Received March 21, 2019, accepted April 11, 2019, date of publication April 22, 2019, date of current version May 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912631

Concept Drift Detection and Adaption in Big Imbalance Industrial IoT Data Using an Ensemble Learning Method of Offline Classifiers

CHUN-CHENG LIN[1], (Senior Member, IEEE), DER-JIUNN DENG[2], (Member, IEEE), CHIN-HUNG KUO[1], AND LINNAN CHEN[1]

[1]Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan
[2]Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua 500, Taiwan

Corresponding author: Der-Jiunn Deng (djdeng@cc.ncue.edu.tw)

This work was supported in part by the Industrial Technology Research Institute, Taiwan, under the project of Big Data Technologies and Applications.

ABSTRACT In a smart factory, thousands of industrial Internet of Things (IIoT) devices or sensors are installed in production machines to collect big data on machine conditions and transmit it to a cyber-physical system in the cloud center of the factory. Then, the system employs a variety of condition-based maintenance (CBM) methods to predict the time point when machines start to be operated abnormally and to maintain them or replace their components in advance so as to avoid manufacturing enormous detective products. CBM suffers from problems of concept drifts (i.e., the distribution of fault patterns may change over time) and imbalance data (i.e., the data with faults accounts for a minority of all data). Ensemble learning that integrates the diversity of multiple classifiers provides a high-performance solution to address these problems. In practice, most companies may not have a sufficient budget to establish a sound infrastructure to support real-time online classifiers, but may have off-the-shelf offline classifiers in their existing systems. However, most previous works on ensemble learning only focused on supporting online classifiers. Consequently, this work proposes an ensemble learning algorithm that supports offline classifiers to cope with three-stage CBM with concept drifts and imbalance data, in which Stages 1 (training an ensemble classifier) and 3 (creating a new ensemble) employ an improved Dynamic AdaBoost.NC classifier and the SMOTE method to address imbalance data; and Stage 2 (detecting concept drifts in imbalance data) employs an improved LFR (Linear Four Rates) method. The experimental results on datasets with different degrees of imbalance show that the proposed method can successfully detect all concept drifts, and has a high accuracy rate in detecting minority-class data, which is over 94%.

INDEX TERMS Ensemble learning, imbalance data, concept drift, data adaption, smart manufacturing, Industry 4.0.

I. INTRODUCTION

The industrial Internet of Things (IIoT) has been driving development and advances of smart manufacturing and Industry 4.0 [1], from conventional manufacturing to smart manufacturing. More and more IIoT technologies and facilities are incorporated into manufacturing factories. Generally, a large number of IIoT devices or sensors are attached to machines in the factory. Enormous machine conditions are collected continuously and in time, and are uploaded to the cloud center of the factory, in which production managers can adopt cyber-physical systems to control all operations of each machine ideally in real time.

In practice, machine components get aging over time. If they were not replaced in time, enormous defective or low-quality products would be manufactured, and machines would perform abnormally or be damaged. Therefore, condition-based maintenance (CBM) is to analyze conditions of machine components collected by IIoT devices or sensors to predict the time point when they start to perform abnormally and to replace them in advance. Since customers have continuously requested a higher product quality,

The associate editor coordinating the review of this manuscript and approving it for publication was Patrick Hung.

manufacturing factories need to take more attention on improving the product quality by CBM. However, development of CBM in real factories has been increasingly challenging because it requires to consider concept drifts and imbalance data.

Distribution of fault patterns in the collected data forms a concept. However, when machine components get aging or are maintained/replaced, the concept of fault patterns changes to be with different features, so that the CBM method without adaption to this new concept performs worse. On the other hand, with rapid advances in manufacturing technologies, machines become much precise and make rare faults. Hence, the amount of fault data points (called the *minority class*) is rare as compared to that of normal data points (called the *majority class*). Such an imbalance data distribution makes it difficult to differentiate faults from normal data.

To concurrently address the classification problems with concept drifts and imbalance data, most previous works focused on online ensemble learning (e.g., [2]–[4]), which integrates diversity of multiple online classifiers to address these problems. However, online learning is much suitable for real-time systems, which require support of advanced infrastructures that cost a lot. Additionally, most online classifiers are simple models or can only train a small amount of data, so that a large amount of data is not considered in a total. In practice, companies have off-the-shelf offline classifiers according to existing infrastructures of their factories. It would be convenient for them to design an ensemble learning method based on offline classifiers.

To the best of our understanding, no previous works proposed any ensemble learning method based on offline classifiers to address concept drifts and imbalance data concurrently. Therefore, this work proposes an ensemble learning method called *dynamic AdaBoost.NC with multiple subclassifiers for imbalance and drifts* (DAMSID) for coping with CBM with concept drifts and imbalance data. The work in [2] proposed a three-stage online ensemble learning framework based on *diversity for dealing with drifts* (called DDD) to address concept drifts: ensemble learning, concept drift detection, and drift adaptation. Following the three-stage DDD framework, the proposed DAMSID improves the ensemble learning methods used at Stages 1 and 3 with an offline ensemble learning method called Dynamic AdaBoost.NC [5] and SMOTE (Synthetic Minority Oversampling TEchnique) and improves the drift detection method at Stage 2 with the LFR (Linear Four Rates) method [6] to address concept drifts in imbalance data. By simulation on the datasets with four degrees of imbalance data, performance of the proposed DAMSID is evaluated.

The contributions of this work are as follows: The proposed DAMSID provides an ensemble learning method based on offline classifiers to address the CBM with concept drifts and imbalance data, in which the main components (Dynamic AdaBoost.NC and LFR) are improved and adjusted to solve the problem. Because most companies have off-the-self offline classifiers based on existing infrastructures, they can employ the proposed method with no need to invest in more facilities. In addition, the forecast decision of the proposed ensemble learning method is based on a weighted voting of multiple offline subclassifiers, and it is easy to extend the proposed method with a variety of offline classifiers.

The rest of this work is organized as follows. Section II gives the related works. Section III gives the details on the proposed three-stage ensemble learning method, and Section IV gives implementation of the proposed method and experimental analysis. Section IV concludes this work with future work.

II. RELATED WORK

This work first gives the literature review on CBM, and then introduces the ensemble learning methods for the CBM. Next, this work reviews the ensemble learning methods for addressing concept drifts and imbalance data, respectively. Finally, an ensemble learning method for coping with them together is reviewed.

A. CBM

The major factors that affect variation of machines are from a lot of complicated machine control components, which are in charge of controlling quantity of physical property (e.g., pressure and temperature) or quantity of chemicals (e.g., those added to production processes) during manufacturing products. In practice, control components are aging as their usage time increases. If they are not maintained or replaced in time, the machines manufactures enormous defective products, and more seriously, they are broken. Therefore, CBM is to predict the time point when the control components attached to machines start to perform abnormally, and to replace them in advance.

Generally, production managers are based on their domain knowledge and previous experiences to judge the time point when a control component starts to perform abnormally, but the judgement may not always be precise. If the judged time is earlier (i.e., the component still performs well but is replaced earlier), the cost of control components increases. If the judged time is too late (i.e., the component should be replaced but not in actual), the machine could be malfunctioned. Conventionally, CBM methods have been developed based on statistics theory, e.g., regression analysis [7], time series [8], and data mining [9]. These methods were based on historical observations to search for a trend or pattern of the concerned problem, and then predicted the future event according to this trend or pattern. However, to achieve a high accuracy level, these methods required a large number of effective observations, and supposed them to follow a certain probability distribution (e.g., normal distribution and Poisson distribution). That is, only when the above two conditions were met, these methods performed well.

On machine learning methods for CBM, Jardine *et al.* [10] reviewed a lot of methods based on artificial neural networks for CBM. Caesarendra *et al.* [11] proposed a CBM approach that integrates relevance vector machine (RVM) and logistic regression (LR) to predict the machine aging time. Patel and Giri [12] applied the random forest (RF) classifier to detect multiclass mechanical faults in bearing of an induction motor. Lin *et al.* [13] proposed a novel hybrid grey forecasting and harmony search approach, in which grey forecasting was shown to perform well for small data samples. Wan *et al.* [14] proposed a big data solution for active preventive maintenance in manufacturing environments.

B. ENSEMBLE LEARNING

Ensemble learning is a supervised machine learning method [15]. The idea of ensemble learning is to consider a "committee" consisting of a number of "experts" (i.e., machine learning models), and to determine a final result according to a certain voting scheme of all the experts. Different from conventional forecasting methods that adopted only a single model to forecast, ensemble learning incorporates the forecast results from multiple models into a single forecast result. Some works have showed that ensemble learning often performs better than any single forecasting model [16].

Ensemble learning was originated from [17], which adopted multiple classifiers to divide the feature space. Hansen and Salamon [18] proposed an ensemble learning algorithm similar to artificial neural networks, and showed that it can increase performance of conventional classification methods in addressing classification problems. Schapire [19] proposed an ensemble learning algorithm based on boosting, and showed that it can effectively reduce the forecast error rate in solving binary classification problems, so that ensemble learning receives a lot of attention. Ensemble learning is suitable for the problems in which the data is of a huge scale and is hard to be computed, or in which the data samples are too few or are hard to be obtained. The latter case can be addressed by bootstrapping [20]. In addition, for the classification problems for imbalance data, the amount of data samples in the minority class is increased by oversampling methods, e.g., SMOTE [21]. On the other hand, the amount of data samples in the majority class is decreased by undersampling methods, but undersampling methods have a drawback of losing partial information of the minority class [22].

C. ENSEMBLE LEARNING FOR CONCEPT DRIFTS

At a certain time point, consider a data instance in which each data point has a feature vector X and a class label y in the feature space. The joint distribution of these feature vectors and class labels is denoted by $p(X, y)$, which is called a *concept*. A *concept drift* is the process in which the original joint distribution changes to a new joint distribution [2].

In practice, the environment of the time point when to replace machine components changes dynamically over time, e.g., the environment changes when machine components are replaced or get aging. These local changes would change the whole machine environment, so that the concept changes.

On the ensemble learning algorithms that address concept drifts, Minku and Yao [2] proposed an ensemble learning algorithm based on *diversity for dealing with drifts* called DDD. When a concept drift is detected, the DDD trains two new classifiers based on the datasets with high and low diversities, respectively, and incorporates them with the two original classifiers with high and low diversities, respectively, to adapt to concept drifts. Kolter and Maloof [23] proposed a weighted voting scheme in ensemble learning, but their proposed method only solved online concept drift problems. Wang *et al.* [24] incorporated the concept of ensemble learning with multiple classifiers to address concept drifts, and their results showed that the proposed ensemble learning method performs better than the method using only one classifier in addressing concept drifts. Antwi *et al.* [25] proposed an algorithm which adopts the cosine similarity to compare whether two datasets belong to two different concepts. Wang *et al.* [26] proposed a method which detects faults through evaluating forecast error rates. Wang and Abraham [6] proposed a method called LFR to calculate changes of TP, TN, FP, and FN ratios in the confusion matrix to detect faults. Lin *et al.* [27] proposed a multi-classifier DDD based on the MapReduce framework, which adopts multiple classifiers and a dynamic adjustment scheme to construct an ensemble learning model for adaption to concept drifts.

D. ENSEMBLE LEARNING FOR IMBALANCE DATA

With continuous advances in manufacturing technologies, machine components become increasingly precise. Hence, the amount of fault or abnormal data of machine components (i.e., minority class) is relatively much less than that of normal data (i.e., majority class) in long-run observation. Such an imbalance data problem has existed in a lot of real-world cases, e.g., credit card frauds, disease diagnosis, risk management, and fault detection in manufacturing productions. Most cases consider categorizing data into multiple classes. In a binary classification problem, data is divided into majority and minority classes, e.g., the probability that a machine manufactures a detective product could be less than 0.001%; the patients with a certain disease accounts for only 0.1% of healthy people. From these instances, the data from the minority class is generally important than that from the majority class.

Recently, imbalance data problems have received much attention, e.g., Ho *et al.* [28] and Rokach [29] emphasized that the methods based on only a single classifier cannot obtain precise results in addressing the data with multiple classes and much noise, and hence, they adopted ensemble learning methods with multiple classifiers to address imbalance data; Brown *et al.* [30] integrated multiple learning methods to increase the overall performance.

Recently, ensemble learning algorithms based on AdaBoost (Adaptive Boosting) have attracted a lot of attention. Freund and Schapire [16] proposed an ensemble learning method based on AdaBoost to reduce the forecast error. Wang *et al.* [31] incorporated the AdaBoost method with negative correlation learning to establish a novel AdaBoost.NC forecast model, which performs better than pervious methods in addressing classification problems. Wang and Yao [5]

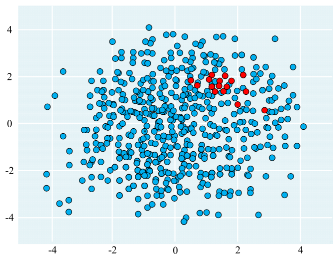


FIGURE 1. Illustration of imbalance data.

established a Dynamic AdaBoost.NC forecast model, which adds a method of automatically adjusting the training parameters to the AdaBoost.NC method, to effectively reduce the training time and increase the overall performance. The latter two methods proposed in [31], [5] improved the AdaBoost method to address data imbalance problems.

E. ENSEMBLE LEARNING FOR CONCEPT DRIFTS AND IMBALANCE DATA

To address both concept drifts and imbalance data, Ditzler and Polikar [32] proposed a novel method called Learn++.SMOTE, in which the Learn++.NSE method addresses concept drifts, and the SMOTE method addresses data imbalance.

III. PROPOSED ENSEMBLE LEARNING METHOD

This section first gives the framework of the proposed DAMSID method, consisting with three stages: ensemble learning, concept drift detection, and concept drift adaption. Then, the DAMSID algorithm is detailed.

A. DAMSID FRAMEWORK

The DAMSID framework is based on the DDD [2] consisting of three stages: ensemble learning, drift detection, and drift adaption. In the DAMSID, Stage 1 adopts the Dynamic AdaBoost.NC ensemble learning method incorporated with the SMOTE method to address imbalance data; Stage 2 adopts the LFR to detect drifts; and Stage 3 adopts the Dynamic AdaBoost.NC ensemble learning method to create a new model to adapt to the detected concept drift. The three stages in the DAMSID are detailed as follows:

1) STAGE 1: ENSEMBLE LEARNING BASED ON SMOTE AND DYNAMIC ADABOOST.NC

In imbalance data, the amounts of data points between different classes have remarkable differences. For example, Fig. 1 shows two classes of imbalance data, in which the amount of blue data points (majority class) is much more than that of red data points (minority class).

The proposed ensemble learning framework is illustrated in Fig. 2, which is based on the Dynamic AdaBoost.NC method that trains a sequence of weak classifiers and weight updates. At the beginning, the initial training dataset is either the initial dataset or the dataset collected after a concept drift.

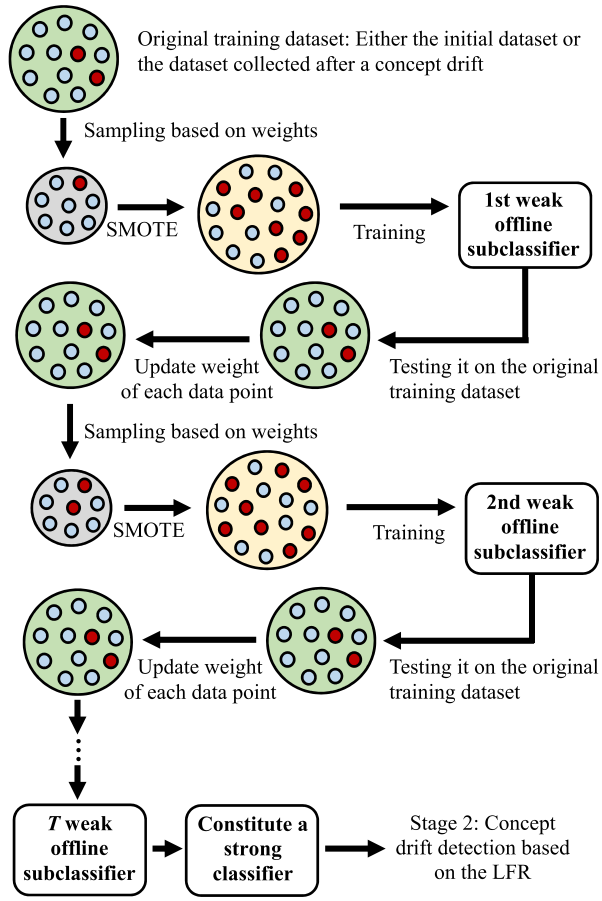


FIGURE 2. Flowchart of the proposed ensemble learning framework consisting of SMOTE and Dynamic AdaBoost.NC.

If the initial training dataset is the initial dataset, each data point in the training dataset is assigned to an equal weight initially, and then a smaller dataset is randomly selected from the training dataset; otherwise (i.e., the initial training dataset is the dataset collected after a concept drift), a smaller dataset consists of the data points with larger weights. Then, the SMOTE method is adopted to oversample the data points in the minority class (i.e., the red data points in Fig. 2). Then, the new dataset is adopted to train the 1st weak offline subclassifier. Then, we test whether this weak offline subclassifier performs accurately on the original training dataset, and use this forest result to update weight of each data point. Repeat the same procedure until T weak offline subclassifiers are trained. Finally, the T weak offline subclassifiers constitute a strong classifier whose output is a weighted sum of outputs of the T weak classifiers.

In what follows, the SMOTE method and the Dynamic AdaBoost.NC method are detailed, respectively.

In the application of detecting faults of machine components in manufacturing, the minority class is more important than the majority class, and it would be perfect if all the data points in the minority class are detected correctly. Before training the ensemble model, the SMOTE method (see Fig. 3)

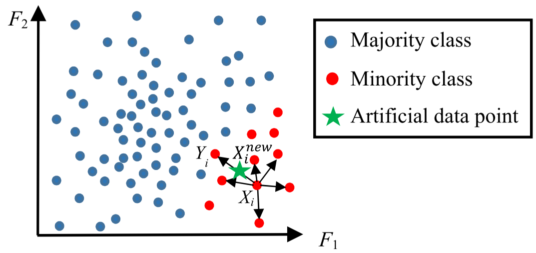


FIGURE 3. Illustration of the SMOTE method, in which F_1 and F_2 are two features of each data point.

is employed to oversample the data points in the minority class.

Key steps of the SMOTE method is as follows:

Step 1: Randomly select a data point X_i from the minority class.

Step 2: Calculate the distance between X_i and each of the other data points in the minority class. Select k data points in the minority class that are the closest to X_i.

Step 3: Randomly select one of these k data points, say Y_i.

Step 4: Generate an artificial data point x_i^{new} at a random location on the line segment between X_i and Y_i.

Given a dataset, the Dynamic AdaBoost.NC employs a "sequential learning" method to sequentially train a number of subclassifiers, in which each data point has a "weight" to represent the degree of the attention taken to the later learning and a "penalty" to record the degree of forecast mistakes. Detailed steps of the proposed Dynamic AdaBoost.NC are as follows:

1. Given a dataset $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$ consisting of m data points, in which the ith data point has a feature vector x_i and a class label y_i, we initialize its weight $D_1(x_i) = 1/m$, penalty $p_1(x_i) = 1$, and penalty strength λ to be a given parameter (set to 9 in [5]). That is, the initial weight, penalty, and λ value of each data point are equal.
2. Employ this dataset to sequentially train T subclassifiers as follows. Consider the iteration number $t = 1, 2, \ldots, T$.
   a) Based on the weight distribution D_t to train a weak subclassifier f_t: $X \rightarrow R$, in which $R = \{1, -1\}$ (which represents positive and negative outcomes, respectively).
   b) Calculate the penalty $p_t(x_i)$ of each data point x_i as follows:

$$p_t(x_i) = 1 - |amb_t(x_i)| \tag{1}$$

   where $amb_t(x_i)$ is calculated as follows:

$$amb_t(x_i) = \frac{1}{2t} \sum\nolimits_{j=1}^{t} (H_0 - f_j) \tag{2}$$

   where H_0 is the original strong classifier.
   c) Calculate the weight α_t of subclassifier f_t as follows:

$$\alpha_t = \frac{1}{2} \log \frac{\sum_{i, y_i = h_t(x_i)} D_t(x_i)(p_t(x_i))^{\lambda}}{\sum_{i, y_i \neq h_t(x_i)} D_t(x_i)(p_t(x_i))^{\lambda}} \tag{3}$$

   where $h_t(x_i) = 1$ if the forecast result of x_i is correct; otherwise, it is -1.
   d) If $Acc(f_t) \geq Acc(f_{t-1})$, then $\lambda = \lambda + 1$; otherwise, $\lambda = \lambda - -1$. That is, we check whether the accuracy at this iteration is better than that at the previous iteration. If yes, λ increases by 1; otherwise, it decreases by 1. The Acc (f_t) value is evaluated as follows:

$$1 - \sqrt{((0 - PF)^2 + (1 - PD)^2)/2} \tag{4}$$

   The above formula is explained as follows. First, calculate PF (Probability of False Alarm) and PD (Probability of Detection), and then test whether (PF, PD) is close to (0, 1) in terms of Euclidean distance. If yes, it means that the accuracy is higher. To the extreme, $PF = 0$ and $PD = 1$ imply the perfect accuracy [33].
   e) Update weight $D_t(x_i)$ of each data point x_i. Then, calculate new weight $D_{t+1}(x_i)$ of each data point x_i as follows:

$$D_{t+1}(x_i) = \frac{(p_t(x_i))^{\lambda} D_t(x_i) \exp(-\alpha_t f_t(x_i) y_i)}{Z_t} \tag{5}$$

   where Z_t is a normalization factor so that the total sum of all $D_{t+1}(x_i)$ is equal to 1.

After T subclassifiers $f_1, f_2, \ldots, f_T$ are obtained, construct a strong classifier H whose final forecast result is calculated according to the following ensemble of T weak subclassifiers:

$$H(x) = \text{sign}(\sum\nolimits_{t=1}^{T} \alpha_t f_t(x)) \tag{6}$$

2) STAGE 2: CONCEPT DRIFT DETECTION BASED ON THE LFR

After a strong classifier is trained at Stage 1, each testing data point is tested by this strong classifier, and the forecast result is obtained. This work supposes that the real label of each testing data point has been known. Hence, Stage 2 checks whether the forecast label and the real label are matched, to further detect whether a concept drift occurs. Stage 2 is based on the LFR, which has been shown to have outperformance in addressing imbalance data problems [6].

The LFR considers the four rates in the confusion matrix: TP (true positive) and FP (false positives) record the numbers of all positives that obtain positive and negative test outcomes, respectively; and TN (true negative) and FN (false negative) record the numbers of all negatives that obtain negative and positive test outcomes, respectively. Based on the four numbers, the four rates are calculated as follows:

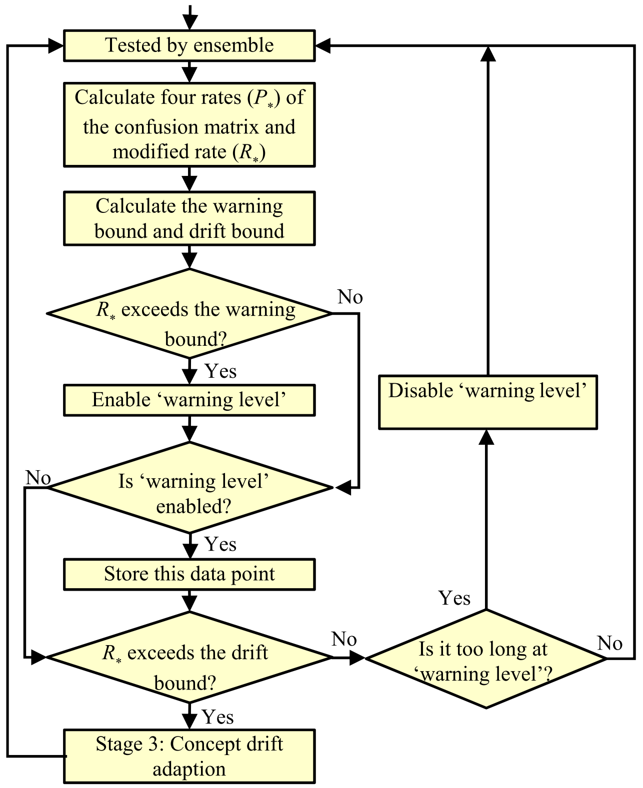


FIGURE 4. Flowchart of concept drift detection based on the LFR.

P_{tpr} = TP/(TP + FN); P_{tnr} = TN/(TN + FP); P_{ppv} = TP/(FP + TP); P_{npv} = TN/(TN + FN).

In theory, if there were no false test outcomes, each of the four rates would be 1. That is, if the data is stable and no concept drift occurs, each probability approaches to 1; otherwise, it starts to be less than 1. The flowchart of the LFR is given in Fig. 4.

The details of Fig 4 are explained as follows. After a data point is tested by the ensemble model trained at Stage 1, we have the test outcomes and the real label. Hence, if we let $*$ denote any of $\{tpr, tnr, ppv, npv\}$, the four rates P_* and the modified rate R_* are calculated. Then, the *warning bound* and the *drift bound* are calculated. Details of calculating the four rates P_*, the four modified rate R_*, *warning bound*, and the *drift bound* are referred to [6]. If R_* exceeds the warning bound, then a flag called the 'warning level' is enabled. Then, check if 'warning level' is enabled. If true, store the data point. That is, once we enter the 'warning level', the data points after this level are stored for training a new classifier. Then, check if R_* exceeds the drift bound. If true, go to Stage 3; otherwise, check if it is too long at the 'warning level'. If true, the 'warning level' is disabled

3) STAGE 3: DRIFT ADAPTATION BASED ON DYNAMIC ADABOOST.NC

After a concept drift occurs, it implies that the strong classifier trained at Stage 1 performs worse. Remind that the Dynamic AdaBoost.NC trains a strong classifier consisting of

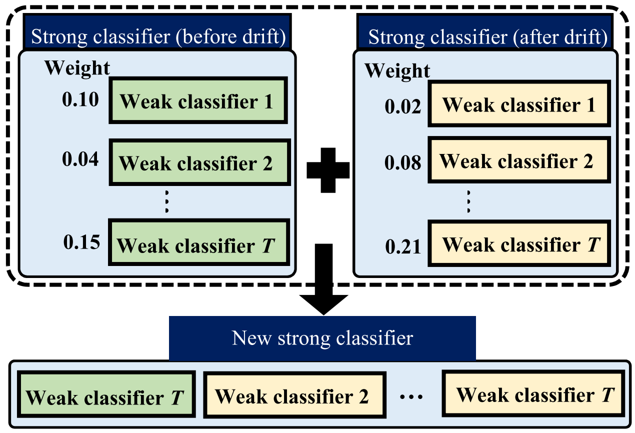


FIGURE 5. Illustration of generating a new strong classifier at Stage 3.

T weak subclassifiers $f_1, f_2, \ldots, f_T$ in which each subclassifier f_i is assigned to a weight α_i, which represents the accuracy of the subclassifier. Therefore, this work establishes a new strong classifier by combining the strong classifier trained at Stage 1 (i.e., before the drift) and the strong classifier trained by the data points stored from the warning level to the drift level (i.e., after the drift). As shown in Fig. 5, this new strong classifier consists of one weak subclassifier from the strong classifier at Stage 1 and $(T - 1)$ weak subclassifiers from the strong classifier trained after the drift. The new strong classifier is used to test the later data points.

The differences of the proposed DAMSID method from previous works are detailed as follows:

- Different from the conventional Dynamic AdaBoost.NC methods, Stage 1 of the proposed DAMSID extends the Dynamic AdaBoost.NC with data weights and integration with SMOTE to cope with data imbalance.
- Most works on concept drift adaption were to retrain the forecast model from scratch. Stage 3 of the proposed DAMSID has a new dataset adjustment for training the forest model to adapt to concept drifts.

B. DAMSID ENSEMBLE LEARNING ALGORITHM

The DAMSID ensemble learning algorithm is shown in Algorithm 1, which is detailed as follows. Let *mode* be a flag variable to record whether a warning level or a drift level has occurred. The possible values of flag *mode* are 1 (before warning level), 2 (after warning level), and 3 (after drift level). Let *afterDriftData* denote the set of data points collected after warning level. First, flag *mode* is initialized as 1, and the set *afterDriftData* is initialized to be empty (Line 1). Then, we take an amount of data to train an initial ensemble model, and let h be referred to the model (Line 2).

Then, the while loop in Lines 3 – 18 iteratively considers each data point in the data stream D. Let d denote the next data point from D (Line 4). If flag *mode* is 1 (i.e., before warning level), then we input data point d to the model h to obtain the predict result denoted by *prediction* (Line 6). Then, we use the

Algorithm 1 DAMSID

```
Inputs:
  Initial ensemble learning model: initial_ensemble
  Offline ensemble learning: ensemble
  LFR drift detection method: DetectDrift
  Combine old and new ensembles in adaption:
  ensemble_combine
  Data stream: D
1:  mode ← 1 and afterDriftData ← Ø
2:  h ← initial_ensemble   //Stage 1
3:  while D is not empty do
4:    d ← next data point from D
5:    if mode = 1 then
6:      prediction ← h(d)
7:    end if
8:    mode ← DetectDrift(d, prediction)   // Stage 2
9:    if mode is not equal to 1 then
10:     afterDriftData = afterDriftData ∪ {d}
11:   end if
12:   if mode = 3 then
13:     new_ensemble ← ensemble(afterDriftData) // Stage 1
14:     h ← ensemble_combine(h, new_ensemble) // Stage 3
15:     mode ← 1 and afterDriftData ← Ø
16:   end if
17:   output d, prediction
18: end while
```

LFR drift detection method (Stage 2) taking d and *prediction* as the input to update flag *mode* (Line 8).

Then, if *mode* is not equal to 1 (meaning that the data point d is a point after warning level), then d is included to the set *afterDriftData* (Line 10). Furthermore, if *mode* is equal to 3 (i.e., after drift level), then we use the data set *afterDriftData* to train an ensemble model by Stage 1, and this new ensemble model is denoted by *new_ensemble* (Line 13). Then, we use Stage 3 to combine the original ensemble model h and the new ensemble model *new_ensemble*, and this combined model is set as h (Line 14). Then, reset *mode* and *afterDriftData* to 1 and empty set, respectively (Line 15).

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

This section introduces implementation of the proposed DAMSID, and evaluates performance of the DAMSID. This work further implements the ensemble learning algorithm of the DAMSID with three layers of classifiers: super-strong, strong, and weak classifiers, in which each higher-layer classifier is an ensemble of lower-layer classifiers, i.e., the forecast result of the higher-layer classifier is obtained by a weighted voting sum of the results of lower-layer classifiers. In the experiments, the lowest-layer classifiers apply three types of classifiers: linear discriminant analysis (LDA), naïve Bayes (NB), and support vector machine (SVM).

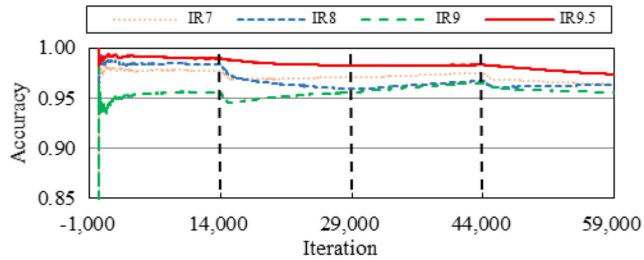


FIGURE 6. Results using the DAMSID with SMOTE on four datasets.

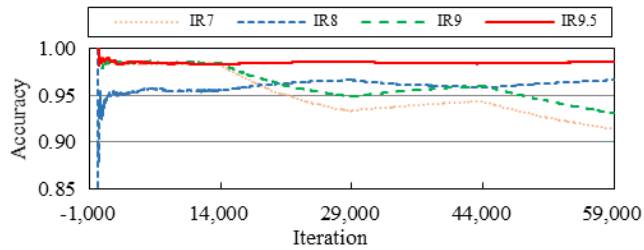


FIGURE 7. Results using the DAMSID without SMOTE on four datasets.

A. EXPERIMENTAL DATA

The experimental dataset is generated based on the method of generating the SEA dataset [34]. The dataset has 60,000 data points, each of which has three attributes and one real class label. The data is a time series. Three concept drifts occur at the data points of 15,000, 30,000, and 45,000, respectively. This work is referred to [24] to generate the dataset with various levels of imbalance ratios (i.e., the ratios of the minority class over majority class): 30%, 20%, 10%, and 5%, and the corresponding datasets are denoted by IR7, IR8, IR9, and IR9.5, respectively.

Among the 60,000 data points of a dataset, the proposed DAMSID uses the first 1,000 data points to train the initial strong classifier, and then uses this strong classifier to test the remaining 59,000 data points, during which the classifier is adapted if a concept drift is detected.

B. ANALYZING THE DAMSID WITH AND WITHOUT SMOTE

The results using the proposed DAMSID with and without SMOTE are shown in Figs. 6 and 7, respectively, in which the vertical axis represents the overall accuracy (i.e., the rate of the total correct outcomes over the total number of data points considered so far); the horizontal axis represents the number of data points considered; data points from $-1,000$ to 0 are the initial training dataset; concept drifts occur at points 14,000, 29,000, and 44,000. From Figs. 6 and 7, the accuracy of the results with SMOTE increases for the IR7 and IR9 datasets, but decreases a bit for the IR8 and IR9 datasets.

The confusion matrices of the results using the DAMSID with and without SMOTE on the first 14,000 testing data points (i.e., those before the first concept drift) of four datasets are shown in Table 1.

TABLE 1. Comparison of the confusion matrices of the results using the DAMSID with and without SMOTE on four datasets with different imbalance ratios.

	Without SMOTE			With SMOTE		
	Predict result	True label		Predict result	True label	
		1	–1		1	–1
IR7	1	360	0	1	409	0
	–1	336	13304	–1	287	13304
IR8	1	2289	0	1	2625	**67**
	–1	526	11185	–1	190	11118
IR9	1	876	0	1	1245	**32**
	–1	516	12607	–1	148	12575
IR9.5	1	360	0	1	409	0
	–1	336	13304	–1	287	13304

TABLE 2. Statistics of detecting concept drifts running 70 times of the DAMSID on the IR7 and IR9.5 datasets.

Dataset	Correct detections	Fault detections	Total detections
IR7	180	350	530
IR9.5	179	166	345

In addition to the overall accuracy, this work is more concerned about the accuracy of testing the data points in the minority class (i.e., those with label '1' in Table 1). From Table 1, all the results with SMOTE have a better accuracy in the minority class, and remarkably reduce the number of false negatives. On the other hand, the results with SMOTE have false positives. The reason is that the SMOTE creates artificial data points of the minority class, so that it increases the ability of forecasting the minority class, but decreases the ability of forecasting the majority class. Therefore, it is concluded that the SMOTE can effectively assist the DAMSID in increasing the accuracy of forecasting the minority class.

C. ANALYZING THE LFR IN THE DAMSID

This subsection analyzes the effect of the LFR in the DAMSID. Because the LFR has been shown to perform well in addressing imbalance data, this subsection analyzes the LFR in the datasets with two extreme degrees of data imbalance: IR7 and IR9.5. Hence, we run 70 times of the DAMSID on the IR7 and IR9.5 datasets, and record the number of detecting concept drifts for each 1,000 data points in the 70 times of ruining the DAMSID, as shown in Figs. 8 and 9, in which the height of each bar represents the number of detections for each 1,000 data points. The statistics of correct and fault detections are given in Table 2. Because the DAMSID collects 1,000 data points for later training before entering the drift level, the correct detections should occur during the 1,000 data points after drifts, i.e., 14,000–16,000, 29,000–31,000, and 44,000–46,000. Hence, the two bars responded to these ranges are shaded in Figs. 8 and 9.

From Figs. 8 and 9, the length of shaded bars is relatively longer, i.e., concept drifts have a high probability to be detected. From Table 2, the number of fault detections on the IR7 dataset is double that on the IR9.5 dataset. It is speculated that the LFR performs better in the datasets with a higher imbalance degree.

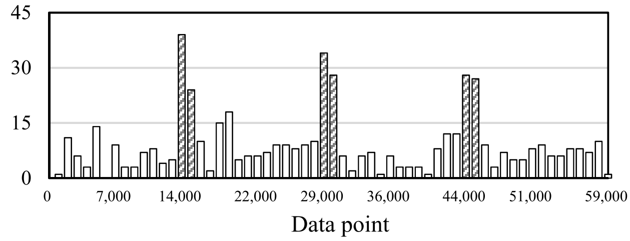


FIGURE 8. The frequency of detecting concept drifts running 70 times of the DAMSID on the IR7 dataset.

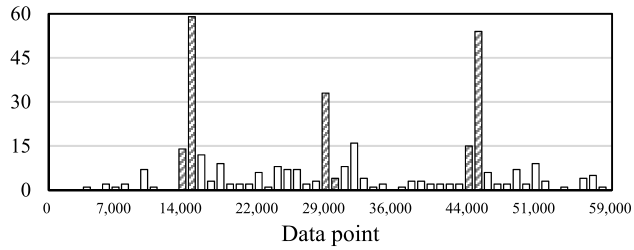


FIGURE 9. The frequency of detecting concept drifts running 70 times of the DAMSID on the IR9.5 dataset.

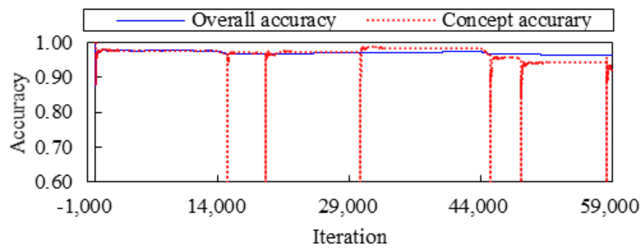


FIGURE 10. Results of overall and concept accuracies using the DAMSID on the IR7 dataset.

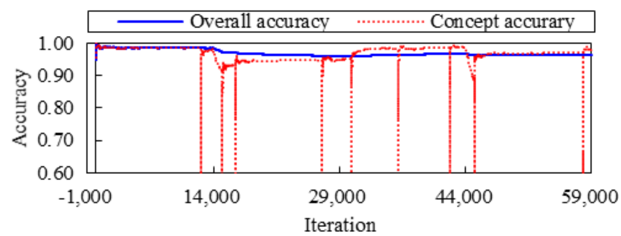


FIGURE 11. Results of overall and concept accuracies using the DAMSID on the IR8 dataset.

D. ANALYZING THE RESULTS USING THE DAMSID

In addition to the overall accuracy, another important measure is the *concept accuracy*, which is the rate of the total correct outcomes over the total number of data points considered so far after a concept drift. The results of overall and concept accuracies using the DAMSID on the four datasets are shown in Figs. 10–13, in which the overall accuracies of the results for the IR7, IR8, IR9, and IR9.5 datasets show outperformance (96.01%, 96.3%, 95.6%, and 97.02%, respectively). All the results show that all concept drifts can be detected within 1,000 data points after drifts.

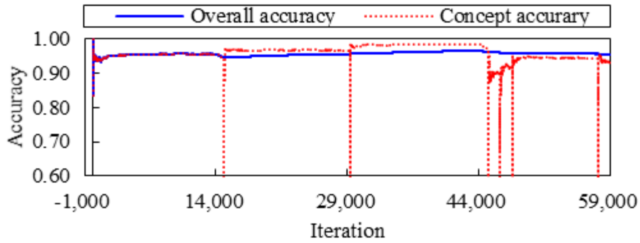


FIGURE 12. Results of overall and concept accuracies using the DAMSID on the IR9 dataset.

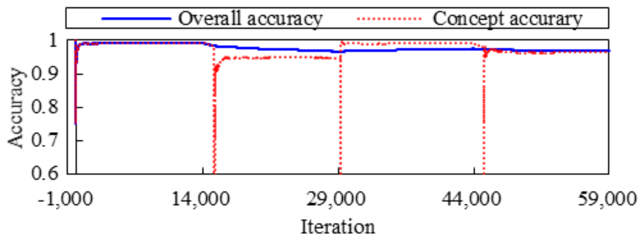


FIGURE 13. Results of overall and concept accuracies using the DAMSID on the IR9.5 dataset.

TABLE 3. The confusion matrices of the results using the DAMSID on four concepts of the IR7 dataset.

		Real label							
		Concept 1		Concept 2		Concept 3		Concept 4	
		1	−1	1	−1	1	−1	1	−1
Predict	1	4174	294	4494	516	4313	80	4492	1058
	−1	24	9508	6	9984	187	10420	8	9442

TABLE 4. The confusion matrices of the results using the DAMSID on four concepts of the IR8 dataset.

		Real label							
		Concept 1		Concept 2		Concept 3		Concept 4	
		1	−1	1	−1	1	−1	1	−1
Predict	1	2768	183	2992	940	2872	145	2878	603
	−1	47	11002	8	11060	128	11855	122	11397

TABLE 5. The confusion matrices of the results using the DAMSID on four concepts of the IR9 dataset.

		Real label							
		Concept 1		Concept 2		Concept 3		Concept 4	
		1	−1	1	−1	1	−1	1	−1
Predict	1	1381	608	1493	669	1291	25	1500	1114
	−1	12	11999	7	12831	209	13475	0	12386

TABLE 6. The confusion matrices of the results using the DAMSID on four concepts of the IR9.5 dataset.

		Real label							
		Concept 1		Concept 2		Concept 3		Concept 4	
		1	−1	1	−1	1	−1	1	−1
Predict	1	655	77	655	847	642	23	731	640
	−1	41	13227	9	13407	108	14227	19	13610

To realize the outcomes of minority and majority classes for each concept, we analyze the confusion matrices of the results using the DAMSID on four concepts of the four datasets shown in Tables 3–6, in which the TPRs for the IR7, IR8, IR9, and IR9.5 datasets are 98%, 97%, 99.1%, and 94%, respectively. Hence, the DAMSID is shown to perform well in forecasting minority-class data in imbalance data problems. In addition, the TPR decreases as the imbalance rate increases. The reason is speculated that the amount of minority-class data (i.e., label '1') decreases.

V. CONCLUSION

With development of the IIoT, deployment of a large-scale number of sensors in manufacturing industries can continuously collect machine conditions (which are big data). CBM analyzes the machine conditions to predict the time point when the machine starts to perform abnormally and to replace or maintain it in advance. Because most classifiers can be trained in an offline way, this work has proposed a DAMSID ensemble learning algorithm based on offline classifiers to address the CBM with concept drifts and imbalance data. The DAMSID improves the concept detection method by the Dynamic AdaBoost.NC with SMOTE method, improves the concept drift detection method by LFR, and includes a novel drift adaption method. Experimental results show that the proposed DAMSID can successfully detect all concept drifts; the accuracy rate of the forecast results using the DAMSID can achieve over 90%; the overall accuracy rate for the extreme imbalance data (IR9.5 dataset) can arrive at 97.02%; the accuracy rate for the minority-class data can achieve over 94%.

From experimental trials, the performance of experimental outcomes significantly depends on the data sampling. Therefore, a line of future work is to propose a novel data sampling in the DAMSID. It is also of interest to improve the computing efficiency of the algorithm. In addition, it is interesting to design a robust ensemble learning method to adapt to various degrees of data imbalance. And, in practice, unlabeled or semilabeled data is common, and hence it is of crucial to investigate the classification for these data types.

REFERENCES

[1] F. Tao, J. Cheng, and Q. Qi, "IIHub: An industrial Internet-of-Things hub toward smart manufacturing based on cyber-physical system," *IEEE Trans. Ind. Inform.*, vol. 14, no. 5, pp. 2271–2280, May 2018.

[2] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 619–633, Apr. 2012.

[3] B. Mirza, Z. Lin, and N. Liu, "Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift," *Neurocomputing*, vol. 149, pp. 316–329, Feb. 2015.

[4] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[5] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Trans. Rel.*, vol. 62, no. 2, pp. 434–443, Jun. 2013.

[6] H. Wang and Z. Abraham, "Concept drift detection for streaming data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–9.

[7] Y. H. O. Chang and B. M. Ayyub, "Fuzzy regression methods—A comparative assessment," *Fuzzy Sets Syst.*, vol. 119, no. 2, pp. 187–203, 2001.

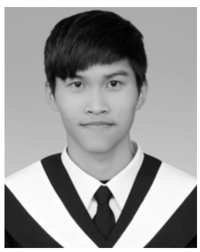
[8] H. Niu and J. Wang, "Quantifying complexity of financial short-term time series by composite multiscale entropy measure," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 22, nos. 1–3, pp. 375–382, 2015.

[9] K. Nishimura, Y. Maehata, and W. Sunayama, "Improved inspection of facilities for high-voltage class using data mining," *Elect. Eng. Jpn.*, vol. 191, no. 2, pp. 1165–1170, 2015.

[10] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, 2006.
[11] W. Caesarendra, A. Widodo, and B.-S. Yang, "Application of relevance vector machine and logistic regression for machine degradation assessment," *Mech. Syst. Signal Process.*, vol. 24, no. 4, pp. 1161–1171, 2010.
[12] R. K. Patel and V. K. Giri, "Feature selection and classification of mechanical fault of an induction motor using random forest classifier," *Perspect. Sci.*, vol. 8, pp. 334–337, Sep. 2016.
[13] C. Lin, D. Deng, J. Kang, S. Chang, and C. Chueh, "Forecasting rare faults of critical components in LED epitaxy plants using a hybrid grey forecasting and harmony search approach," *IEEE Trans. Ind. Inform.*, vol. 12, no. 6, pp. 2228–2235, Dec. 2016.
[14] J. Wan *et al.*, "A manufacturing big data solution for active preventive maintenance," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2039–2047, Aug. 2017.
[15] R. Polikar, "Ensemble learning," in *Ensemble Mach. Learn.*, New York, NY, USA: Springer, 2012, pp. 1–34.
[16] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, Jul. 1996, pp. 148–156.
[17] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proc. IEEE*, vol. 67, no. 5, pp. 708–713, May 1979.
[18] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.
[19] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.
[20] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979.
[21] 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.
[22] 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.
[23] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: A new ensemble method for tracking concept drift," in *Proc. 3rd IEEE Int. Conf. Data Mining*, Jul. 2003, pp. 1–10.
[24] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Dec. 2003, pp. 226–235.
[25] D. K. Antwi, H. L. Viktor, and N. Japkowicz, "The PerfSim algorithm for concept drift detection in imbalanced data," in *Proc. IEEE 12th Int. Conf. Data Mining Workshops*, Jul. 2012, pp. 1–6.
[26] S. Wang, L. L. Minku, D. Ghezzi, D. Caltabiano, P. Tino, and X. Yao, "Concept drift detection for online class imbalance learning," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–10.
[27] C. Lin, L. Shu, D. Deng, T. Yeh, Y. Chen, and H. Hsieh, "A MapReduce-based ensemble learning method with multiple classifier types and diversity for condition-based maintenance with concept drifts," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 38–48, Sep. 2017.
[28] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, Jan. 1994.
[29] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.
[30] G. Brown, J. L. Wyatt, and P. Tiňo, "Managing diversity in regression ensembles," *J. Mach. Learn. Res.*, vol. 6, pp. 1621–1650, Dec. 2005.
[31] S. Wang, H. Chen, and X. Yao, "Negative correlation learning for classification ensembles," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2010, pp. 15–25.
[32] G. Ditzler and R. Polikar, "An ensemble based incremental learning framework for concept drift and class imbalance," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2010, pp. 1258–1268.
[33] T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 2–13, Jan. 2007.
[34] W. N. Street and Y. S. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2001, pp. 377–382.

CHUN-CHENG LIN (S'06–M'08–SM'17) received the B.S. degree in mathematics, the M.B.A. degree in business administration, and the Ph.D. degree in electrical engineering from National Taiwan University, in 2000, 2007, and 2009, respectively. He has been a Professor of industrial engineering and management, National Chiao Tung University, since 2016. Since 2017, he has been the Associate Dean of the College of Management, National Chiao Tung University, which he joined in 2011. He was an Assistant Professor of computer science with the University of Taipei (2010–2011) and the National Kaohsiung University of Science and Technology (2009–2010). His main research interests include wireless networks, information visualization, design and analysis of algorithms, as well as computational management science.

DER-JIUNN DENG (M'10) received the Ph.D. degree in electrical engineering from the National Taiwan University, in 2005. He joined the Department of Computer Science and Information Engineering, National Changhua University of Education, as an Assistant Professor, in 2005, and then became a Distinguished Professor, in 2016. In 2018, he was seconded to the Overseas Chinese University as the Dean of the Research and Development for a period of two years. His research interests include multimedia communication, quality-of-service, and wireless local networks. He has received a number of research awards, such as the Research Excellency Award of National Changhua University of Education, the Outstanding Faculty Research Award of the National Changhua University of Education, the ICS 2014 Best Paper award, the NCS 2017 Best Paper Award, and the Chinacom 2017 Best Paper Award. He also served or is serving on several program chairs, symposium chairs, and technical program committees for the IEEE and other international conferences. He is the Co-Editor-in-Chief of the *EAI Endorsed Transactions on Internet of Things* and the *Journal of Computers*. He serves as an Associate Editor for the *IEEE Network Magazine* and the *International Journal of Communication Systems*.



CHIN-HUNG KUO received the B.S. degree in industrial engineering and management from Yuan Ze University, Taiwan, in 2017, and the M.S. degree in industrial engineering and management from National Chiao Tung University, Taiwan, in 2019. His main research interests include data science, operations research, production management, as well as metaheuristics.

LINNAN CHEN received the B.S. degree in industrial engineering and enterprise information from Tunghai University, Taiwan, in 2017, and the M.S. degree in industrial engineering and management from National Chiao Tung University, Taiwan, in 2019. His main research interests include wireless networks, data science, operations research, as well as metaheuristics.