/10.17531/ein.2019.1.9.

Xiaohui CHEN
Zhiyao ZHANG
Ze ZHANG

REAL-TIME EQUIPMENT CONDITION ASSESSMENT FOR A CLASS-IMBALANCED DATASET BASED ON HETEROGENEOUS ENSEMBLE LEARNING

OCENA STANU SPRZĘTU W CZASIE RZECZYWISTYM DLA ZBIORÓW DANYCH O NIEZRÓWNOWAŻONYM ROZKŁADZIE W KLASACH. METODA OPARTA NA UCZENIU ZESPOŁOWYM

This study proposes an ensemble learning model for the purpose of performing a real-time equipment condition assessment. This model makes it possible to plan desired preventive maintenance activities before an unexpected failure takes place. This study focuses on the class-imbalanced problem in equipment condition assessment research. In reality, equipment will experience multiple conditions(states), most of the time remaining in the normal condition and relatively rarely being in the critical condition, which means that, from the perspective of data modelling, the distribution of samples is highly imbalanced among different classes(conditions). The majority of samples belong to the normal condition, while the minority belong to the critical condition, which poses a great challenge to the classification performance. To address this problem, a genetic algorithm-based ensemble learning model is presented. Furthermore, a self-updating learning strategy is presented for online monitoring, contributing to adaptability and reliability enhancement along with time. Many previous studies have attempted feature extraction and to set thresholds for equipment health indicators. This study has an advantage of omitting these steps, as it can directly assess the equipment condition through the proposed ensemble learning model. Numerical experiments, including two types of comparison studies, have been conducted. The results show the greater effectiveness of our proposed model over that of previous research in terms of the stability and accuracy of its classification performance.

***Keywords**: condition assessment; heterogeneous ensemble learning; genetic algorithm; class-imbalanced.*

W pracy przedstawiono model uczenia maszynowego opartego na zespołach niejednorodnych klasyfikatorów (ensemble learning), który pozwala przeprowadzać ocenę stanu sprzętu w czasie rzeczywistym. Model ten umożliwia zaplanowanie niezbędnych czynności konserwacji profilaktycznej przed wystąpieniem niespodziewanego uszkodzenia. Tematem pracy jest zagadnienie niezrównoważonego rozkładu w klasach poruszane w badaniach dotyczących oceny stanu sprzętu. W warunkach rzeczywistych, sprzęt chrakteryzuje wiele różnych stanów, przy czym przez większość czasu pozostaje on w stanie normalnym, a relatywnie rzadko znajduje się w stanie krytycznym, co oznacza, że z punktu widzenia modelowania danych, rozkład prób w poszczególnych klasach (stanach) jest wysoce niezrównoważony. Większość prób należy do stanu normalnego, a mniejszość do stanu krytycznego, co stanowi duże wyzwanie jeśli chodzi o wydajność klasyfikacji. W celu rozwiązania tego problemu, przedstawiono model uczenia zespołowego oparty na algorytmie genetycznym. Ponadto zaprezentowano samoaktualizującą się strategię uczenia wykorzystywaną do monitorowania online, która wraz z upływem czasu zwiększa adaptacyjność i niezawodność modelu . W wielu poprzednich badaniach podejmowano próby ekstrakcji cech oraz ustalania progów dla wskaźników stanu sprzętu. Zaletą przedstawionej metody jest to, że pozwala ona pominąć te etapy i bezpośrednio oceniać stan sprzętu za pomocą proponowanego modelu uczenia zespołowego. Przeprowadzono eksperymenty numeryczne, w tym dwa rodzaje badań porównawczych. Wyniki pokazują większą skuteczność proponowanego modelu w stosunku do poprzednich badań pod względem stabilności i trafności klasyfikacji.

***Słowa kluczowe**: ocena stanu; uczenie zespołowe; algorytm genetyczny; niezrównoważony rozkład w klasach.*

1. Introduction

Prognostics and health management (PHM) is beneficial for daily operation and maintenance [21]. PHM covers condition assessment, fault diagnosis, remaining useful life (RUL) prediction, maintenance decision and other considerations. Condition assessment is a fundamental activity to identify the current condition/state of equipment. Equipment ages and degrades with time. When equipment degrades to a certain degree or pass a certain threshold, it cannot operate well, which results in unqualified products, system breakdown or even casualties. Since equipment's reliability and stability are meaningful for ensuring the safe and continuous operation, effective equipment condition assessment is an important prerequisite. Moreover, condition assessment could provide a convenience for several subsequent activities, such as condition-based maintenance, planning and scheduling [35, 39]. Condition assessment could be performed through either removing a component from operation (off-line) or doing on-line monitoring. Considering the cost and complexity of installation and removal, real-time condition assessment with continuous on-line monitoring is more economical and feasible.

Overall, there are three major categories, (i) criteria-based approaches [36, 41], (ii) statistical-based approaches [3, 13, 18, 20], and (iii) data-driven approaches. In criteria-based approaches, health indicators (i.e. main functions, reliability degree, working time, and deterioration degree) are proposed [37] to evaluate equipment condition. But these approaches have difficulties on indicators quantifying and

indicators causal interrelationship quantification. The fundamental principle behind the statistical-based approaches is the formulation of theoretical mathematical models for interpreting equipment deterioration. Although they describe well the deterioration of equipment over time, they have a limitation on stability and sensitivity when facing unexpected impacts (power failure, shocks, instantaneous overload or no-load, etc.) [11, 12]. As for data-driven methods [4, 30, 33], it tries to learn the data intrinsic properties and underlying relations through monitoring data in order to assess the equipment condition. Compared the former approaches, data-driven approaches are less complex and more applicable. We do not need to construct complex hierarchical structure, or extract feature, because most data-driven techniques always have automatically learning ability. However, these approaches rely heavily on properties of the training data. So, for a real-life condition assessment problem, new challenge comes out, as its database is highly class-imbalanced.

Our study focuses on this class-imbalanced problem for condition assessment. A dataset exhibits the class-imbalanced problem when the data samples of one class (majority class) outnumber the data samples of the other classes (minority classes). The latter usually denotes a topic of interest in a data classification problem. Actually, in real-world data-oriented applications, the class-imbalanced dataset is prevalent, e.g. in fraud/intrusion detection and medical diagnosis/monitoring. With class-imbalanced dataset, the standard classifier, such as Decision Tree (DC), Random Forest and Support Vector Machine (SVM) [4, 30], performs badly, because it has a tendency to bias towards the over represented class. Dominated by the majority class, the classifier lacks the generalization ability of classification rules for classes with minor samples, because the classifier may consider the samples in minority classes to be noise.

The research on classification with the class-imbalanced dataset has gotten much attention, since Japlowicz performed the experiments on a dataset with characteristics of various size, complexity and class-imbalanced in 2000 [17]. In his study, he discussed the assumption that the training set is well balanced in the majority of concept-learning systems, and he verified that class-imbalanced hinder the performance of standard classifiers. Further studies pay more attention to the classification performance for the class-imbalanced classification problems. Standard classification algorithms on class-imbalanced dataset suffer from a significant loss of performance, providing suboptimal classification results [5, 22]. The results often bias the majority class, leading to a higher classification error for minority classes [28]. Therefore, new rules have been studied to better generalize the minority class to avoid treating them as noise. Increasingly, research has focused on trying to excavate and magnify the data intrinsic properties of the minority class.

Generally, the approaches for tackling classification problems for the class-imbalanced dataset are typically categorized as data-level, algorithmic level and the combination of these two levels [14]. Re-sampling is a common approach at the data-level, which aims at re-balancing highly imbalanced class distributions. Under-sampling strategies [1, 23] decrease samples in the majority class, and over-sampling strategies [9, 24] increase samples in the minority class (classes). However, both strategies show drawbacks. Over-sampling may increase the risk of over-fitting and worsen computational burden of the learning algorithm, and under-sampling may lose some useful information. As a result, the re-sampled dataset can be completely different from the original one, because the original class distribution is altered. Mathew et al. [27] emphasised that fault stage diagnosis in industrial machines are often imbalanced and consist of multiple categories or classes. In their study, a weighted kernel-based oversampling algorithm has been put forward to generate minority samples in order to balance the class distribution in an SVM classifier. With this algorithm, a higher overall accuracy have been obtained.

Cost-sensitive learning incorporates approaches at the data-level, at the algorithmic level, or at both levels combined, considering higher costs for the misclassification of examples of the positive class with respect to the negative class and therefore trying to minimize higher cost errors [38]. Cost-sensitive learning allocates unequal costs for different classes in the learning process based on the assumption that misclassification costs are already known [10]. However, there is difficulty in determining the costs because the prior cost information is not available. If positive instances are sparse, cost-sensitive learning may not have the ability to construct appropriate decision boundaries. Another limitation is that cost-sensitive learning may work well when facing a not-highly imbalance dataset, but fail when dealing with a highly imbalance dataset [16]. In [40], instability events were considered to be the reason for class-imbalanced dataset in power system short-term voltage stability assessment problem. combined the forecasting-based nonlinear synthetic minority oversampling technique and cost-sensitive learning, respectively dealing with class-imbalanced dataset in data-preprocessing and algorithm aspects and achieved desirable performance.

Regard to algorithm level, ensemble learning is one of the best performing approaches. It is oriented towards the adaptation of base learning methods to be more attuned to class imbalance issues. The basic idea behind ensemble learning is to use more than one classifier to improve the overall accuracy. Ensemble learning has been widely used in many fields e.g. finance, manufacturing, bioinformatics, geography, medicine, information security and recommender systems to improve the classifier performance of single models [25]. Othman et al. [29] proposed an ensemble discriminant classifier with four base learners to power transformers condition assessment problem, and verified that the proposed ensemble model outperformed the SVM classifier. Boosting, bagging and stacking are the three main strategies [16]. Among them, boosting is the most commonly used. It highlights the misclassified samples at each iteration and reduces the bias from data by combining classification results from several weak learners. In each round, the weights for samples in minority classes are increased.

The ensemble learning model optimizes the overall classification performance depending on two factors, individual success of the individual learners and diversity. One way of providing diversity is to use different types of individual learners. Another way is using different training datasets. In this way, the same type of individual learner is adopted. According to whether different individual learners are used, ensemble learning can be divided into two groups: heterogeneous ensemble and homogenous ensemble. Homogenous ensemble is a convenient and prevalent approach, as choosing a certain amount of the same type of individual learner (homogeneous individual learner). Adaboost is the classic demon for boosting model with homogeneous individual learner [8, 34]. Lee et al. [19] conducted a SVM-based Adaboost model to address the class-imbalanced classification. Different factor scores were computed by categorizing samples based on the SVM margin. Another strategy for individual learners is to choose heterogeneous individual learners. Models with heterogeneous individual learners have advantages in learning different characteristics of the training dataset, since they use a diverse set of individual learners. However, in this strategy, previous research always makes the implicit assumption that every heterogeneous individual learner category only has one individual learner. For example, in [15], 20 individual learners coming from five categories were chosen to construct the individual learner base and genetic algorithm was implemented to search for the appropriate individual learners to combine the ensemble learning model. In our study, we doubt this assumption, and we argue that the heterogeneous ensemble learning model constructed by heterogeneous individual learners with a certain amount may have better performance than heterogeneous ones with only one learner in every heterogeneous individual learner category.

This study offers two main contributions. First, this study establishes a genetic algorithm-based ensemble learning model, which can greatly enhance the classification accuracy of the class-imbalanced dataset. Another contribution is that we optimize this model to realize dynamic condition assessment and achieve self-updating ability. This optimized model can help to improve the availability and reliability. Another contribution is that the proposed model, which omits the steps of feature extraction and setting thresholds for equipment health indicators, can directly be used to assesse equipment condition. In practice, the result of equipment condition assessment may help managers to

Table 1. notations

Notation	Description
S	the whole dataset, $S = \{(X,Y)\}^I$
X	the set for key measurements, $X = \{x_1, x_2 \ldots x_M\}^T \in R^{M\times N}$
x_i	a row vector, one data about the key measurements, $x_i = \{x_{i1}, x_{i2} \ldots x_{iN_1}, x'_{i1}, x'_{i2} \ldots x'_{iN_2}\} \in R^N$
x_{ij}	external data among the key measurements
x'_{ij}	internal data among the key measurements
Y	the set for equipment condition $Y = \{y_1, y_2 \ldots y_M\}^T \in R^M$
y_i	the equipment condition, $y_i \in \{0,1,2\}$
N_1	the number of key measurements about external data
N_2	the number of key measurements about internal data
N	the number of key measurements
M	the number of samples
$G(x_i)$	the function depicting the relationship between x_i and y_i
P_i	the precision for class i
R_i	the recall for class i
F_i	the F_measure for class i
δ	the coefficient for the bias of the recall and precision
CM	the matrix for the result of condition assessment
C_{coe}	the coefficient matrix for penalty and award
C_t	the coefficient matrix for the relationship of results in time order
c_e , c_l	the coefficient for early assessment and late assessment
M_{ap}	the award- penalty matrix
p, q	the weights for computing the utility function f_u
m_{ij}	the value for award-penalty function
f_u	the utility function
t_{sp}	the setup time for maintenance
t_{int}	the monitoring interval
t_{com}	the computation time for the classification model
λ	the coefficient to constraint t_{com} , $t_{com} \le \lambda t_{int}$, $\lambda \in (0,1]$
E	the maximum number of each heterogeneous individual learner
L	the sum length of the gene
β_1, β_2	the tolerance and allowance coefficient
t_f	the time of fault point
t_{rul}	the RUL at time t
t_{up}	the time interval for updated the database
K	the times for K-fold validation

make decisions about operation and maintenance of the equipment. For example, it helps managers to decide when to prepare necessary materials and human resources before the occurrence of a failure.

The remainder of this paper is organized as follows. Section 2 formulates the equipment condition assessment problem. Section 3 explicitly describes the proposed model and the way optimize it. Experiments are conducted to verify the performance of the proposed model in Section 4. Finally, concluding remarks and future research suggestions are given in Section 5.

2. Problem formulation

The notation that will be used throughout the paper is summarized in the Table 1.

Equipment condition assessment is an important activity that can visually reflect the current condition of equipment. This activity benefits managers, as it provides information about equipment condition and makes it possible to plan maintenance activities before failure.

In our study, equipment condition is graded into three broad classes, (i) "Healthy", (ii) "Minor defect", and (iii) "Critical defect". The descriptions of these three conditions are provided in Table 2. Additionally, as depicted in this table, three colour codes [2, 7] are utilized to visually indicate the corresponding potential danger level of the three conditions.

Table 2. descriptions of equipment conditions

Class	Descriptions
Healthy	All the critical characteristic quantities are successively decreasing but always stay in a safe region, above the standard limit values.
Minor defect	Some of the critical characteristic quantities are out of bounds, but the comprehensive influence is small. There appear slight defects in the ability to resist risks and adapt to the environment.
Critical defect	Serious deterioration appears, and critical characteristic quantities are out of bounds. The comprehensive influence is large, the equipment cannot normally carry out the regulated functions any longer, and failures can happen at any time.

Condition "Healthy" is the initial condition under which equipment can work well. Under the condition of "Minor defect", managers should pay more attention to the equipment and the maintenance plan should scheduled (which means the spare parts, human resources, maintenance tools and other required resources should be considered and prepared in case of need.) to prevent consequential damages and avoid undesirable consequences. Under the condition of "Critical defect", the maintenance activity is a pressing need because a fault could occur at any time.

Let $S = \{(X,Y)\}^T$ denote the whole dataset, where the measurement set X($X = \{x_1, x_2 \ldots x_M\}^T \in R^{M\times N}$) is a matrix consisting of N key measurements from condition monitoring reflecting the health condition of the equipment and M is the number of samples in this dataset. The value x_i ($x_i = \{x_{i1}, x_{i2} \ldots x_{iN_1}, x'_{i1}, x'_{i2} \ldots x'_{iN_2}\} \in R^N$) is a row vector consisting of two sections: N_1 external data ($\{x_{i1}, x_{i2} \ldots x_{iN_1}\}$) and N_2 internal data ($\{x'_{i1}, x'_{i2} \ldots x'_{iN_2}\}$), in which $N=N_1+N_2$. External data are related to the operation settings of the equipment, while internal data contain internal information such as vibration, temperature increases. The class label set Y ($Y = \{y_1, y_2 \ldots y_M\}^T \in R^M$) is a column vector, where $y_i \in \{0,1,2\}$ denotes the condition of equipment (Eq. 1). In the following parts, class 0, class 1 and class 2 are alternative expressions for condition "Healthy", "Minor defect" and "Critical defect", respectively:

$$y_i = \begin{cases} 0 & \text{if the condition is "Healthy"} \\ 1 & \text{if the condition is "Minor defect"} \\ 2 & \text{if the condition is "Critical defect"} \end{cases} \quad (1)$$

In essence, the condition assessment problem is a classification problem. That means we should identify the current condition/state (class 0, class 1, class 2) of the equipment. So a classification learning model should be conducted to learning the corresponding relationship between X and Y. As a result, when a certain x_i is given, the classification learning model should quickly give the corresponding condition for the equipment, namely, y_i. The relationship described by this model is denoted as Eq. 2:

$$y_i = G(x_i) \quad (2)$$

In reality, equipment is working with desired reliability most of the time. That means, for the equipment condition assessment problem, the majority of samples belong to class 0, while minority of samples belong to class 1 and class 2. In this paper, we call class 0 the majority class, and call class 1 and class 2 the minority classes (class). As the distribution of samples in these conditions is highly skewed, this equipment condition assessment problem is not a simple classification problem, but a specific one with class-imbalanced dataset. In addition, the primary interest is devoted to class 1 and class 2 because they contain relevant information when making production operation and maintenance plans. Therefore, classification performance for the minority classes is key for condition assessment.

3. The genetic algorithm-based ensemble learning model

This section contains three main parts, a description of criteria for classification evaluation, the steps for the genetic algorithm-based ensemble learning model (GAEM) and how to optimise this model.

3.1. Criteria for classification evaluation

As minority class would be dominated by majority class, it is often meaningless to achieve high accuracy when dataset is class-imbalanced, especially when situation where the minority classes are more important and cannot be sacrificed. How to choose suitable criteria to evaluate the classification model's performance is also an important research point. In our study, for multi-class classification, a 3×3 contingency table named confusion matrix is illustrated in Eq. 3, in which cm_{ij} denotes the number of samples whose actual condition is i and the classification result for this sample is j :

$$CM = \begin{bmatrix} cm_{00} & cm_{01} & cm_{02} \\ cm_{10} & cm_{11} & cm_{12} \\ cm_{20} & cm_{21} & cm_{22} \end{bmatrix} \quad (3)$$

The criteria for classification evaluation in the class-imbalanced classification problem are important when comparing the performances of different classification models. Moreover, in regard to the multi-class classification, criteria become more crucial and intractable. As we should give more emphasis to the minority classes than the majority class, the commonly used criteria in binary classification,

accuracy and error, are not adequate. A single performance criterion can be misleading and may fail to evaluate performance on unseen data [29]. Therefore, to obtain a more reliable evaluation, we utilize precision, recall, and F-measure to evaluate the output quality of our classification model.

Precision measures the number of samples that are classified as positive and are actually positive, while recall measures the number of positive samples which are correctly classified as positive [16]. More complicated than the binary classification problem, which only contains positive and negative samples, this multi-class classification problem contains three classes, denoted as 0, 1, and 2. A small difference from binary classification is that we make a fine adjustment about "positive samples" and "negative samples" in this multi-class classification. In terms of the precision for class 0, we consider the samples that belong to class 0 to be "positive samples". The other classes, namely, class 1 and 2, are considered "negative samples". In this way, these three classes will have their own precisions and recalls. P_i and R_i denote the precision and recall for class *i,* which are defined as below. Additionally, we adopt F-measure (also called F-score) [16] to give a balance to the conflict between precision and recall. Here, we use F_i to denote F-measure, in which δ is a coefficient for the bias of the two criteria.

$$P_i = \frac{cm_{ii}}{cm_{0i} + cm_{1i} + cm_{2i}} \tag{4}$$

$$R_i = \frac{cm_{ii}}{cm_{i0} + cm_{i1} + cm_{i2}} \tag{5}$$

$$F_i = \frac{\left(\delta^2 + 1\right) P_i \times R_i}{\delta^2 \left(P_i + R_i\right)} \tag{6}$$

In addition to those three criteria mentioned above, we also take the interrelationship between the assessment result and the actual result into account, not only in terms of the values of the two results, but also the time relationship. In terms of the relationship between the value, a symmetric matrix C_{coe} is drawn (Eq. 7), in which $C_{20} = C_{02} \le C_{21} = C_{12} \le C_{10} = C_{01} < 0 < C_{00} \le C_{11} \le C_{22}$. The positive value in this matrix means award, while the negative value means penalty. In terms of the time relationship C_t (see Eq. 8), two coefficients c_e and c_l ($c_e \le c_l$) are proposed, for early assessment and late assessment, respectively. We define three literal evaluations for the relationship in time order between the two results, shown in Table 3. Thus, the award-penalty matrix M_{ap} (Eq. 9) has been proposed to integrate the effect of the value and the time relationship between these two results:

$$C_{coe} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix} \tag{7}$$

Table 3. Literal evaluations for assessment result

Literal evaluation		Assessment condition		
		0	**1**	**2**
actual condition	**0**	correct assessment	early assessment	early assessment
	1	late assessment	correct assessment	early assessment
	2	late assessment	late assessment	correct assessment

$$C_t = \begin{bmatrix} 1 & c_e & c_e \\ c_l & 1 & c_e \\ c_l & c_l & 1 \end{bmatrix} \tag{8}$$

$$M_{ap} = C_{coe} \cdot C_t \cdot CM \tag{9}$$

To evaluate the classification performance more accurately, we combine the aforementioned indicators to obtain the utility function f_u (Eq. 10), in which m_{ij} denotes an element in M_{ap}. This function contains two parts, F-measure and the award-penalty function value, with the weight *p* and *q*, respectively. The award-penalty value m_{ij} is mapped into the value domain of $[0,1]$ by a sigmoid function ($\frac{1}{1+e^{-x}}$):

$$f_u = p\left(\frac{1}{3}\sum_{i=1}^{3} F_i\right) + q\left(\frac{1}{1 + e^{-\frac{1}{M}\sum_{i=1}^{3}\sum_{j=1}^{3} m_{ij}}}\right) \tag{10}$$

3.2. Steps of GAEM

A systematic method for condition assessment via a genetic algorithm-based ensemble learning model is proposed in this section. The flowchart of this method is graphically shown in Fig. 1. The blue part illustrates the process for training or retraining the ensemble learning model. The yellow part illustrates the process for the genetic algorithm. Two italic abbreviations, *POP* and *IND*, are used here. *POP* denotes the population and *IND* denotes the individual in the genetic algorithm. We will elaborate the key steps in the following parts.

Class strategy: In reality, some datasets may lack the labels (class) y_i for each piece of data. In this case, first, we should label these data. For simplification, we consider RUL as the key factor in this strategy. Here, we define four terms, β_1, β_2, t_{sp} and t_f, where $\beta_1 (0 < \beta_1 \le 1)$ is a tolerance coefficient, $\beta_2 (\beta_2 \ge 1)$ is an allowance coefficient, t_{sp} denotes the setup time for maintenance, and t_f denotes the time of fault point. When $t_{rul} = \beta_1 t_{sp}$, we consider the time the boundary value between condition "Minor defect" and condition "Critical defect". In contrast, when $t_{rul} = \beta_2 t_{sp}$, we consider the time the boundary value between condition "Healthy" and condition "Minor defect". In Fig 2, we depict the class strategy and the change curves of the point value in this figure denote the key measurements from condition monitoring reflecting the health condition of the equipment.

Data preprocessing: All the features are standardized to be a Gaussian distribution with zero mean and unit variance. Standardization of datasets is a common requirement for most ensemble learning models implemented in scikit-learn [32]. With the un-standardized distributed dataset, the ensemble learning model might behave badly.

Form individual learner base: Different from previous studies, which assume that the number of each category of heterogeneous individual learners is the only one, this paper argued that the number maybe flexible. With this assumption, there are three main problems, (i) which heterogeneous individual learner should be taken in this ensemble learning model, (ii) the number of each heterogeneous individual learner, and (iii) how to rank these heterogeneous individual learners.

When tackling problem (i), some properties should be considered. Traditionally, it is generally believed that the individual learners should be as ac-

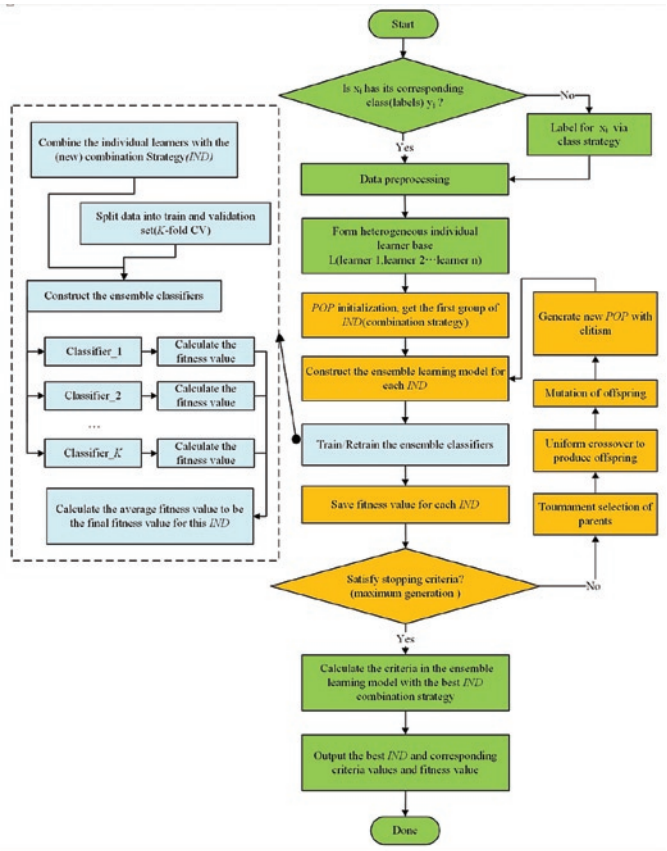


Fig. 1. Flowchart for GAEM

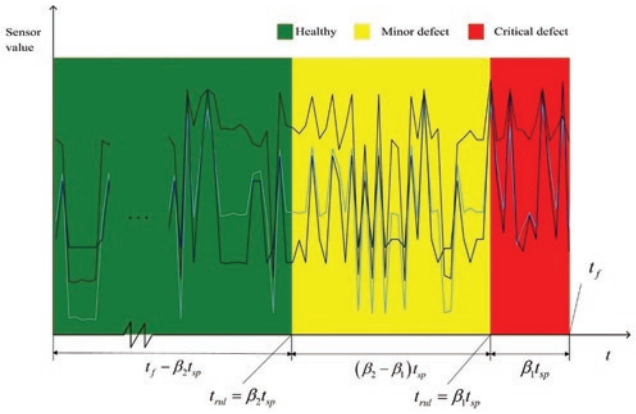


Fig. 2. Class strategy

curate as possible and as diverse as possible [16]. Moreover, the simpler the individual learner, the better performance with lower variance the ensemble learning model will get. Thus, to ensure accuracy and diversity, pilot experiments on each individual learner should be conducted beforehand. In our study, we gathered 23 individual learners from the scikit-learn class libraries [32] for selection in Table 4. Pilot experiments are conducted with the given dataset. Then, the heterogeneous individual learner base is formed through a comprehensive selection strategy composed of a series of constraints, which is shown in Eq. 11. In this equation, t_{com} and t_{int} denote the computation time for the classification model and the interval time for condition monitoring, respectively:

$$\begin{cases} P_i \geq 0.5 & i = 1,2,3 \\ R_i \geq 0.5 & i = 1,2,3 \\ t_{com} \leq \lambda t_{int} & \lambda \in (0,1] \end{cases} \quad (11)$$

Table 4. Individual learners

Individual learner	Category
Bernoulli NB	naïve Bayes
Gaussian NB	naïve Bayes
DT	tree
Extra Tree Classifier	tree
Extra Trees Classifier	tree
K Neighbors Classifier	neighbors
Nearest Centroid	neighbors
Radius Neighbors Classifier	neighbors
Linear Discriminant Analysis	discriminant analysis
Quadratic Discriminant Analysis	discriminant analysis
Linear SVC	SVM
Nu SVC	SVM
SVC	SVM
MLP	neural network
Gaussian Process Classifier	Gaussian process
Logistic Regression	linear model
Logistic Regression CV	linear model
Ridge Classifier	linear model
Ridge Classifier CV	linear model
Logistic Regression CV	linear model
SGD Classifier	linear model
Perceptron	linear model
Passive Aggressive Classifier	linear model

GA search: After selecting the suitable individual learners, GA is proposed to address the last two problems: the number of heterogeneous individual learners and how to rank these heterogeneous individual learners. In previous studies, greedy selection is the most widely used method for finding the best combination [26]. Caruana et al. [6] used greedy algorithms for searching the best ensemble combination. They added one individual leaner at each step into the ensemble combination to maximize the model performance. Greedy selection is explicable and easy to operate, but this selection has obvious limitations that these algorithms can easily be stuck in a local optimum. Additionally, as the number of individual leaners increases, the number of possible combinations for ensembles increases exponentially. An exhaustive search for the optimal combination is not practical, since evaluation of each combination is computationally expensive [31]. For this reason, heuristic algorithms, such as genetic algorithm, are more feasible for finding a near-optimal solution in a reasonable time. We used binary encoding to represent the number of each heterogeneous individual learner. The maximum number of each heterogeneous individual learner is E, and the total length of the chromosome is L. The chromosome is shown in Fig. 3. For simplicity, we set the utility function as the fitness function in this genetic process.

For the rank type of heterogeneous individual learners, we choose series connection strategy to combine heterogeneous individual learners to achieve the target that uses the next individual learner to opti-

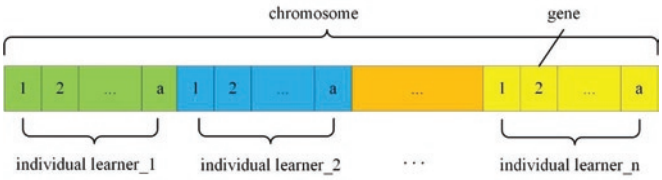


Fig. 3. Gene coding for the individual learners

mize the prior ones. In this way, the advantages of the diversity of individual learners can be retained. The results of the prior learner will be transmitted to the following ones. The samples that are correctly classified and wrongly classified will be split into two parts and greater weight will be put on the wrongly classified part. Then, these reweighted samples will serve as the new input data for the next individual learner to be reclassified. Repetition goes on until iterating through all the individual learners. In this way, we can easily see the phenomenon that individual learners increasingly focus on samples that are difficult to correctly classify, as in each round, the weight for samples in minority classes increases.

Train/Retrain the Ensemble Learning Model: The detailed processes for train/retrain the ensemble learning model are given in the blue dashed box in Fig. 1. *K*-fold cross-validation (*K*-CV) is adopted to construct a train dataset and validation dataset. The dataset is randomly and equally split into *K* folds. Out of these *K* folds, one is preserved as the validation dataset, and the other *K-1* folds are used as the training dataset. This cross-validation process is repeated *K* times, with each of the *K* folds used exactly once as the validation dataset.

After giving a certain combination strategy for the heterogeneous individual learners, the structure of ensemble learning model is confirmed. Then, this model will be trained by feeding the training dataset. After that, a well-trained ensemble learning model will be validated by inputting the validation dataset. The results, known as the assessed condition, from the well-trained ensemble learning model will be compared with the actual condition of the validation dataset. These steps will be performed *K* times to realize the *K*-fold validation. Then the average fitness values of the *K* times will be saved as the final fitness value.

3.3. A self-updating strategy for GAEM

The self-updating process runs continually in parallel with the online monitoring assessment. At a certain time interval (t_{up}), we put (save) the online monitoring data into the historical database (the database for GAEM) to form a new database. In this way, both the data volume and diversity are increased. By periodically inspecting and learning kinds of new situations, we can strengthen the reliability and adaptability of GAEM and to keep the classifier tightly tied to the newest equipment situation. With retraining/relearning from the new database, the parameters in GAEM are self-updated, and we named the new ensemble learning model (GAEM-II). The processes for GAEM and GAEM-II are illustrated in Fig. 4, there they are shown by solid line and imaginary line, respectively.

Through GAEM, we gain the Trained Classifier, which will give the condition assessment result ('0', '1', or '2') by feeding the real-time monitoring data. After t_{up} , a new dataset is formed by adding the monitoring data into the historical dataset. Then, we retrain/relearn GAEM to update the parameters to attain GAEM-II. When GAEM-II is confirmed, GAEM will be replaced by GAEM-II to undertake in the work of real-time condition assessment for time t_{up} . After time t_{up} , GAEM-II will be re-updated. That means the self-updating process is always running in parallel with real-time assessment, and the period time for this process is t_{up} . Here, we do not consider the updating time, because the time for updating the process is much faster than the monitoring interval t_{int} .

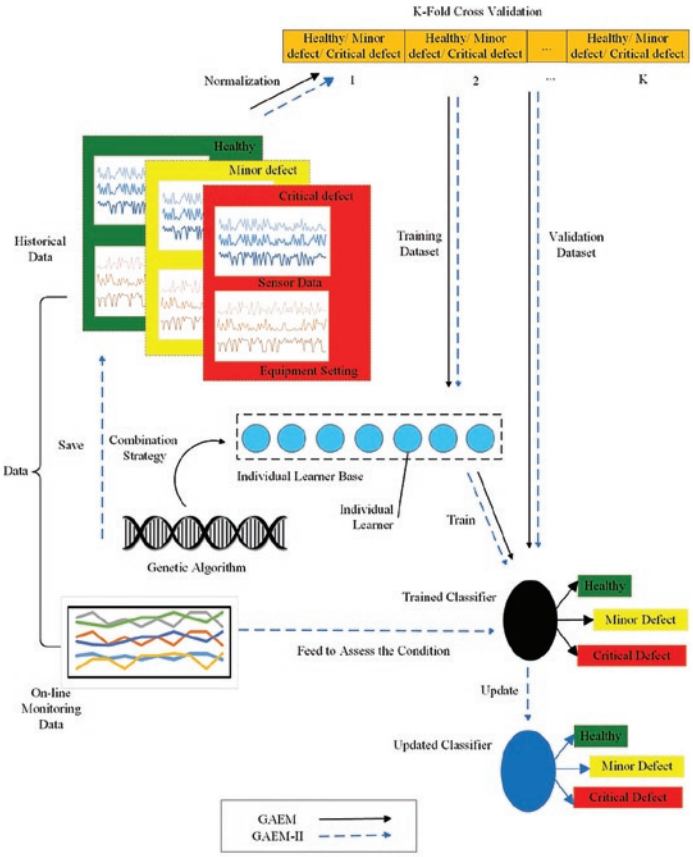


Fig. 4 process of GAEM-II

4. Experiments and results

In this section, to evaluate the performance of the proposed model, numerical experiments, including two types of comparison studies, are described. The first type of comparison study is composed of (i) comparison with individual leaners, (ii) comparison with its homogeneous ensemble learning models, and (iii) comparison with common heterogeneous ensemble learning models. Another type of comparison study has been conducted by comparison with three popular ensemble learning models, namely, Adaboost, Random Forest and Gradient Boosting.

4.1. Dataset description

The dataset comes from the prognostics challenge competition at the International Conference on Prognostics and Health Management (PHM 2008). The dataset contains multiple multivariate time series, which are the life-cycle data of different engines, and the engines can be considered to be of the same type. Each engine starts from a different condition, and the degree of initial wear and variation is different and unknown. Therefore, the engine can be perfect or imperfect but not failing. In addition, the dataset contains noise and perturbations because of sensor noise. There are two types of data in this dataset, 3 operational settings data (internal data) and 21 sensor measurement data (external data data). All the experiments in this study were executed in an Wicro-Star with NVidia GeForce GTX 1050Ti GPU, an Intel Core i7-7700HQ (3.6 GHz, 4 cores) CPU and 16 GB RAM. All individual leaners are implemented from the software library skit-learn, and all codes are written by Python 3.6.

In the experiments part, the parameters are set as follows (Table 5). F1-measure (F1 for short, seen in Eq. 12) is utilized to balance the effect of precision and recall. Eq. 13 gives the expressions of the coefficient matrix C_{coe} :

$$F1_i = \frac{2P_i \times R_i}{P_i + R_i} \tag{12}$$

$$C_{coe} = \begin{bmatrix} 1 & -3 & -7 \\ -3 & 5 & -5 \\ -7 & -5 & 7 \end{bmatrix} \tag{13}$$

Table 5. Parameter values

Parameter	Value
δ	1
β_1	0.3
β_2	1.2
t_{sp}	$10t_{int}$
c_e	1.2
c_l	1.5
p	0.5
q	0.5
E	15
L	4
K	6

As this dataset does not have labels for each piece of data, we first label this dataset through the class strategy mentioned in Section 3. The distribution of this dataset (Fig. 5) shows the majority of samples belong to condition "Healthy", which is more than 26 times to condition "Critical defect".

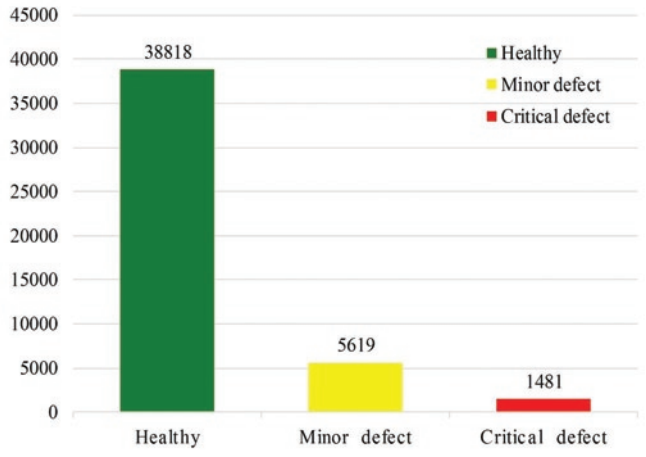


Fig. 5. Distribution of samples

To verify the challenge in class-imbalanced classification, first we use DT, which is a prevalent model for classification, to classify this dataset. We set the maximum depth of the tree to be 4. Table 6 illustrates the criteria results for the DT, and Fig. 6 shows two confusion matrixes, the original confusion matrix on the left-hand and the normalized confusion matrix on the right-hand. It is obvious that the precision, recall and F1 are very high (approximately 0.947) in class 0, while in class 1 and class 2, these three criteria are exceedingly low (approximately 0.562 and 0.130, respectively), which reflects that this DT classifier has a superior classification ability for the class with major samples, but a weakness for classes with minor samples, because the number of samples in class 2 are so few that the DT Classifier cannot accurately learn the features and properties of this class. Because of the skewed trend between classes, the fewest samples among class 2 are likely to be treated as noise, which also reduce the criteria for this class. In addition, in class 0, no samples among these 13776 test samples are classified as class 2, but in class 2, most samples are wrongly classified, with 76.4 percent samples classified as class 1 and 22.5 percent samples classified as class 0. Only 5 samples are correctly classified. There is another reason for the phenomenon that class 0 and class 2 are obstructed by class 1, it is easier to distinguish samples from class 0 vs. samples from class 2 than distinguish samples from class 0 vs. samples from class 1.

Table 6. criteria results for DT

Criteria	Class 0	Class 1	Class 2
Precision	0.935	0.540	0.357
Recall	0.959	0.585	0.011
F1	0.946	0.562	0.022
Utility	0.572		

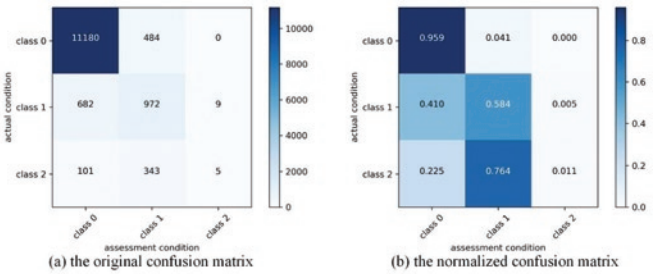


Fig. 6. Confusion matrixes with DT

In reality, the conditions with relatively few samples, namely, class 1 (condition "Minor defect") and class 2 (condition "Critical defect") are more important than class 0 (condition "Healthy"). Managers will put more emphasis on the classification performance of class 1and class 2. As traditional methods show weakness in terms of classes with relatively few samples, more suitable approaches should be proposed to achieve better performance in this class-imbalanced classification.

4.2. Experiment with GAEM

In this section, we report the results of the experiment performed with GAEM on the PHM 2008 database. After data pretreatment, the normalized dataset is attained. Then, pilot experiments on the 23 individual learners are conducted. Each of classifier is used to train the classification model on train dataset, and test on validation dataset. According to the selection strategy, we obtain the specified classifiers to form the individual learner base in Table 7.

The genetic algorithm searched for the optimal combination strategy of heterogeneous individual learners in GAEM: $[8,7,5,4,7,5,9]$. The sequence of these 7 heterogeneous individual learners is Logistic Regression (LR), KNN, DC, Extra Tree Classifier (ETC), Quadratic Discriminant Analysis (QDA), MLP and SVC. Table 8 illustrates the criteria results for GAEM, and Fig. 7 shows two confusion matrixes, the original confusion matrix on the left-hand and the

Table 7. Individual learner base

Individual learner	Class libraries
SVC	SVM
KNN	neighbors
DT	tree
Extra Tree Classifier	tree
Quadratic Discriminant Analysis	discriminant analysis
MLP	neural network
Logistic Regression	linear model

normalized confusion matrix on the right-hand. Table 9 illustrates the computation time of three main activities. The GA processing aims to find the proper individual learners' combination, and it costs approximately 2 hours. In addition, approximately 20 minutes are consumed for GAEM Training. These tow processes are conducted off-line, which means it is accessible for the manager, because they do not disturb real-time condition assessment process. For the on-line part, the proposed model GAEM only costs less than one second to assess the condition that the equipment remain in, which make it possible for real-time condition assessment. Therefore, even though the off-line parts are computationally expensive, the model GAEM still have the superiority to perform well in real-time equipment condition assessment.

Table 8. Criteria results for GAEM

Criteria	Class 0	Class 1	Class 2
Precision	0.984	0.848	0.806
Recall	0.986	0.846	0.791
F1	0.985	0.847	0.798
Utility	0.835		

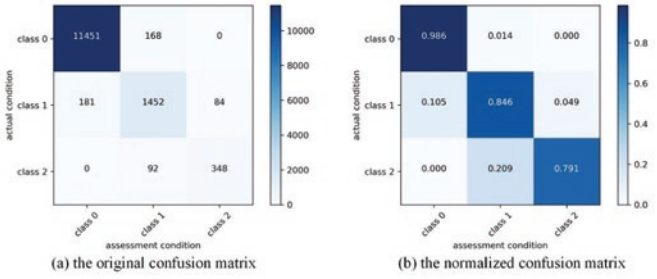


Fig. 7. Confusion matrixes with GAEM

According to Fig. 8, the upper right corner and lower left corner of the confusion matrix are equal to 0, which means among 13776 test samples, no samples in class 0 is classified into class 2, and similarly, no samples in class 2 are classified into class 0. These results show the superiority of GAEM when recognizing between class 0 and class 2.

Table 9. Computation time for activities

Activities	GA Process	GAEM Training	Real-time Condition Assessment
Property	off-line	off-line	on-line
Time (s)	7315.128	1161.243	0.002

As samples deteriorated from class 0 to class 2, the classification performance on different classes show a large difference, with an obvious downtrend in each criterion from class 0 to class 2.

4.3. Comparison studies

4.3.1. Comparison with individual learners, homogeneous and heterogeneous ensemble models

Comparison with Individual Leaners: To verify that the ensemble learning model will perform better than its individual leaners, we ran experiments on the 7 heterogeneous individual leaners. Fig. 8 shows the reports for criteria on these models. It is obvious that GAEM outperforms any individual leaners. So combining individual leaners indeed has the ability to optimize the classification performance.

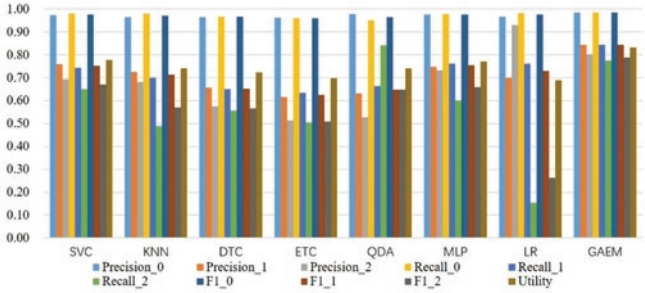


Fig. 8. Reports for individual leaners and GAEM

Comparison with Homogeneous Ensemble Learning Models: To verify the competiveness of the modified heterogeneous ensemble learning model to homogeneous ensemble models, we ran a set of comparison experiments. The homogeneous ensemble learning models here are composed of the individual learners used in GAEM, with the same number and the same parameter setting. The result is shown in Fig. 9. Comparing there results to those in Fig. 8, for most individual leaners, as the number of individual leaner increases, some criteria show better results because of the ensemble effect. Each individual learner1 optimizes the prior ones by adding the weight for the wrongly classified samples, while reducing the weights of correctly classified samples. On the other hand, GAEM still performs better than these homogenous ones, because GAEM has diversity, as it contains different types of heterogeneous individual learners, which the homogenous ensemble learning models do not have.

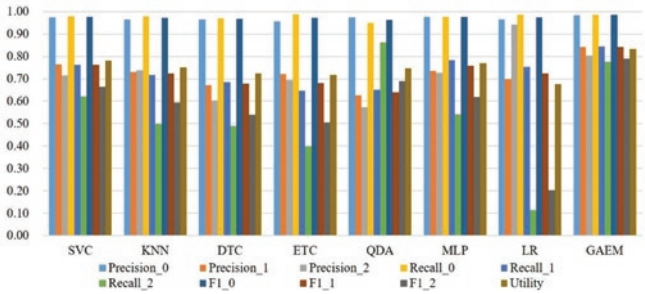
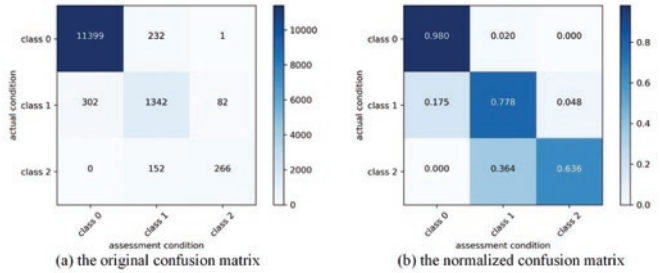


Fig. 9. Reports for homogenous ensemble learning models and GAEM

Comparison with the Common Heterogeneous Ensemble Learning Model: To verify that heterogeneous ensemble learning model constructed by a certain number of heterogeneous individual learners has better performance than heterogeneous ones with only one learner in every heterogeneous individual learner category, we ran this comparison study between the common heterogeneous ensemble learning model

and GAEM. The common heterogeneous ensemble learning model is composed of the 7 heterogeneous individual learners that have been chosen in GAEM. The result is shown in Fig. 10 and the criteria results are compared in Fig. 11.

From the confusion matrix in Fig. 10, the performance of this model is good and acceptable with a majority of samples correctly classified. That means combining these individual learners together to form this heterogeneous ensemble learning model is feasible and reasonable. Fig. 13 further shows that having more than one classifier in each heterogeneous individual learner will have better performance. That means it is indeed an optimal statedy to increase the number of each heterogeneous individual learners.



(a) the original confusion matrix (b) the normalized confusion matrix

Fig. 10. Confusion matrixes with the common heterogeneous ensemble learning model

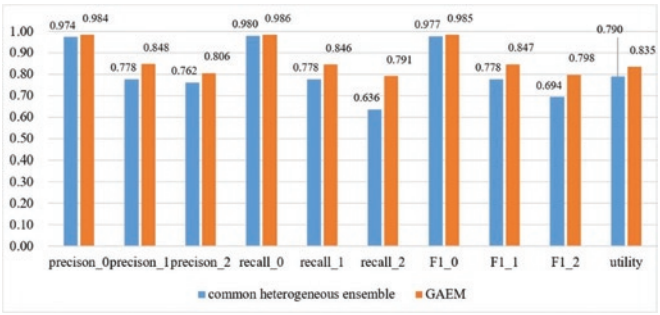


Fig. 11. Criteria results for the common heterogeneous ensemble learning model and GAEM

4.3.2. Comparison with Adaboost, Random Forest and Gradient Boosting

To validate the competitive performance, we contrasted the performance of GAEM with three popular ensemble learning models, namely, Adaboost, Random Forest and Gradient Boosting, under the same experimental setting on the PHM 2008 dataset.

To better understand the detailed performance, we have drawn the normalized confusion matrix for each classifier in Fig. 12, and Table 10 illustrates the performance of these four models. Adaboost and Random Forest show disadvantages in the recognition between class 0 and class 2 compared to Gradient Boosting and GAEM. All four methods perform well in recognizing class 0, as reflected in the high values for precision, recall and F1, because class 0 contains sufficient samples, which make it more amenable to learning the intrinsic properties of this class. However, Adaboost, Random Forest, and Gradient Boosting perform relatively badly on class 1, with all criteria under 0.8. This situation becomes worse in class 2, especially in terms of recall and F1, which fluctuate in $[0.4, 0.7]$. GAEM has higher precision and recall for class 1 and class 2, so better generalization is gained through this proposed method.

To compare the stability of these four models, a box-plot is drawn in Fig. 13. Because there is just a fine fluctuation in the criteria for class 0, the boxes for class 0 are omitted. It is clear from the figure that GAEM outperforms other ensemble learning models in terms of the stability of most reported criteria. On the criteria for class 2, these ensemble learning models perform unstably, with a wide range of fluctuations, while GAEM shows good stability and reliability with little fluctuations and high scores in all criteria.

4.4. Experiment on GAEM-II

In this section, we report the result of the experiments performed with GAEM-II by adding samples in the original dataset to form the new dataset. In four experiments, we add 5000 samples, 10,000 samples, 15,000 samples, and 20,000 samples, respectively. Then, we retrain the learning model to obtain the updated GAEM-II. The results are shown in Table 11 and the confusion matrixes are shown in Fig.14, Fig. 15, and Fig. 16, in which GAEM-II.1, GAEM-II.2, and GAEM-II.3 denote these three new models with these three modified datasets. It is obvious that as the number of samples increase, the utility shows

Table 10. Criteria results for Adaboost, Random Forest, Gradient Boosting and GAEM

Classifier	Class 0			Class 1			Class 2			Utility
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	
Adaboost	0.976	0.964	0.970	0.654	0.788	0.715	0.677	0.372	0.480	0.726
Random Forest	0.969	0.985	0.977	0.937	0.941	0.739	0.786	0.428	0.554	0.748
Gradient Boosting	0.974	0.983	0.979	0.770	0.772	0.771	0.760	0.591	0.665	0.782
GAEM	0.984	0.986	0.985	0.848	0.846	0.847	0.806	0.791	0.798	0.835

Table 11. Criteria results for GAEM-II

Classifier	Class 0			Class 1			Class 2			Utility
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	
GAEM	0.984	0.986	0.985	0.848	0.846	0.847	0.806	0.791	0.798	0.835
GAEM-II.1	0.988	0.988	0.988	0.867	0.849	0.857	0.761	0.821	0.790	0.836
GAEM-II.2	0.986	0.833	0.985	0.853	0.864	0.858	0.814	0.822	0.818	0.843
GAEM-II.3	0.982	0.987	0.989	0.869	0.892	0.880	0.801	0.826	0.813	0.848

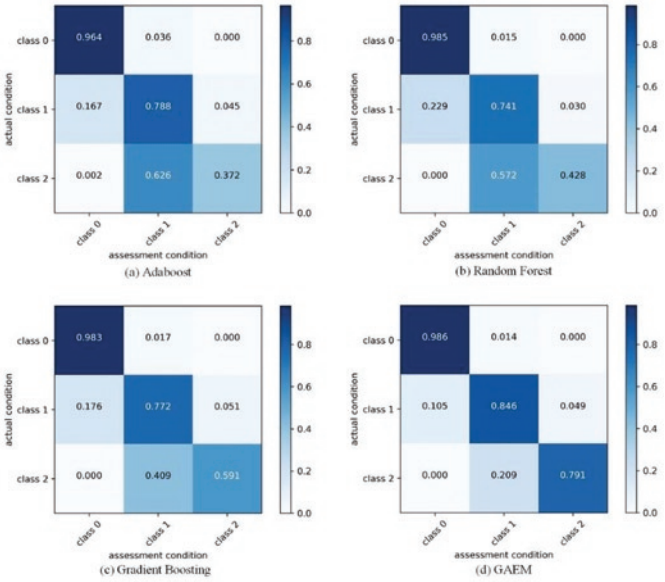


Fig. 12. The normalized confusion matrixes for Adaboost, Random Forest, Gradient Boosting and GAEM

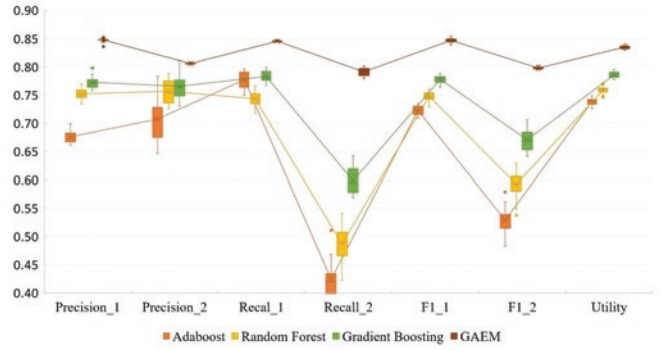


Fig. 13. Box-plot for Adaboost, Random Forest, Gradient Boosting and GAEM

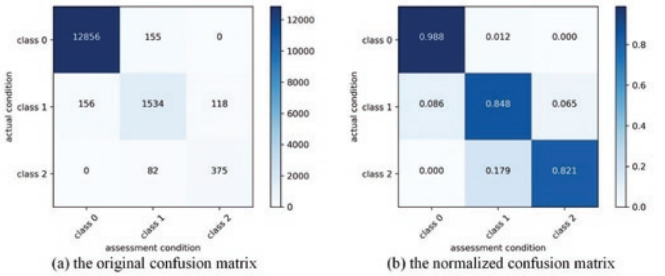


Fig. 14. Confusion matrixes with GAEM-II.1

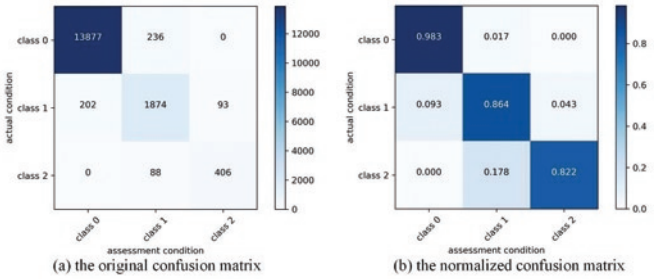


Fig. 15. Confusion matrixes with GAEM-II.2

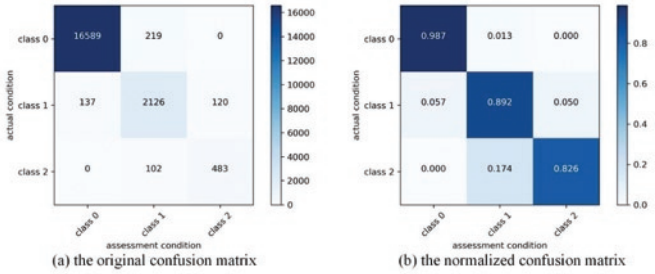


Fig. 16. Confusion matrixes with GAEM-II.3

an uptrend from 0.835 to 0.848. That means GAEM-II can optimize the classification performance of GAEM through adding more samples to the dataset.

5. Conclusion

This study developed a genetic algorithm-based search method, replacing greedy search and exhaustive search, to search for a combination strategy for a heterogeneous ensemble learning model. In addition, a new attempt was made to modify the traditional heterogeneous ensemble learning models. We argue that the heterogeneous ensemble learning model constructed from a number of heterogeneous individual learners has better classification performance than that of heterogeneous models that only have one learner in every heterogeneous individual learner category. We made experiments and comparison studies to verify this opinion.

Another contribution of this study lies in the effectiveness of the proposed model. In contrast to other condition assessment method, the proposed method does not require feature extraction or indicators setting to assess the equipment condition. The proposed method can automatically extract the inherent and generalizable features of the dataset. In addition, with our model, real-time equipment condition assessment can be achieved, depending on the fast computation and the self-updating learning strategy. The biggest advantages of the proposed condition assessment method are the accuracy and stability in this class-imbalanced classification problem.

Our study discussed supervised classification for the equipment condition assessment with class-imbalanced dataset. Future work can also explore semi-supervised classifications in this field, as label process is costly and less available in reality.

Acknowledgements

This research was co-supported by the National Science Foundation of China (No. 51035008), National Science & Technology Major Project of China (No. 2016ZX04004-005) and Fundamental Research Funds for the State Key Laboratory of Mechanical Transmission of Chongqing University (No. SKLMT-ZZKT-2017M16).

References

1. Albisua I, Arbelaitz O, Gurrutxaga I, Lasarguren A, Muguerza J, Pérez JM. The quest for the optimal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets. Progress in Artificial Intelligence 2013; 2(1): 45-63, https://doi.org/10.1007/s13748-012-0034-6.
2. Amir M D M, Muttalib E S A, editors. Health index assessment of aged oil-filled ring main units. Power Engineering and Optimization Conference; 2014, https://doi.org/10.1109/PEOCO.2014.6814452.
3. Baik H-S, Jeong H S, Abraham D M. Estimating transition probabilities in Markov chain-based deterioration models for management of wastewater systems. Journal of water resources planning and management 2006; 132(1): 15-24, https://doi.org/10.1061/(ASCE)0733-9496(2006)132:1(15).
4. Benkedjouh T, Medjaher K, Zerhouni N, Rechak S. Health assessment and life prediction of cutting tools based on support vector regression. Journal of Intelligent Manufacturing 2015; 26(2): 213-223, https://doi.org/10.1007/s10845-013-0774-6.
5. Bennin K E, Keung J, Phannachitta P, Monden A, Mensah S. Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Transactions on Software Engineering 2018; 44(6), https://doi.org/10.1109/TSE.2017.2731766.
6. Caruana R, Niculescu-Mizil A, Crew G, Ksikes A, editors. Ensemble selection from libraries of models. International Conference on Machine Learning 2004, https://doi.org/10.1145/1015330.1015432.
7. Carvalho E, Tang F, Allen E, Sharma P, editors. A Case Study of Asset Integrity and Risk Assessment for Subsea Facilities and Equipment Life Extension. Offshore Technology Conference 2015, https://doi.org/10.4043/25701-MS.
8. Chan C W, Paelinckx D. Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. Remote Sensing of Environment 2008; 112(6): 2999-3011, https://doi.org/10.1016/j.rse.2008.02.011.
9. Chawla N V, Bowyer K W, Hall LO, Kegelmeyer W P. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 2002; 16: 321-357, https://doi.org/10.1613/jair.953.
10. Cheng F, Zhang J, Wen C. Cost-Sensitive Large margin Distribution Machine for classification of imbalanced data. Pattern Recognition Letters 2016; 80: 107-112, https://doi.org/10.1016/j.patrec.2016.06.009.
11. Fan M, Zeng Z, Zio E, Kang R. Modeling dependent competing failure processes with degradation-shock dependence. Reliability Engineering & System Safety 2017; 165: 422-430, https://doi.org/10.1016/j.ress.2017.05.004.
12. Fan M, Zeng Z, Zio E, Kang R, Chen Y. A stochastic hybrid systems based framework for modeling dependent failure processes. PloS one 2017; 12(2), https://doi.org/10.1371/journal.pone.0172680.
13. Giorgio M, Guida M, Pulcini G. An age- and state-dependent Markov model for degradation processes. IIE Transactions 2011; 43(9): 621-632, https://doi.org/10.1080/0740817X.2010.532855.
14. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications 2017; 73: 220-239, https://doi.org/10.1016/j.eswa.2016.12.035.
15. Haque M N, Noman N, Berretta R, Moscato P. Heterogeneous Ensemble Combination Search Using Genetic Algorithm for Class Imbalanced Data Classification. Plos One 2016; 11(1): e0146116, https://doi.org/10.1371/journal.pone.0146116.
16. Hastie T, Tibshirani R, Friedman J. Ensemble Learning: Springer New York; 2009. 605-624.
17. Japkowicz N. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. Aaai Workshop on Learning from Imbalanced Data Sets 2000: 10-15.
18. Kleiner Y, Sadiq R, Rajani B B. Modeling Failure Risk in Buried Pipes Using Fuzzy Markov Deterioration Process. Pipeline Engineering and Construction@sWhat's on the Horizon; 2004.
19. Lee W, Jun C-H, Lee J-S. Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. Information Sciences 2017; 381: 92-103, https://doi.org/10.1016/j.ins.2016.11.014.
20. Li Z, Zhang B, Wang Y, Chen F, Taib R, Whiffin V, et al. Water pipe condition assessment: a hierarchical beta process approach for sparse incident data. Machine Learning 2014; 95(1): 11-26, https://doi.org/10.1007/s10994-013-5386-z.
21. López A J G, Márquez A C, Macchi M, Fernández JFG. Prognostics and Health Management in Advanced Maintenance Systems. Advanced Maintenance Modelling for Asset Management: Springer; 2018. 79-106, https://doi.org/10.1007/978-3-319-58045-6_4.
22. López V, Fernández A, García S, Palade V, Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences 2013; 250: 113-141, https://doi.org/10.1016/j.ins.2013.07.007.
23. López V, Fernández A, Herrera F. On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. Information Sciences 2014; 257: 1-13, https://doi.org/10.1016/j.ins.2013.09.038.
24. Luengo J, Fernández A, García S, Herrera F. Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling. Soft Computing 2011; 15(10): 1909-1936, https://doi.org/10.1007/s00500-010-0625-8.
25. Huk M, Szczepanik M, Multiple classifier error probability for multi-clas problems. Eksploatacja i Niezawodnosc - Maintenance and Reliability 2011; 51(3): 12-16.
26. Margineantu D D, Dietterich T G, editors. Pruning Adaptive Boosting. Fourteenth International Conference on Machine Learning; 1997
27. Mathew J, Pang C K, Luo M, Leong W H. Classification of imbalanced data by oversampling in kernel space of support vector machines. IEEE Transactions on Neural Networks and Learning Systems 2018; 29(9): 4065-4076, https://doi.org/10.1109/TNNLS.2017.2751612.
28. Oreški Stjepan, Oreški Goran. Cost-Sensitive Learning from Imbalanced Datasets for Retail Credit Risk Assessment. TEM JOURNAL - Technology, Education, Management, Informatics 2018.
29. Othman A, Tahir M, El Shatshat R, Shaban K, editors. Application of ensemble classification method for power transformers condition assessment. Electrical and Computer Engineering (CCECE), 2017 IEEE 30th Canadian Conference; 2017: IEEE.
30. Parradohernández E, Robles G, Ardilarey J A, Martíneztarifa J M, Sciubba E. Robust Condition Assessment of Electrical Equipment with One Class Support Vector Machines Based on the Measurement of Partial Discharges. Energies 2018; 11(3).
31. Partalas I, Tsoumakas G, Vlahavas I, editors. Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection. Conference on ECAI 2008: European Conference on Artificial Intelligence; 2008.

32. Pedregosa F, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 2011; 12(10): 2825-2830.
33. Razi-Kazemi A A, Vakilian M, Niayesh K, Lehtonen M. Data Mining of Online Diagnosed Waveforms for Probabilistic Condition Assessment of SF6 Circuit Breakers. IEEE Transactions on Power Delivery 2015; 30(3): 1354-1362, https://doi.org/10.1109/TPWRD.2015.2399454.
34. Reddy M V, Sodhi R. A rule-based S-Transform and AdaBoost based approach for power quality assessment. Electric Power Systems Research 2016; 134: 66-79, https://doi.org/10.1016/j.epsr.2016.01.003.
35. Sugier J, Anders G J. Modelling and evaluation of deterioration process with maintenance activities. Eksploatacja i Niezawodnosc - Maintenance and Reliability 2013; 15(4): 305-311.
36. Tseng M-L, Wu K-J, Ma L, Kuo TC, Sai F. A hierarchical framework for assessing corporate sustainability performance using a hybrid fuzzy synthetic method-DEMATEL. Technological Forecasting and Social Change 2017, https://doi.org/10.1016/j.techfore.2017.10.014.
37. Xu J, Xu L. Integrated system health management-based condition assessment for manned spacecraft avionics. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 2013; 227(1): 19-32, https://doi.org/10.1177/0954410011431395.
38. Zadrozny B, Langford J, Abe N, editors. Cost-sensitive learning by cost-proportionate example weighting. Data Mining, 2003 ICDM 2003 Third IEEE International Conference on; 2003.
39. Zhang L, Zhang J, Zhai H, Zhou S. A new assessment method of mechanism reliability based on chance measure under fuzzy and random uncertainties. Eksploatacja i Niezawodnosc - Maintenance and Reliability 2018; 20(2): 219-228, https://doi.org/10.17531/ein.2018.2.06.
40. Zhu L, Lu C, Dong Z Y, Hong C. Imbalance Learning Machine-Based Power System Short-Term Voltage Stability Assessment. IEEE Transactions on Industrial Informatics 2017; 13 (5): 2533-2543, https://doi.org/10.1109/TII.2017.2696534
41. Zhu L J, Cong H. The State Assessment of Armored Vehicle Engine Based on Analytic Hierarchy Process and Fuzzy Synthetic Evaluation. Advanced Materials Research 2014; 988(988): 606-610, https://doi.org/10.4028/www.scientific.net/AMR.988.606.

Xiaohui CHEN
Zhiyao ZHANG
Ze ZHANG
The State Key Lab of Mechanical Transmission
Chongqing University
Chongqing, China

E-mail: chenxiaohui@cqu.edu.cn, 20160701007z@cqu.edu.cn, zhangzexmail@163.com