Scientific Research Publishing

International Journal of Intelligence Science, 2018, 8, 43-74
http://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Using Hybrid and Diversity-Based Adaptive Ensemble Method for Binary Classification

Xing Fan, Chung-Horng Lung, Samuel A. Ajila

Department of Systems and Computer Engineering, Carleton University 1125 Colonel By Drive, Ottawa, Canada
Email: XingFan@cmail.carleton.ca, chlung@sce.carleton.ca, ajila@sce.carleton.ca

How to cite this paper: Fan, X., Lung, C.-H. and Ajila, S.A. (2018) Using Hybrid and Diversity-Based Adaptive Ensemble Method for Binary Classification. *International Journal of Intelligence Science*, **8**, 43-74.
https://doi.org/10.4236/ijis.2018.83003

Received: May 26, 2018
Accepted: July 27, 2018
Published: July 30, 2018

Abstract

This paper proposes an adaptive and diverse hybrid-based ensemble method to improve the performance of binary classification. The proposed method is a non-linear combination of base models and the application of adaptive selection of the most suitable model for each data instance. Ensemble method, an important machine learning technique uses multiple single models to construct a hybrid model. A hybrid model generally performs better compared to a single individual model. In a given dataset the application of diverse single models trained with different machine learning algorithms will have different capabilities in recognizing patterns in the given training sample. The proposed approach has been validated on Repeat Buyers Prediction dataset and Census Income Prediction dataset. The experiment results indicate up to 18.5% improvement on F_1 score for the Repeat Buyers dataset compared to the best individual model. This improvement also indicates that the proposed ensemble method has an exceptional ability of dealing with imbalanced datasets. In addition, the proposed method outperforms two other commonly used ensemble methods (Averaging and Stacking) in terms of improved F_1 score. Finally, our results produced a slightly higher AUC score of 0.718 compared to the previous result of AUC score of 0.712 in the Repeat Buyers competition. This roughly 1% increase AUC score in performance is significant considering a very big dataset such as Repeat Buyers.

Keywords

Bigdata Analytics, Machine Learning, Adaptive Ensemble Methods, Binary Classification

1. Introduction

In supervised learning techniques, ensemble learning method is the technique that uses multiple single models to construct a hybrid model in order to achieve

better performance compared to that of using a single model [1]. A general workflow for solving classification problems using ensemble methods consists of the following steps. Firstly, raw data usually need to be pre-processed for initial training dataset, during which feature extraction and normalization are applied. Secondly, training sets for each individual single model are derived from the initial training dataset. Thirdly, single models are trained from different training datasets or by different algorithms. Parameter tuning and validation for each individual model are included in this phase in the third step. Fourthly and finally, all the single models are combined to construct the ensemble. The final hybrid model is constructed based on the results of individual models and the model is further validated and tested.

There are a number of ensemble learning techniques proven to be successful in practice. Based on our understanding, most of the techniques could be generally categorized into two types according to their different concerns: *Type* I techniques focus on deriving new training sets from the initial training set to train multiple different single models. *Type* II techniques focus on finding ways to blend the individual models [1]. In this paper, we are interested in *Type* II techniques as a method for improving predictive performance in binary classification problems.

For a given dataset the application of diverse single models trained with different machine learning algorithms will have different capabilities in recognizing patterns in the given training sample. It is therefore very important to select those highly diverse single models and combine them properly using an efficient diversity principle in order to achieve best performance [2]. The most popular *Type* II ensemble techniques assign weights to all trained single models and then linearly combine them [3], [4]. Since, for all unknown instances, the contribution of each individual model to the final prediction is fixed, which may limit performance. It is therefore important to dynamically adjust each single model's contribution for different instances and this requires methods based on non-linear blending which is one of the goals of this research.

Moreover, the existing *Type* II ensemble methods take all the outputs of the combined single models into account, which means that the weaknesses of each of the single models are also kept in the hybrid model. If an ensemble method can adaptively select the most suitable single model to predict each instance, then the weaknesses of the single models can be avoided. In order to maximize the complementation effect, only those single models with the highest pairwise diversities should be selected to construct the ensemble model.

The objective of this research is therefore to find an ensemble model that can: 1) select base models based on their pairwise diversities; 2) recognize the best suitable base model for each data instance; and 3) predict each unknown instance using a suitable base model. If this objective is achieved, the overall performance of the predictions can be improved by using the new diverse hybrid ensemble. The contributions of this paper are as follows: 1) a novel ensemble method for solving binary classification problem is proposed; 2) the pairwise di-

versities of single classifiers are measured using two different methods and the results of these measurements are used to select and combine the base classifiers; 3) multiple machine learning algorithms are used to train different base models and a comprehensive framework is designed and implemented; and 4) the proposed ensemble method is validated using the Repeat Buyer Prediction Competition [5] and the Censure Income Prediction CIP [6] dataset from University of California machine learning repository.

Furthermore, compared to our exploratory paper on the subject matter [7], the research presented in this paper includes a thorough performance and complexity analysis and more study on the classifiers. In addition, we have added a second dataset, the Census Income Prediction. This dataset is less complex and we used it as a control for the more complex Repeat Buyers Prediction dataset.

The remainder of this paper is organized as follows: Section 2.0 discusses the background and related work. Section 3.0 describes the proposed ensemble method. Section 4.0 presents the analysis of the experimental results and finally, conclusion and possible directions for the future research are discussed in Section 5.0.

2. Background and Related Work

2.1. Classification

Machine learning is divided into two main types: supervise and unsupervised learning. Other types such as semi-supervise learning and reinforcement learning can be derived from the two main types [8]. The main objective of supervise learning is to train a model using a training data with labels so that the model can be used for predicting test (unknown) data. An unsupervised learning technique reveals patterns of an unlabeled data. Supervised learning problems fall into two categories—classification and regression. The two are similar except for the outputs of the trained models. The outputs of classification are discrete while that of regression is continuous. In this paper, the proposed ensemble method is for solving classification problems.

Classification problems can be categorized into two types based on the number of classes to be identified: binary classification and multiclass classification [8]. The former identifies instances as one of the two pre-defined classes, whereas the latter classifies instances into one of the more than two classes. The research work in this paper focuses on binary classification.

2.2. Feature Engineering, Binary Classifiers and Confusion Matrix

Feature engineering is a process of selecting a set of features that describes a set of problems and matching machine learning algorithms to solve those problems. Also, feature engineering includes data normalization since the original raw data may include redundant properties or noise, it is then necessary to tweak the data using normalization to achieve better performance. Additionally, some features of the raw data may impose extra workload on the training and testing which

could be very expensive in terms computational time and space. These features may be redundant just occupying memory space and more computation time with no benefit to the training or testing of the final model. It is therefore necessary to carefully perform feature analysis with the aim of removing or reducing the effect of redundant or noisy features.

Formally, a classifier is a derived function that maps instances to targets, of which all parameters are determined [9]. A machine learning algorithm is a process that estimates the parameters of the function by learning the train data, so that the classifier could fit the train data. A binary classifier is a function that maps instances to positive or negative class label [10]. According to the types of outcomes, classifiers could be recognized as a discrete classifier or probabilistic classifier. A discrete classifier directly generates a predicted class label for an instance whereas a probabilistic classifier assigns each instance a score to indicate its degree of confidence in belonging to one class [11]. Moreover, for the probabilistic classifiers, the assigned score could either be an absolute probability, or could be any uncalibrated real value with a higher value representing a higher probability. In other words, probabilistic classifiers rank instances according to their probabilities of being a member of a class in ascending or descending order. Afterwards, compared to the absolute predictions by discrete classifiers, a relative threshold is assigned to probabilistic classifiers to determine the predicted classes of instances. The instances with a score higher than the threshold is considered as positive, while those with a score lower than the threshold is classified as negative.

Confusion Matrix is generally used for presenting and interpreting the performance of a classifier on a dataset. It is an n × n matrix (n represents the number of classes) constructed from the Cartesian product of actual classes and predicted classes [12]. Each entry of the matrix indicates the number of instances in each ordered pair of the Cartesian product. For binary classification, n equals to 2, therefore represents the two classes of Positive and Negative. There are four outcomes for test instances in a classification [12]:

- **True Positive (TP):** the number of instances that have been correctly classified as Positive by the classifier;
- **False Positive (FP):** the number of instances that have been incorrectly classified as Positive by the classifier;
- **True Negative (TN):** the number of instances that have been correctly classified as Negative by the classifier;
- **False Negative (FN)**: the number of instances that have been incorrectly classified as Negative by the classifier.

Based on these four outcomes, the following rates and measures are defined:

- **True Positive Rate (TPR):** also called Sensitivity or Recall is the proportion of real positive instances that are correctly identified;
- **True Negative Rate (TNR):** also called Specificity is the proportion of real negative instances that are correctly identified;

- **False Positive Rate (FPR):** also called Fall-out is the proportion of real negative instances that are incorrectly identified, can be calculated from (1—Specificity) as well;
- **False Negative Rate (FNR):** also called Miss Rate is the proportion of real positive instances that are incorrectly identified;
- **Positive Predictive Value (PPV):** also called Precision is the proportion of predicted positive instances that are correctly identified;
- **Accuracy:** The proportion of all instances that are correctly identified.

Now, in this paper, the accuracy, precision and recall for the diverse hybrid ensemble model are defined as:

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

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

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

Among these terminologies, Precision (PPV) and Recall (TPR) are the two universal metrics describing the performance of a classifier on a dataset. However, it is worth mentioning that the accuracy may not be an adequate measure of performance because it is very sensitive to class distribution of the given dataset [13]. In the case of imbalance dataset, for example, if 99% of the instances are negative and only 1% is positive, a classifier could even achieve a high accuracy of 99% by simply identifying all instances as negative, which is meaningless in practice. Therefore, in this paper, Area Under the Receiver Operating Characteristic Curve (AUC) and F1 score are selected as the metrics to evaluate the performance of the proposed ensemble method.

2.3. Receiver Operating Characteristics (ROC), AUC, and F_1 Score

A ROC curve is a plot that visualizes the performance of a binary classifier. It is drawn as a 1 × 1 square area called ROC space, of which x-axis and y-axis are defined as False Positive Rate (*i.e.* FPR) and True Positive Rate (*i.e.* TPR), respectively [13]. Therefore, the plots in ROC space describe the trade-off between the benefit (TPR) and the cost (FPR) of a classifier.

For each discrete classifier on a given dataset, there is only one corresponding confusion matrix, which is plotted as a single point in ROC space [13]. On the other hand, probabilistic classifiers could yield different confusion matrixes as threshold varies, so that multiple points are plotted in ROC space [14].

A threshold represents the strictness of a probabilistic classifier making a positive prediction. A higher threshold could result in lower FPR with TPR sacrificed. On the contrary, lower thresholds could lead to both increased FPR and TPR from which the points plotted are closed to (1, 1) point in the ROC space. If more actual positive instances could be assigned relatively higher scores by the probabilistic classifier, then a higher TPR with a lower FPR could be achieved even

when a relatively higher threshold is set, so that the points plotted from higher thresholds are close to the (0, 1) point in ROC space. The ROC curve, in this case, is pulled up to the perfect classification point, which leads to a larger area under the ROC curve (AUC). AUC represents the probability that a classifier assigns a higher score to a randomly selected positive instance than a negative one [13] and a larger AUC indicates that a better performance is achieved by the classifier.

F_1 score [15] is a commonly used metric to evaluate the performance of a classifier. F_1 score for the diverse hybrid ensemble model is defined as the harmonic mean of precision and recall:

$$F_1 = \frac{2 * \text{Presicion} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * \text{PPV} * \text{TPR}}{\text{PPV} + \text{TPR}}$$

In this case, precision denotes the degree of confidence for a classifier to predict an instance as positive, while recall indicates the ability of a classifier to identify the positive instances. The best value of F_1 score is 1 and the worst is 0. F_1 score conveys the balance between precision and recall. For example, an F_1 score of 0.45 is calculated from a precision of 0.9 and a recall of 0.3, while another F_1 score of 0.5 is obtained with both precision and recall being 0.5. Therefore, for a classifier, a tradeoff between its precision and recall is beneficial for achieving a higher F_1 score. Moreover, the reasonable tradeoff can be accomplished by tuning the threshold, which is highly correlated with the class distribution of the given dataset [16]. For example, a threshold of 0.5 may be appropriate for the dataset with balanced class, while a higher threshold could be better if the proportion of positive instances is very small. F_1 score is selected as the performance evaluation metric in this paper because it is considered more suitable in practice compared to AUC [17] when a dataset is suffering from imbalanced class problem [16]. In addition, AUC is a "curve" metric indicating the general or average performance of a classifier, while the F_1 score is a "point" metric that could be more meaningful in practice [18]. Both AUC and F_1 score are compared in this work.

2.4. Classification Learning Algorithms

In classification, each model trained by an algorithm uses a formula for calculating the approximation of the probability or discrete class, which is defined as $\hat{y} = f(x)$. Therefore, a loss function (or cost function) $L(y, f(x))$ of the model is defined as a function that quantizes the cost for the deviation of predictions from the actual values, such as squared loss $L(y, f(x)) = \frac{1}{2}(y - f(x))^2$ [19]. Given a training data, if its loss function is convex and differentiable, then an optimization algorithm can be applied to find a set of parameters $f(x)$, so that the loss function is minimized on that given data. The convexity of the loss function could guarantee that the found minimum is the global one. This process of optimization is to accomplish parameter estimation, and different selection of loss functions could lead to different estimations of the parameters.

In this paper, the following factors are taken into account when selecting ma-

chine learning algorithm. 1) The algorithm should have the ability to learn a probabilistic classifier because of AUC and F_1 score. 2) The algorithm should be capable to learn sparse data since the given dataset is under high sparsity. 3) Since the main concern of this work is the ensemble method it is better for the selected algorithm to be simple so that the parameter tuning for the ensemble models could be easier. 4) The algorithm should achieve a proper balance between the performance of the single model and its training time as well as its resource requirement. So, with this in mind we have selected Logistic Regression under the general regression model (GRM), and for the ensemble, we have selected Boosting, Bagging, and Stacked Generalization.

Logistic regression, a generalization of linear regression (also called Generalized Linear Model GLM) can learn a model and it is a function mapping of an instance to a predicted class. The output of the function is a real value between 0 and 1 that indicates the probability of the instance being one of the binary classes. The instance could then be classified as positive if its predicted probability is larger than the pre-determined threshold [9]. Theoretically, the meta-level model could be learnt by any type of supervised machine learning algorithms, but logistic regression is selected to be the meta-level learner because of its simplicity and efficiency and therefore, all the base models are linearly combined. Furthermore, the GLM is capable of accepting dependent variables that have other distributions other than normal distributions for ordinary linear regression, such as Bernoulli distribution for logistic regression. It is done by mapping the linear regression model to the response variable through a link function and a quantizer. A standard logistic function is applied as the link function, which is defined as following:

$$P(t)=\frac{1}{1+\mathrm{e}^{-t}}$$

where t represents the ordinary linear regression model that is formulated as:

$$t=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+\beta$$

where β denote an unobservable bias from the actual value. The final logistic regression model is formulated as:

$$F(x)=P(y_i=1)=\frac{1}{1+\mathrm{e}^{-\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+\beta\right)}}$$

where $F(x)$ is interpreted as the probability of the i^{th} instance being positive.

Generally, logistic regression can find a hyperplane $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+\beta=0$ as the decision boundary that slices the train data into two parts: the points above the hyperplane denote the instances with higher probabilities of being positive than that of being negative, and the points under the hyperplane are considered as the instances more likely to be negative. Additionally, the points on the hyperplane represent the instances with equal probabilities of being positive and negative. Note that the hyperplane is precisely the quantizer.

The core concept of Bagging [17] is to train multiple base classifiers from dif-

ferent new training sets that are randomly sampled from the original train data with replacement, then make the final prediction for each instance by averaging the scores (for a probabilistic classifier) or majority voting (for a discrete classifier). In this paper, Random Forests provided by Scikit-learn is applied, which is a popular implementation of Bagging and uses decision tree as the base model.

Boosting [19], [20] is a family of ensemble learning algorithms that combines a set of weak learners to construct a relatively strong one. Compared to Bagging method, Boosting algorithms is intended to improve performance by reducing bias. Adaptive Boosting (AdaBoost) and Gradient Boosting Machine (GBM) are the two most popular Boosting algorithms [21]. In this paper, an implementation of AdaBoost provided by Scikit-learn and an implementation of GBM called Extreme Gradient Boosting (Xgboost) are used. Xgboost was initially released as an open source project by [11]. Both of the selected implementations of AdaBoost and GBM are probabilistic classifiers and able to deal with sparse data. Formally, an ensemble model trained by a boosting algorithm can be formulated as follows [11]:

$$F_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x) + \alpha_0$$

where T denotes the number of base models, $F_T(x)$ represents the final ensemble model, and $h_t(x)$ represents the hypothesis generated by the t^{th} base classifier. α_t is the weight assigned to the t^{th} base classifier and α_0 is a constant determined in the first stage for initializing the model. For gradient boosting, each additive base model is trained to fit the residual of the existing ensemble model, so that the final model becomes more accurate by taking the hypothesis generated by this newly added base classifier into account. The gradient boosting algorithm is a process to minimize the loss function of the final ensemble model without prior knowledge of the final loss function. This is achieved by minimizing the loss function of the existing ensemble model whenever a new base model is added. Therefore, with the accumulation of the base models, the loss of the implied final function declines gradually. In other words, with a training set $\{(x_i, y_i)\}_{i=1}^{n}$, a convex and differentiable loss function $L(y, F(x))$ and the number of iterations T, the underlying mathematics of this process can be defined as follows [11]:

1) Initialize the model with a constant:

$$F_0 = \alpha_0 = \arg\min_{\alpha} \sum_{i=1}^{n} L(y_i, \alpha)$$

2) For $t = 1$ to M:

a) Calculate the residuals as the negative gradient:

$$-\frac{\partial L(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)} = y_i - F_{t-1}(x_i) = r_{it}$$

b) Fit a new base model $h_t(x)$ with the new training set $\{(x_i, r_{it})\}_{i=1}^{n}$:

$$h_t(x) = \arg\min_h \sum_{i=1}^{n} L\left(r_{it}, h_t(x_i)\right)$$

c) Determine the multiplier α_t of the newly added base model $h_t(x)$:

$$\alpha_t = \arg\min_{\alpha} \sum_{i=1}^{n} L\left(y_i, F_{t-1}(x_i) + \alpha h_t(x_i)\right)$$

d) Update the existing ensemble model:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x)$$

e) Output the final ensemble model as follows:

$$F_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x) + \alpha_0$$

The bias of the model is reduced by incrementally training the base models to learn the errors of their predecessor ensemble models. However, the concept of correcting the errors made by the predecessor model may be overfitting-prone, especially for the relatively noisy data. Accordingly, several regularization techniques are introduced into the model for constraining the learning procedure, so that the final ensemble model is prevented from overfitting the training set. Consequently, Boosting methods can reduce the bias of model with keeping variance under control.

Stacked Generalization, which is also known as Stacking was first proposed by David H. Wolpert [22]. It is a high level ensemble method that introduces a meta-level model to combine a group of base models, which are usually of diverse types. In Stacking, each of the linear or non-linear models is selected as the meta-level model to combine a group of base models that are the same as those in our proposed method.

Figure 1 shows the process of Stacking, during which all base models are trained from the training set and tested on the testing set. Then, the meta model takes the outputs of all base models for testing data as new feature vectors and the corresponding actual class values in testing data as targets, so that the meta model can learn a combination of all base models to reduce the overall error. Based on our objectives, Stacking is a type of generalized and flexible method of ensemble, which could be easily applied together with other ensemble methods to further enhance the total performance.

2.5. Related Works

Ensemble methods such as Bagging, Boosting and Stacking which combine the decisions of multiple hypotheses are some of the strongest existing ensemble methods and therefore provide a reliable foundation for our proposed ensemble method. The work in [23] applied Random Forests to classification of hyperspectral data on the basis of a binary hierarchical multi-classifier system. The work in [24] presented an algorithm for musical style and artist prediction from an audio waveform by utilizing AdaBoost for the selection and aggregation of audio features. Gradient Boosting Machine (GBM) was used in [25] for incorporating

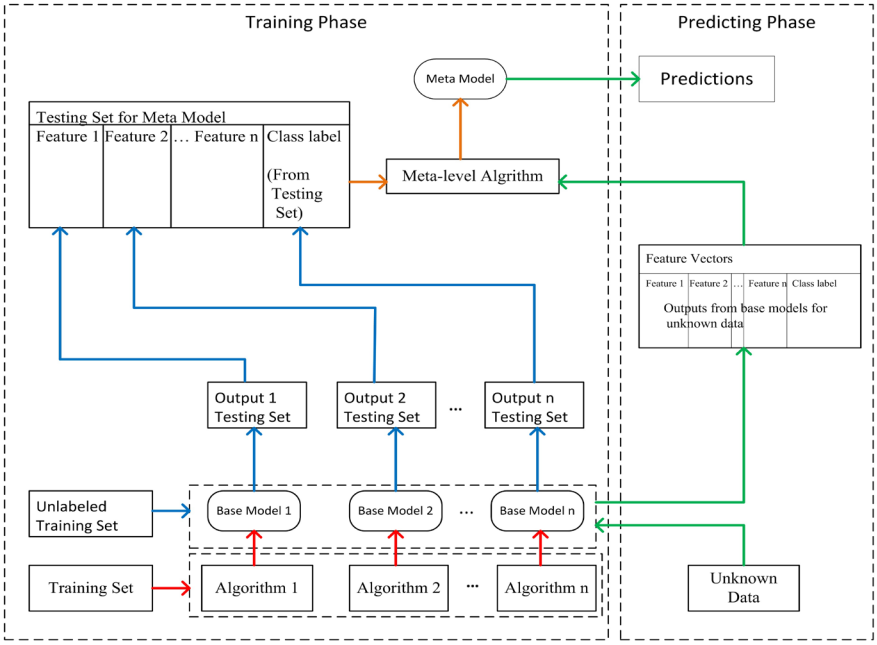

Figure 1. Process of stacking.

diverse measurements of bone density and geometry so as to improve the accuracy of bone fracture prediction compared to standard measurement. The work in [26] examined the generalization behavior by comparing single level learning models to multiple level learning models (stacked generalization method) on a multilayer neural network. The results show that the stacked generalization scheme could improve classification performance and accuracy compared to the single level model. The method is also used in Wang's approach [27] in which the stacking approach performed successfully in predicting membrane protein types based on pseudo-amino acid composition.

There are several schemes proposed for the customized selection of classifiers, thus to find the best base classifier for each individual instance on the basis of local accuracy. For example, the work in [27] proposed the method of Dynamic Integration, in which weighted voting were applied by giving higher weight to a classifier if its training data were in the same region as the testing example. The work in [28] introduced an approach to group classifiers by their similarities and to retain one representative classifier per cluster. The core notion of this type of methods is to enhance the overall performance by choosing and/or assigning more weight to those classifiers that perform best in instances that are similar to the one that is being classified. Also, static ensemble methodologies are gradually developed and used. For instance, the work in [29] proposed a linear ensemble algorithm that takes into account both the accuracy of individual classifiers and the diversity among classifiers, which are also vital factors where importance should be attached when designing ensemble methods.

One of the dataset used in this work is from the Repeat Buyers Prediction Competition [5]. In this two-stage competition with different amount of data

provided, the work by Liu [30] won the first place in Stage 1 by using both feature engineering and model training. The work by He [31] won the first place in Stage 2, in which a four-step solution was used consisting of 1) characteristics analysis and strategy design, 2) feature extraction and selection, 3) data training and 4) hybrid ensemble on both models and features. Both of the award winners suggested that feature engineering is the key element to their works and that the ensemble method applied in these two approaches perform better than any of the individual classifier. However, only linear ensemble methods were utilized in their works. The goal of our approach is to use a non-linear method.

3. The Proposed Ensemble System

3.1. Overview

In this section we use *positive* (+) and *negative* (-) to denote the two classes in a binary classification.

To solve classification problems, a classifier is learnt by an algorithm, so that a hypothesis can be proposed based on the classifier that best fits the given training set. For a dataset, a single classifier may have its own "blind area", *i.e.*, some points cannot be clearly identified, especially when the dataset is high-dimensional or non-linear. For example, in a 2-dimensional space, as shown in **Figure 2**, a plus sign indicates an instance belonging to the positive class, and a minus represents the negative class. Using different single classifiers (green H1, blue H2, and yellow H3 lines, **Figures 2(a)-(c)** the points (circled in red) cannot be correctly classified. By applying the ensemble technique, multiple models can be integrated to reduce those blind areas as shown in the lower three diagrams (**Figures 2(d)-(f)**). Each data instance in the blind areas is then classified by the capable base model. For example, with the combination of H1 and H2 as depicted in **Figure 3**, point 6 can be identified by H1 generated by classifier 1, not circled in red, as is the case in (b), whereas point 2 can be identified by H2 from classifier 2, not circled in red, compared to (a). Accordingly, the combination of H1 and H2 has four unrecognized points, the combination of H2 and H3 has two less unrecognized points, and the combination of H1 and H3 has only one unrecognized point. As a result, classifiers H1 and H3 are considered the proper group for constructing the ensemble model for this example. In summary, high diversities among base models can improve the ability of the ensemble model by satisfying more types of data instances.

In addition, the problem in this example (**Figure 2**) may not be properly solved with the existing ensemble techniques, such as averaging and general stacking method. As illustrated in **Figure 4**, the existing ensemble model is built by simply averaging all the hypotheses (H1, H2, and H3) of the three classifiers, but there are still five unrecognized points for the final hypothesis compared to **Figure 2(f)** with only one unrecognized point. This shows that a pairwise diverse method (as proposed in this paper) is a better option than just averaging or general stacking method.

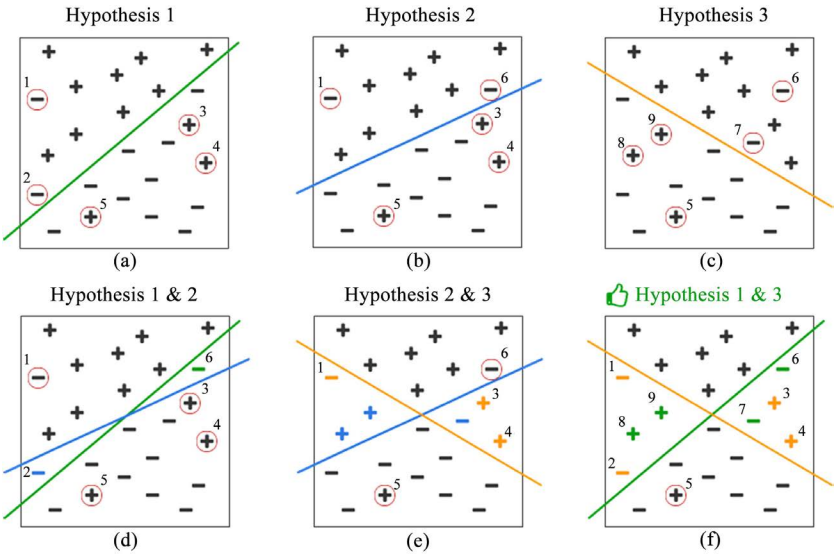

Figure 2. Base model selections using pairwise diversity.

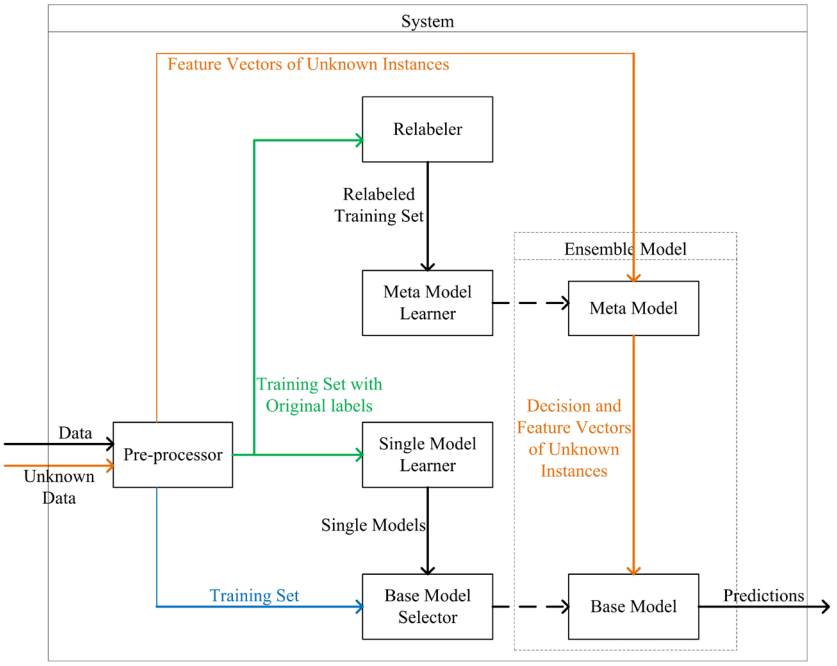

Figure 3. Proposed system's architecture.

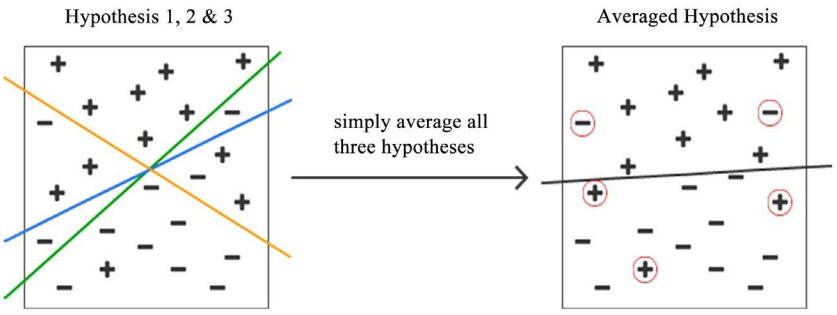

Figure 4. Ensemble model by Averaging.

In the proposed ensemble method, a meta-level model is learnt, which recognizes the capable base classifier for each data instance. In addition to the meta-level model from all trained single classifiers, only those with high pairwise diversities are selected as base models. Finally, the constructed ensemble model is able to support adaptive selection of the proper base model for each data instance. In order to accomplish this, two key problems demand prompt solutions: 1) the pairwise diversity must be defined for the measurement. 2) A training set needs to be designed for learning the meta-level model.

As explained in Section 2.1 a probabilistic classifier performs the classification by ranking the instances according to their predicted probabilities of being positive. Therefore, for the first problem, the pairwise diversity is defined as the difference between the ranks of instances provided by two classifiers. For the second problem, in order to map each data instance to its appropriate base model, the meta-level model should be trained from the same feature vectors that have been used to train the single model. Additionally, each data instance in the training set needs to be relabeled to indicate its proper base model. A base model is considered proper for a data instance, if it can generate the most certain prediction. By ranking all the instances according to their predicted probabilities, the certainty of a prediction is defined as the relative position of an instance in the ordered queue. For example, a prediction for a data instance is considered more certain if a positive instance is ranked at a lower position (or a negative instance at a higher position) in an ascent sorting. Overall, the pairwise diversity is measured on a group of instances for base model selection, while the certainty is on the basis of a single instance for relabeling the training set.

3.2. System Design

The system design for our model consists of seven modules: Pre-processor, Relabeler, Base Model Selector, Single Model Learner, Meta Model Learner, Meta Model and Base Models, which are organized as shown in **Figure 3**. Each module is briefly described as follows.

- The Pre-processor is responsible for data transformation, data normalization and feature engineering, as well as splitting the known data into a training set and a testing set.
- The Single Model Learner learns single classifiers from the training set with the original labels for data instances.
- The Base Model Selector takes both the testing set and the outputs of the Single Model Learner as inputs, to determine the Base Models for constructing the ensemble.
- The Relabeler receives the training set with original labels for data instances, and then relabels the training set with the outputs of the Base Model Selector. The relabeling of each instance is performed according to the behaviors of Base Models on the original training set.
- The Meta Model Learner uses the relabeled training set to train the meta

model.

- The Meta Model takes the feature vector of each unknown instance to predict its most suitable base model, and then passes the feature vector to the selected Base Model for the final prediction, which is the output of the system.

3.3. Base Model Selector and Re-Labeler

The Base Model Selector is the module that determines the base models for constructing the final ensemble model. The pairwise diversities are calculated from the ranks, and the algorithm for determining the single models based on diversities is described as follows:

Algorithm 1: Base Model Selection

1) Given a testing set $S=\{s_i \mid i=1,2,\cdots,n\}$ and a group of trained single classifiers $C=\{c_j \mid j=1,2,\cdots,m\}$, where n and m denote the number of samples in testing set and the number of trained single classifiers, respectively;
2) **For** each classifier c_j in C:

a) Use c_j to make predictions for each s_i in S to obtain a set of probabilities $P_j=\{p_{ij} \mid i=1,2,\cdots,n\}$, where p_{ij} represents the probability of s_i being positive generated by c_j;

b) Sort all s_i according to their p_{ij} in ascending order, and then obtain a set of ranks $R_j=\{r_{ij} \mid i=1,2,\cdots,n\}$, where r_{ij} denotes the rank of s_i in all ordered test samples given by c_j;

3) **End For**
4) Calculate the root-mean-square deviation (RMSD) of each set of two R_j as the metric of pairwise diversity

$$D_{pq}=\sqrt{\frac{\sum_{i=1}^{n}\left(r_{ip}-r_{iq}\right)^2}{n}};$$

5) Choose the k single classifiers c_j with highest RMSDs as the base models for constructing the ensemble.

The function of Re-labeler is to design a new training set for training the meta-level model. The aim is to identify the proper base model for a given instance. A training set consists of two parts: feature vectors and labels. The former describes the characteristics of data instances, while the latter indicates the class of instances in a class space. Accordingly, the algorithm for relabeling is designed as follows:

Algorithm 2: Relabeling Training Set for Meta Model

1) Given a training set $S=\{s_i \mid i=1,2,\cdots,n\}$ obtained from pre-processor and a group of base classifiers $B=\{b_j \mid j=1,2,\cdots,m\}$ determined by base model selector;
2) **For** each base classifier b_j in B:

a) Use b_j to make predictions for each s_i in S to obtain a set of probabilities $P_j=\{p_{ij} \mid i=1,2,\cdots,n\}$, where p_{ij} represents the probability of s_i being positive generated by b_j;

b) Sort all s_i according to their p_{ij} in ascent order, and then obtain a set of ranks $R_j = \{r_{ij} \mid i = 1, 2, \cdots, n\}$, where r_{ij} denotes the rank of s_i in all ordered samples given by b_j;

3) End For

4) **For** each training sample s_i in S:

5) **If** s_i is positive: relabel s_i as b_j which generates the largest r_{ij};

6) Else

7) **If** s_i is negative: relabel s_i as b_j which generates the smallest r_{ij};

8) End If

9) End For

10) **Return** the relabeled training set

The training set derived from Algorithm 2 represents the proper base model for each instance, and then it is further used for training meta-level model, so that the meta-level model can be used to predict the appropriate base models for unknown instances.

4. Experiment and Results

4.1. Datasets and Experiment Environment

The first dataset used in our experiment is the Repeat Buyer Prediction (RBP) dataset, obtained from a machine learning competition, which was held by International Joint Conference on Artificial Intelligence (IJCAI) and Alibaba Group in 2015 [5]. The repeat buyer dataset consists of the behavior log of anonymized users accumulated during the 6 months before and on the "Double 11" day, with labels indicating whether a customer is a repeat buyer of a merchant or not. The dataset is 1.92 GB in total and stored as three comma-separated values (CSV) files: user behavior logs (**Table 1**), user profile information (**Table 2**), and training and testing (**Table 3**).

The Censure Income Prediction (CIP) data was extracted from the census bureau database. It is popularly used in research area for validating and evaluating binary classification algorithms and methods, as it does not require feature engineering and it is relatively convenient to use as a control data. There are 6 continuous attributes and 8 categorical attributes in the raw data. **Table 4** lists the definitions of each data field. Note that the fields in data type of string are all categorical, therefore they need to be transformed to proper formats during pre-processing for further operations.

All the developments and experiments are conducted on a customized compute engine from Google Cloud Platform (Google cloud computing, hosting services & APIs, 2016), Ubuntu 16.04 LTS operating system with 6 cores (2.3 GHz) virtualized from Haswell processors, 32 GB memory and 128 GB SSD.

4.2. The Procedure

The datasets are separately preprocessed using Pandas. In the case of the RBP dataset, firstly, the data stored in three separate tables are merged and to avoid

Table 1. User behavior Logs in RBP.

Data Field	Data Type	Definition
user_id	integer	A unique id for the shopper
merchant_id	integer	A unique id for the merchant
item_id	integer	A unique id for the item
brand_id	float	A unique id for the brand of the item
cat_id	integer	A unique id for the category of the item
time_stamp	integer	Date the action took place (format: mmdd)
action_type	integer	Type of the action, which is enumerated as the set {0, 1, 2, 3}, where 0 represents click, 1 is for add-to-cart, 2 denotes purchase and 3 is for add-to-favorite

Table 2. User profile information in RBP.

Data Field	Data Type	Definition
user_id	integer	A unique id for the shopper
gender	float	Gender of the shopper: 0 for female, 1 for male, 2 and NULL for unknown
age_range	float	Age range of the shopper: 1 for less than 18, 2 for [18, 24], 3 for [25, 29], 4 for [30, 34], 5 for [35, 39]; 6 for [40, 49], 7 and 8 for more than 49, 0 and NULL for unknown

Table 3. Training and Testing Data in RBP.

Data Field	Data Type	Definition
user_id	integer	A unique id for the shopper
merchant_id	integer	A unique id for the merchant
label	integer	A binary label {0, 1} indicating whether a shopper is a repeat buyer of a merchant, where 1 represents repeat buyer and 0 is for non-repeat buyer

Table 4. Data fields in CIP

Data field	Data type	Definition
age	integer	The age of an anonymous person
workclass	string	The work class of an anonymous person
fnlwgt	integer	Independent estimates of the civilian non-institutional population of the US
education	string	The highest degree of an anonymous person
education_num	integer	Number of years an anonymous person is under education
marital_status	string	The marital status of an anonymous person
occupation	string	The occupation of an anonymous person
relationship	string	The role of an anonymous person in a family
race	string	The race of an anonymous person

Continued

sex	string	The gender of an anonymous person
capital_gain	integer	The capital gain of an anonymous person
capital_loss	integer	The capital loss of an anonymous person
hours_per_week	integer	The working hours of an anonymous person per week
Native_country	string	The native country of an anonymous person
Labels	string	Indicates whether an anonymous person makes over $50K/yr

producing redundant rows, the three tables are merged by inner join. Then feature engineering is applied. After pre-processing, single models are trained using logistic regression, gradient boosting machine, AdaBoost, random forests and factorization machine. Then the trained models are tested on the testing set for evaluations. The final ensemble model was constructed based on the trained single models which are summarized as follows: determine base models for combining; relabel the training set and learn the meta-level model; and use meta-level model to combine the selected base models.

With the base models determined, the training set that had been used for training single models was relabeled, so that a meta-level classifier can be trained on this relabeled training set, and it is capable of recognizing the proper base model for each testing instance. Similar to calculating the pairwise diversities, the new labels are also determined based on ranking and base models are applied on the training set rather than the testing set. For simplicity, it is assumed that the number of base models is two, and the two base models are logistic regression and GBM (lr and xgb). Therefore, the new class space consists of two class labels (lr and xgb), and the trained meta-level model is a binary classifier.

The last step of the ensemble model construction is utilizing the trained meta-level model to combine the base models. **Figure 5** demonstrates the principle of the combination. The meta-level model works as a "consultant" for the unknown instances, it recommends a proper base model for each unknown instance, and then the recommended base model makes the prediction for this unknown instance. By applying this method, all the base models are integrated to make predictions for unknown instances.

4.3. Experiment Design

The experiments are designed using AUC and F_1 score to evaluate the performance and overhead that is introduced to the system in terms of time consumption. In addition, both single models and ensemble models are taken into account for the evaluation. For each single model, the changes in performance are traced as its parameters vary. Multiple parameters are selected for each single model, and only one of them is tuned each time with other parameters unchanged. The combination of a set of parameters leading to a relatively best performance for each single model is considered as the best parameters for the single

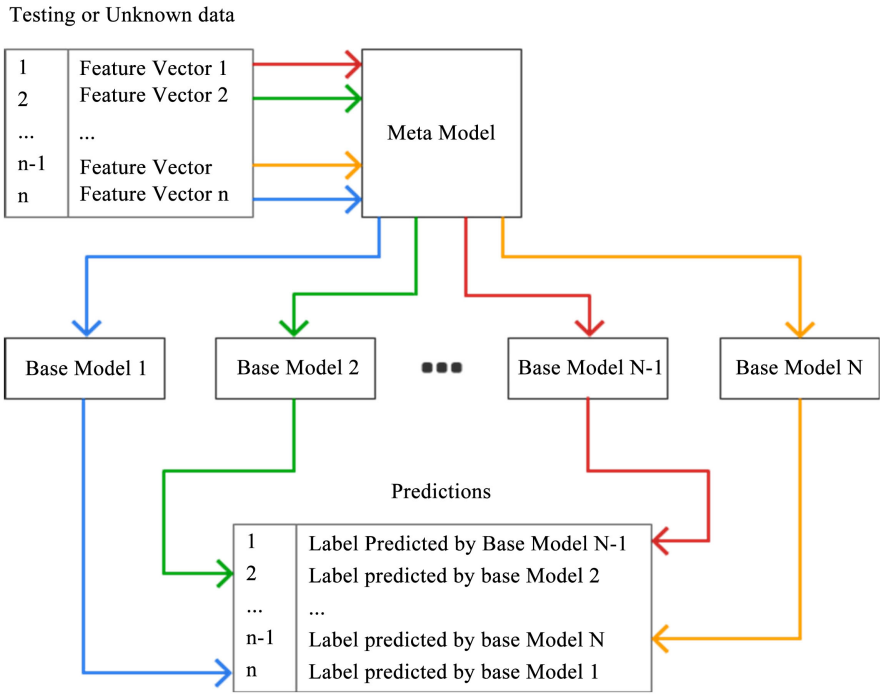

Figure 5. The working principle of the ensemble model.

model. Then single models with best parameters are compared horizontally.

Four machine learning algorithms (Logistic Regression, AdaBoost, Xgboost and Random Forests) are selected to train single models, and the list of tuned parameters for these various single models is highlighted as follows:

- **C:** Inverse of regularization strength is used to prevent the model from over-fitting. Smaller values specify stronger regularization;
- **max_iter:** the maximum number of iterations for Logistic Regression;
- **n_estimator:** the number of constructed decision trees for the tree-based models (AdaBoost, Xgboost, Random Forests);
- **learning_rate:** a rate indicating the learning rate of loss function solver;
- **max_depth:** the maximum tree depth of the constructed decision tree for the tree-based models.

4.4. Results and Analysis

This section presents the experimental results, which include the evaluation and analysis of each of the four selected single models (Logistic Regression, AdaBoost, Xgboost, and Random Forests) and a comparison of multiple ensemble methods. Note that in this paper, for simplicity only two single models are selected as base models for combining. Three methods of base model selection are evaluated and compared, and they are listed as follows:

- **RMSD (Root-mean-square derivation) diversity-based:** the two single models with highest pairwise RMSD of all are selected as the base models for combining;
- **ZOL (Zero-one loss) diversity-based:** the two single models with highest

pairwise ZOL of all are selected as the base models for combining;

- **Performance-based:** the two single models with highest performance (AUC and F_1 score) of all are selected as the base models for combining.

In addition, three other ensemble methods are adopted in the paper for comparison with the proposed ensemble method, and they are listed as follows:

- **Averaging:** final predictions are obtained from averaging the outputs of the two base models;
- **Stacking with Logistic Regression as the Meta model:** Logistic Regression is used as the Meta model. In this case there is a holdout validation dataset for base models, and the training set for meta model consists of the outputs of the two base models on the validation set (as features) and the original labels (as label);
- **Stacking with Xgboost as the Meta model:** same with previous one except for Xgboost as Meta model.

Figure 6 shows the result of the AdaBoost performance for the RBP dataset, from which no direct connection was found between the value of AUC and F_1 score. A higher AUC score does not necessarily correspond to a higher F_1 score. For the RBP dataset, as the value of n_estimator grows, both AUC and F_1 score increased at first and then decreased, but with the different inflection point.

Figure 7 shows the result of AdaBoost performance in CIP dataset, from which the data suggested that a larger n_estimator leads to a higher AUC score. However, as n_estimator grows, the growth of AUC slowed and remained nearly constant. On the other hand, a larger learning rate also leads to a better performance for both AUC and F_1 score on CIP.

Figure 8 shows the performance of Logistic Regression on RBP dataset. A relatively smaller C (stronger regularization) indicates a very strict penalty on the error instance when learning the classifier. It is observed that a smaller C has benefits for both AUC and F_1 score on RBP. On the other hand, as max_iter increases, a growth in performance is observed to certain extent. For CIP, (see Figure 9), it is observed that C contributed less to the overall performance, and the impact is relatively random. In addition, since the number of instances is not big in CIP dataset and the number of features is also limited, the model completely converges within 100 iterations. Thus, when max_iter further increases, almost no effect is observed to both AUC and F_1 score.

The Figure 10 and Figure 11 show the performance of Xgboost on RBP and CIP, respectively. For both RBP and CIP, a relatively lower learning rate is preferred. Both n_estimator and max_depth represent the complexity of a model, in which max_depth indicates the maximum allowable depth of each tree. Normally, a model with relatively lower complexity could benefit AUC. On the other hand, a model with relatively higher complexity could increase F_1 score. Especially in CIP, as shown in Figure 11, a higher max_depth could increase both AUC and F_1 score.

Figure 12 and Figure 13 present the performance of Random Forest in RBP

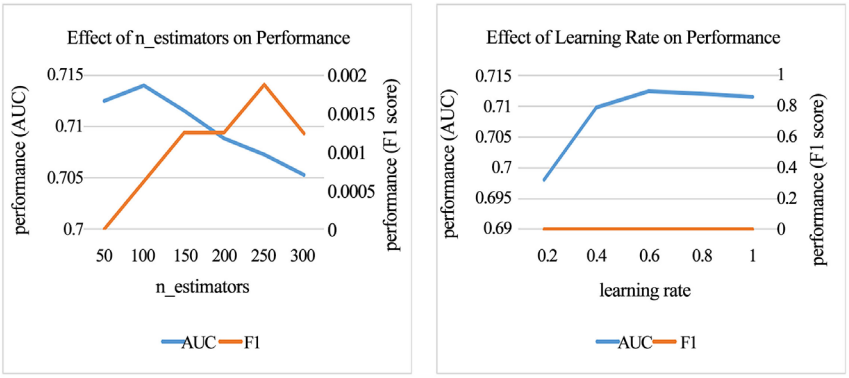

Figure 6. AdaBoost performances in RBP.

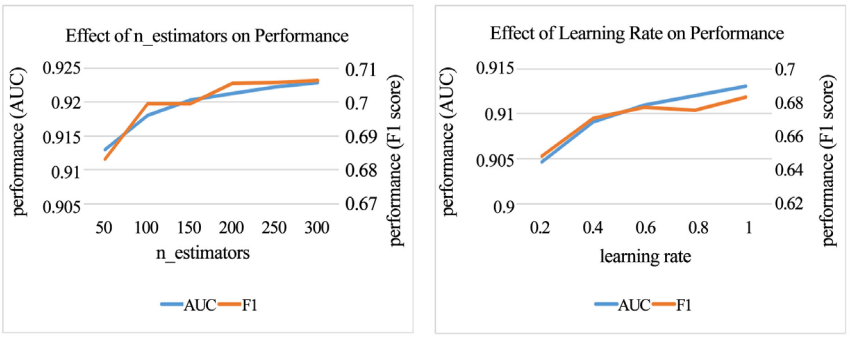

Figure 7. AdaBoost performances in CIP.

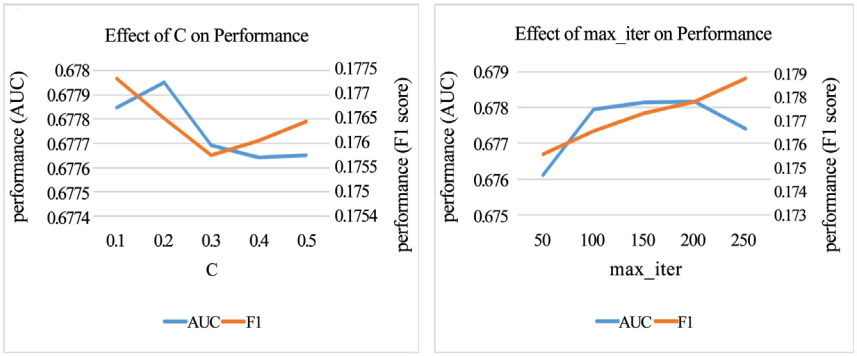

Figure 8. Logistic regression performances in RBP.

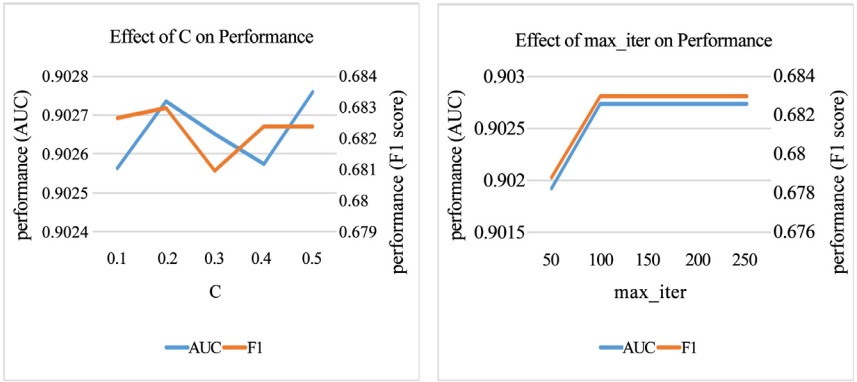

Figure 9. Logistics regression performances in CIP.

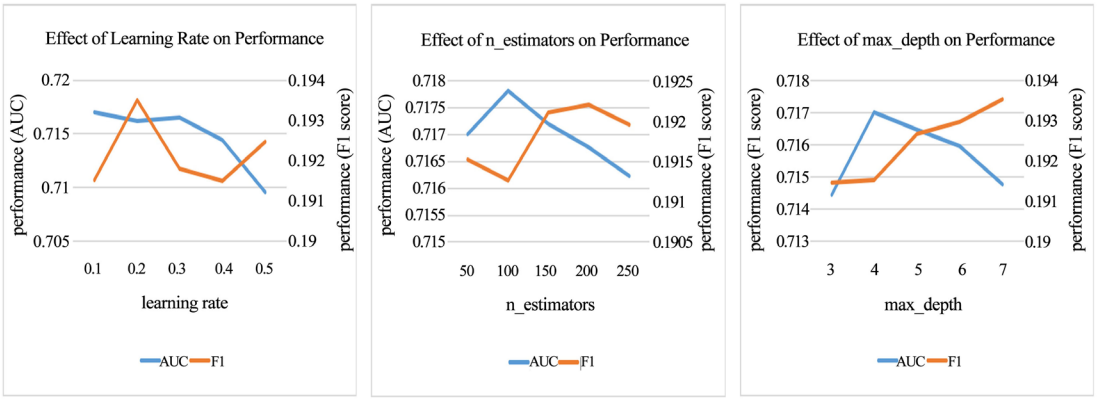

Figure 10. Xgboost performances in RBP.

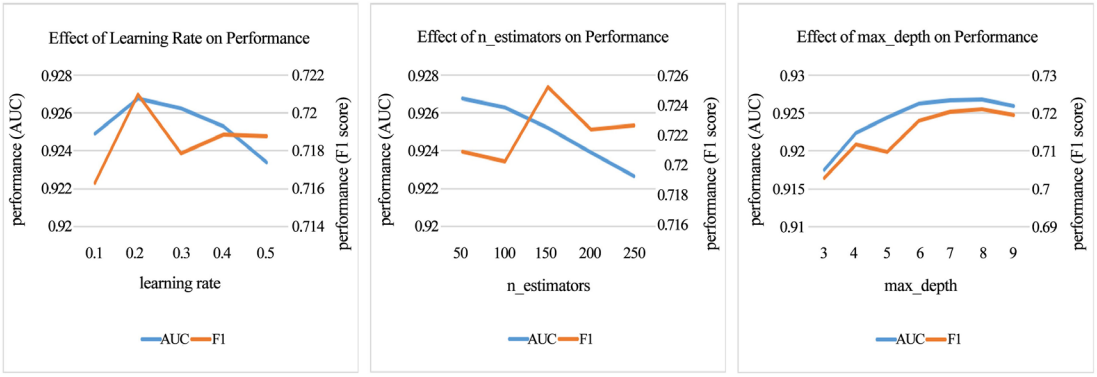

Figure 11. Xgboost performances in CIP.

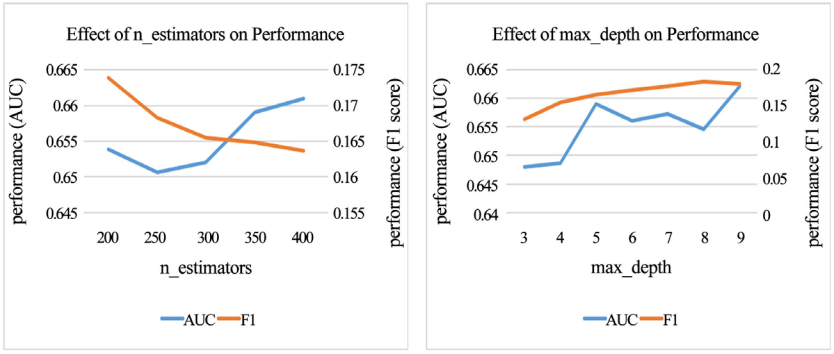

Figure 12. Random forest performances in RBP.

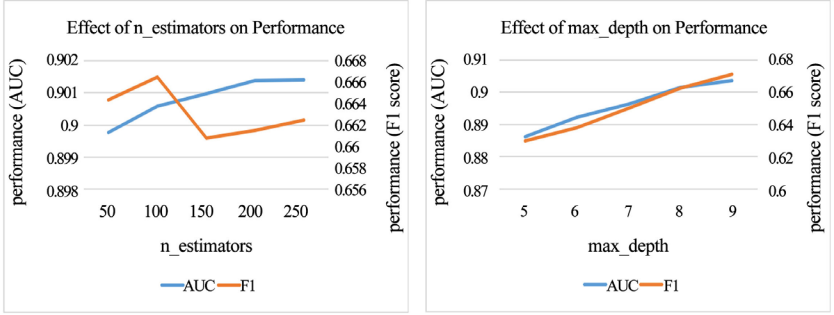

Figure 13. Random forest performances in CIP.

and CIP datasets. It is observed that the increase in n_estimator helps improve AUC. On the contrary, higher F_1 scores were obtained with low n_estimator values. The increasing value of max_depth has positive impact on both AUC and F_1 score for both datasets. Note that there is relatively large fluctuation in performance regarding Random Forests in the case of AUC.

In summary, both the AUC and F_1 score results obtained from Xgboost are better than that of other methods and AdaBoost does not perform well on F_1 score. On the other hand, it is worth mentioning that the performance of the Random Forest on RBP is not stable. The variance of both AUC and F_1 score is relatively high when the model is trained with the same parameter values several times by Random Forest. The general performance of Random Forest is also lower than that of those three other models, especially for RBP. These results indicate that Random Forest may not be appropriate for the high complexity datasets.

4.5. Overhead

Time consumption is considered as a typical overhead regarding any engineering problem. In this paper, time spent in both learning and predicting is evaluated as index for all models. The Figure 14 and Figure 15 show the overhead for AdaBoost in RBP and CIP datasets. It can be observed that increasing n_estimators leads to a noticeable increase in overhead. On the other hand, increased learning rate seems to contribute less to the computational overhead.

Figure 16 and Figure 17 present the overhead in Logistic Regression. There is little effect by C. This is reasonable as C indicates the degree of penalty rather

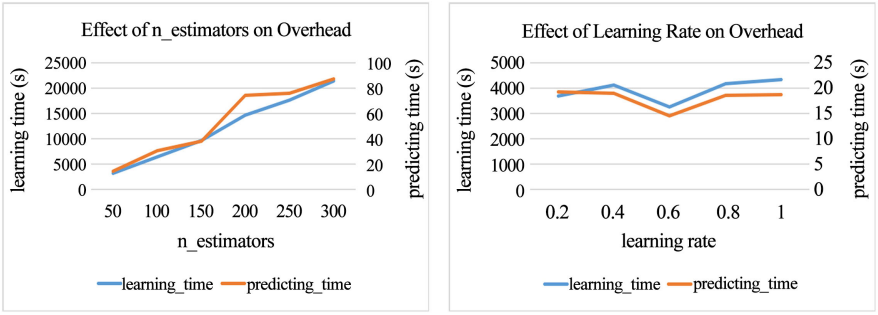

Figure 14. AdaBoost overhead in RBP.

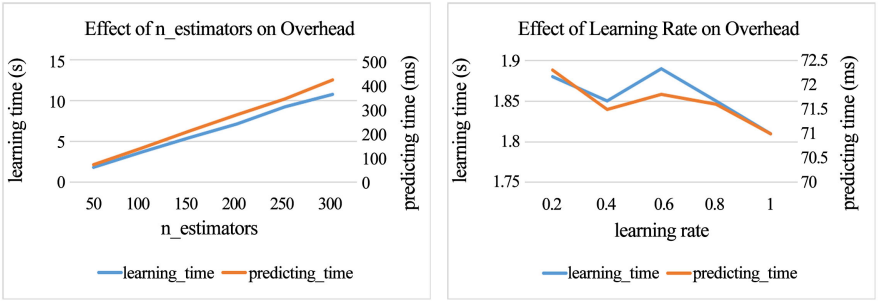

Figure 15. AdaBoost overhead in CIP.

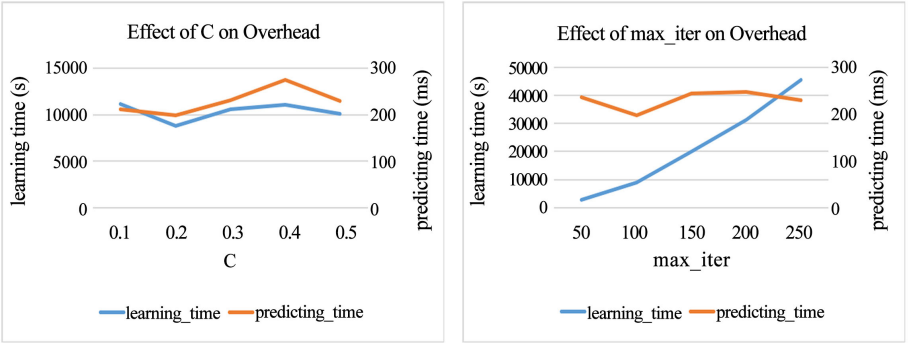

Figure 16. Logistic regression overhead in RBP.

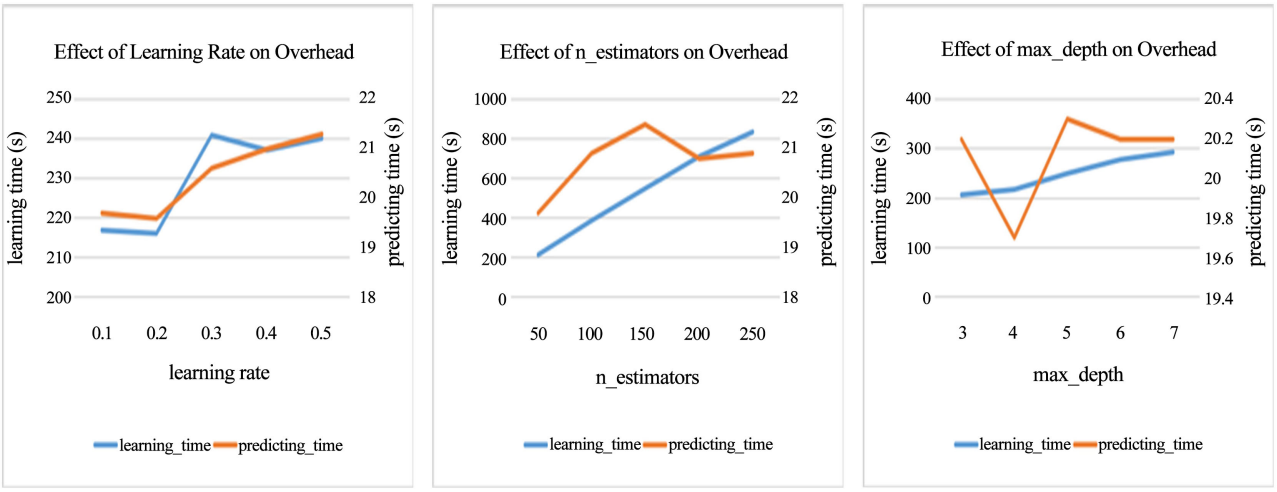

Figure 17. Logistic regression overhead in CIP.

than the learning rate. When the data size becomes bigger and the complexity is higher, then in RBP, max_iter becomes the important determining factor for learning time. Different from tree-based methods, the complexity of the Logistic Regression model does not increase as the two parameters (C and max_iter) vary. Hence, the prediction time was seldom affected, and it always maintains a stable state.

The **Figure 18** and **Figure 19** show the overhead of Xgboost on RBP and CIP. Overall, the increase in learning rate, n_estimator and max_depth may lead to a slight increase in overhead over time. However, there is an exception in CIP. The growth of learning rate can lead to a decrease of overhead. Since the magnitude was too small (<10 ms), this impact is negligible.

4.6. Best Performance Single Models and Analysis of Ensemble Model

The **Figure 20** and **Figure 21** show the overhead for Random Forest in RBP and CIP. It is observed that both n_estimator and max_depth can affect learning time to a large extent, while max_depth has little impact on predicting time, and the effect is relatively random.

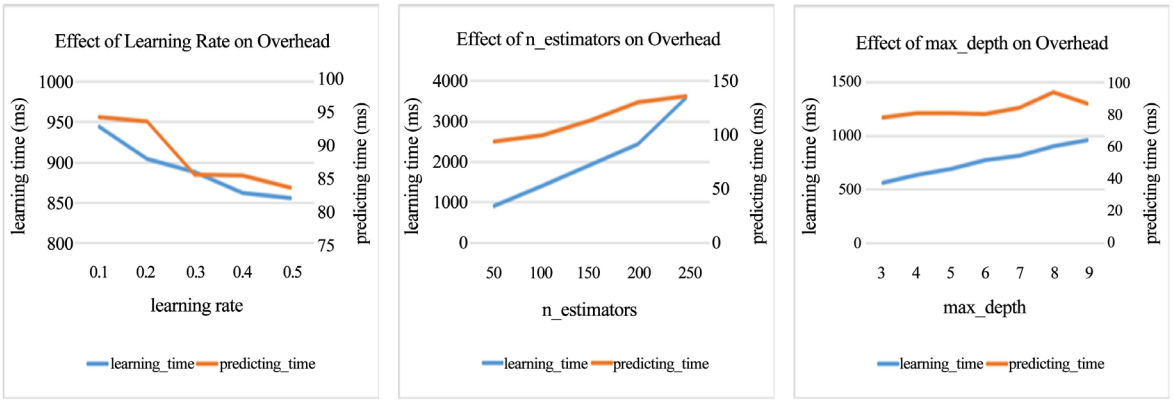

Figure 18. Xgboost overhead in RBP.

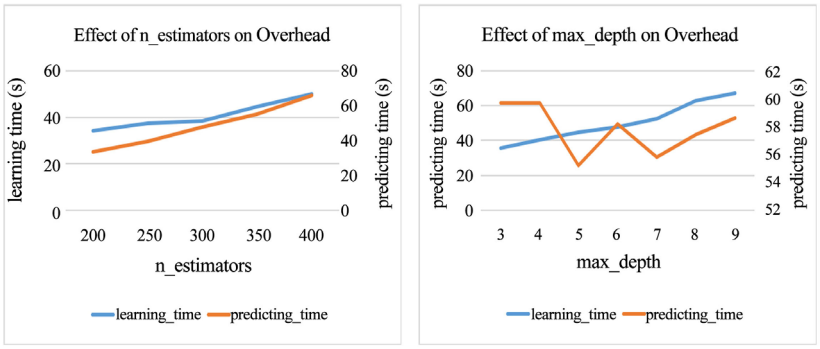

Figure 19. Xgboost overhead in CIP.

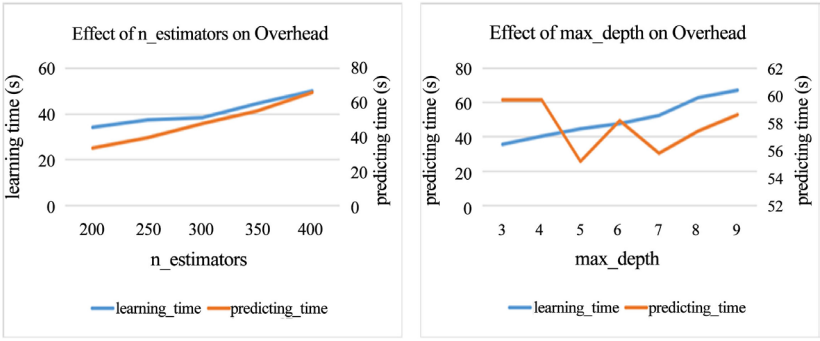

Figure 20. Random forest overhead in RBP.

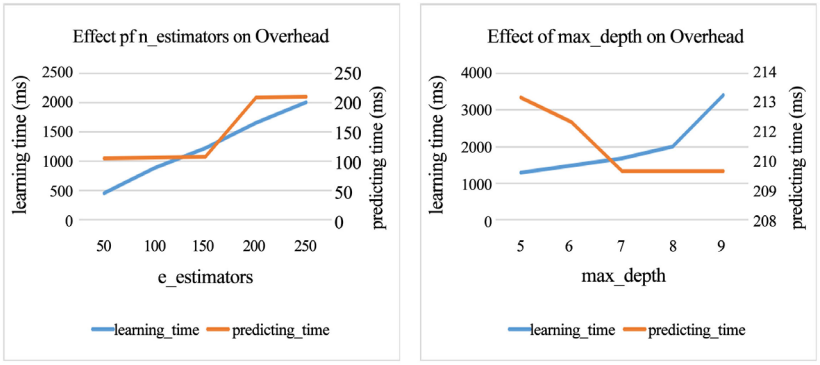

Figure 21. Random forest overhead in CIP.

In summary, the learning time of Logistic Regression can be extremely high on the datasets in large size and high complexity datasets. The same effect for AdaBoost but better than Logistic Regression. However, the prediction time of Logistic Regression is stable. The complexity of the model has effects for all three tree-based models on both learning time and predicting time. On the other hand, the overall overhead introduced by Xgboost is low, and a better scalability is observed for Xgboost. In addition, all these four single models introduce much less overhead when working on the smaller and simpler dataset (CIP).

The Figure 22 and Figure 23 show the performance comparison among different ensemble methods for both RBP and CIP dataset. As shown in the figures, the differences among the single models are larger for RBP than that of CIP. The main reason is likely to do with the data size and the complexity of data. In summary, Xgboost performs the best for both AUC and F_1 score on both RBP and CIP. All the four single models can deliver a considerable performance for both AUC and F_1 score on CIP.

To determine the base models for combining, pairwise diversities are measured using two methods: RMSD and ZOL. The diversities in RBP and CIP are presented in Table 5 and Table 6 respectively. The base models with the largest pairwise diversity value are selected for combination in the ensemble method. Also, in this paper, since only 2 single models are selected as base models for combining, the Meta model is learnt as a binary classifier.

In the case of RBP, Logistic Regression and Random Forests are selected as base models according to the RMSD diversity, while Logistic Regression and AdaBoost are selected according to the ZOL diversity. For CIP, logistic regression and Xgboost are selected based on RMSD diversity, while AdaBoost and random forests are selected in the case of the ZOL diversity. In addition, AdaBoost and Xgboost are also selected, since they generate the best performances among all the single models for both RBP and CIP. The time for calculating RMSD and ZOL is similar on both RBP and CIP.

Table 5. Diversity between single models in RBP.

Diversity	lr & ab	lr & xgb	lr & rf	ab & xgb	ab & rf	xgb & rf	overhead
RMSD	6719	5969	12827	3773	11,821	11,875	103 s
ZOL	18,859	5921	15,086	18,292	3987	14,779	101 s

Table 6. Diversity between single models in CIP.

Diversity	lr & ab	lr & xgb	lr & rf	ab & xgb	ab & rf	xgb & rf	overhead
RMSD	896	1036	759	590	909	907	0.638 s
ZOL	1458	808	876	1350	2024	944	0.619 s

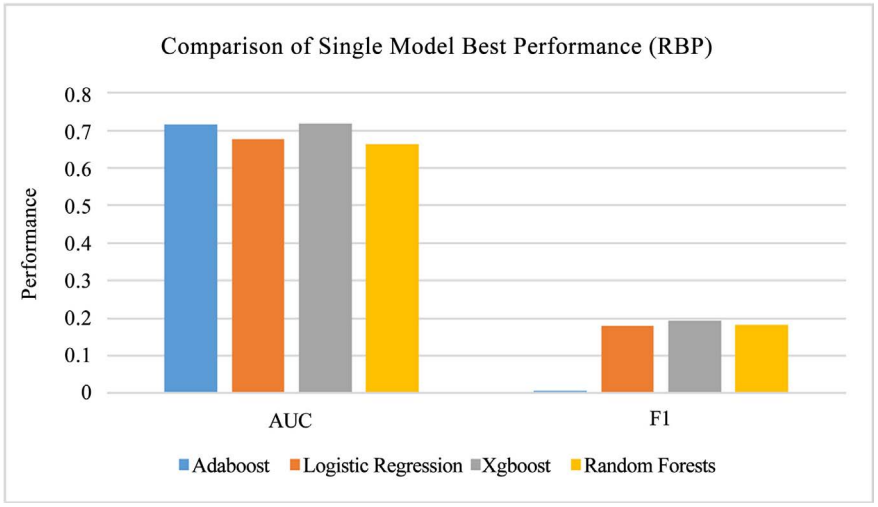

Figure 22. Comparison of single model's best performance on RBP.

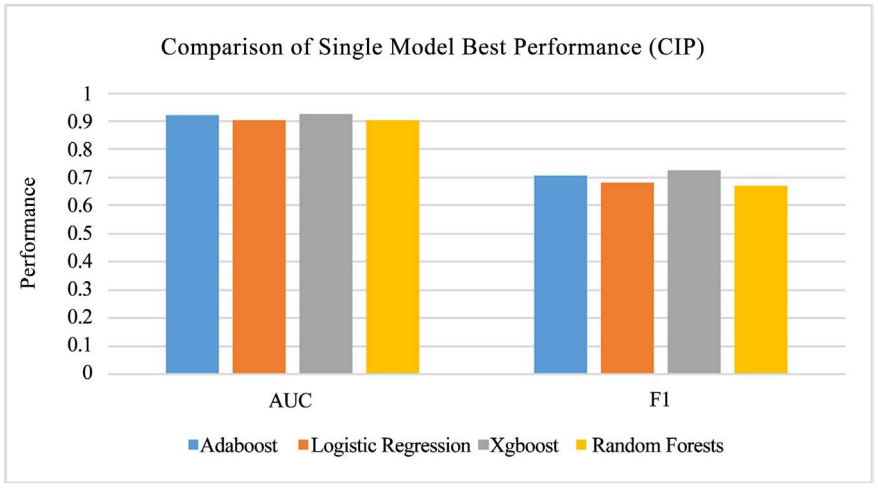

Figure 23. Comparison of single model's best performance on CIP.

This improvement also indicates that the proposed ensemble method has an exceptional ability of dealing with imbalanced datasets. Moreover, a higher pairwise diversity of combined base models can also lead to a further improvement toward the performance of ensemble model.

In the following using the results of the single model best performance (**Figure 22** and **Figure 23**) as a base, the proposed ensemble method is compared to Averaging, Stacking and the best base model on both RBP and CIP. Furthermore, a linear (Logistic Regression) and a non-linear (Xgboost) Meta models are both tested for Stacking.

As can be seen in the **Figure 24**, the averaging of base models, which are selected based on RMSD diversity, generates a little higher AUC performance result than the best base model (Logistic Regression). The model from the proposed method based on RMSD diversity also achieves a slightly higher performance than that of the best base model (but slightly lower than averaging). However, the other two methods, *i.e.*, Stacking (lr) and Stacking (xgb), failed to

further enhance AUC.

As shown in Figure 25, compared to the best base model (Logistic Regression, AdaBoost, and Xgboost) the proposed method and averaging both have increased F_1 score for all the three base models. The combination of two single models with highest ZOL diversity leads to the highest improvement on F_1 score, especially for the proposed method (18.5%). All Stacking methods fail to improve the F_1 score.

The Figure 26 and Figure 27 show the comparisons between different ensemble methods for CIP dataset. Similar to RBP, these ensemble methods do not help much in improving AUC score. The proposed method with different base model selections all contribute to increase F_1 score. Compared to other ensemble methods, the proposed method performs better in combining base models using diversity. When utilizing the stacking method with linear Meta model, the combination of two single models with highest ZOL diversity shows an obvious negative impact on AUC result.

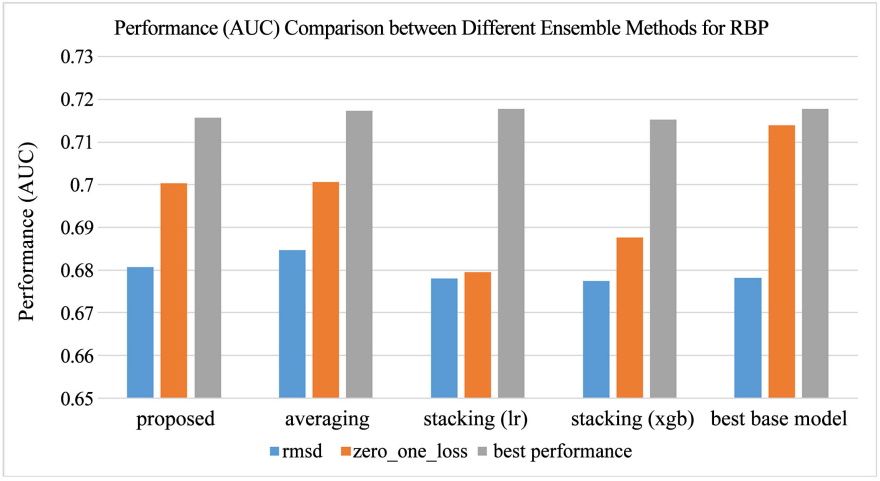

Figure 24. AUC comparison of different ensemble methods for RBP.

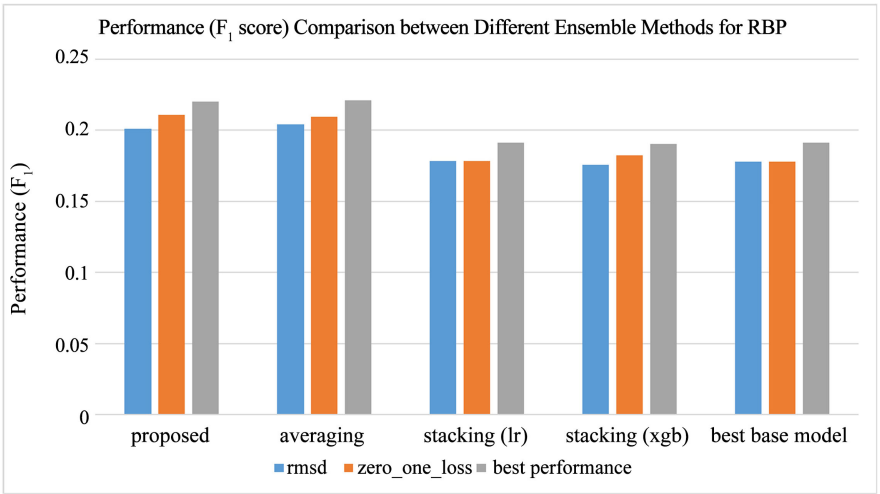

Figure 25. F_1 score comparison of different ensemble methods for RBP.

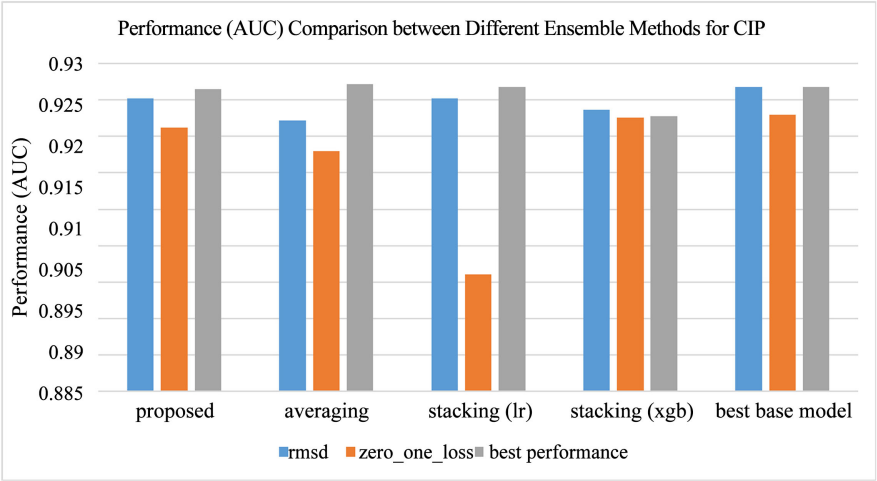

Figure 26. AUC comparison of different ensemble method for CIP.

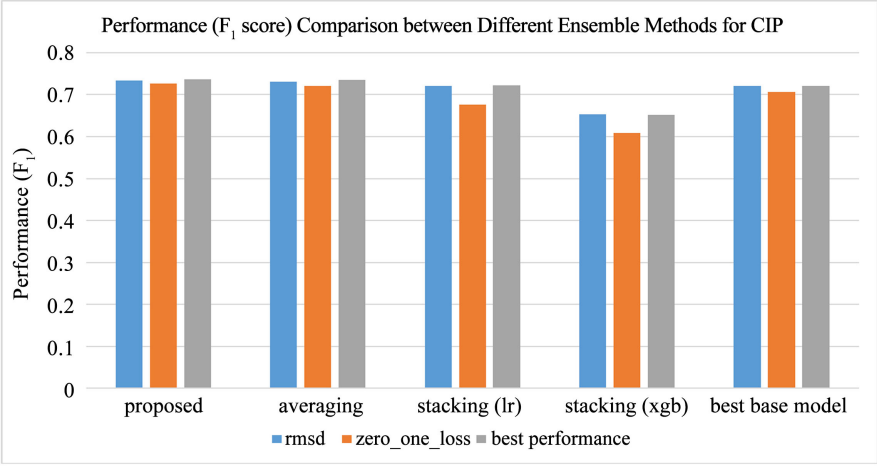

Figure 27. F_1 score comparison of different ensemble method for CIP.

In summary, Averaging and Stacking methods perform slightly better in combining single models with best performance while the proposed method contributes more to increase F_1 score, compared to the other three ensemble methods. For RBP, the largest improvement achieved by the proposed ensemble method is around 18.5% on F_1 score. This can be very valuable for solving practical engineering problems, as the improvement on F_1 score means that the predictive system can help the merchants reach more real repeat buyers with the same cost of promotions. Moreover, this level of improvement is achieved using only the offline version training set. The competitors of the IJCAI, 2015 [5] competition had more data including the complete offline version training set and another online version training set (which is five times the offline version). The highest AUC achieved by the champion of the competition [30] is 0.712, while the best AUC for RBP achieved in this paper is 0.718 and this could be higher with more training data. This is an improvement of roughly 1.0%.

4.7. Overhead of the Ensemble Methods

The Table 7 and Table 8 show the computational overhead of the ensemble methods for both RBP and CIP datasets. It can be observed that the proposed ensemble method introduced higher overhead in both relabel and meta-level model training, but the time consumption is still within an acceptable range.

5. Conclusions and Future Work

In this paper, a novel ensemble method is proposed, which is capable of performing adaptive selection of the best base model for each unknown instance. The base models are selected according to the pairwise diversities of all trained single models. Two metrics (RMSD and ZOL) are used as the measure of diversity. In addition, a relabeled training set is used for learning a meta-level model, which can identify the proper base model for each unknown instance. The relabeling is performed according to the behaviors of the base models on the training set. Also, four algorithms (AdaBoost, Logistic Regression, Xgboost and Random Forest) were used for model selection and relabeling respectively. Finally, the proposed method is validated on RBP and CIP datasets, and the performance of AUC and F_1 score was compared with those of other two existing ensemble techniques (Averaging and Stacking). Our main focus is the RBP dataset so that we can compare the end results to the IJCAI repeat buyers 2015 competition [5]. The highest AUC achieved by the champion of the competition [30] was 0.712, while the best AUC for RBP achieved in this paper is 0.718 and this could be higher with more training data. This is an improvement of roughly 1.0%. This relatively small improvement is significant considering the arithmetic behind the calculation of AUC and the size of our training data.

Additionally, the proposed ensemble method has an overall performance improvement in terms of F_1 score, which is considered a more valuable metric in practice. There is a significant performance improvement of 18.5% in RBP dataset when adopting the proposed ensemble method compared to the best base model. This improvement can lead to important benefit for merchant by helping them reach more repeat buyers with improved return on investment. This improvement also indicates that the proposed ensemble method has an exceptional ability of dealing with imbalanced datasets. Moreover, a higher pairwise diversity of combined base models can also lead to a further improvement toward the performance of ensemble model.

The limitations in this work include the following. The meta-level model is trained as a binary classifier, thus, only two base models can be combined. A further improvement on performance of ensemble model can be achieved when combining more than two base models [32]. Also, only AUC and F_1 score are used for evaluating the performance and only two types of diversity are measured for base model selection. Different measurements of diversity could result in different combination of base models, thus, more methods for measuring the pairwise diversity are needed [33].

Table 7. Ensemble overhead in RBP.

Base Model	Proposed Method		Stacking (lr)	Stacking (xgb)
	Relabel	Meta Model Training		
lr & rf	411 s	188 s	0.444 s	1.39 s
lr & ab	341 s	188 s	0.418 s	1.43 s
ab & xgb	388 s	190 s	0.488 s	1.29 s

Table 8. Ensemble overhead in CIP.

Base Model	Proposed Method		Stacking (lr)	Stacking (xgb)
	Relabel	Meta Model Training		
lr & xgb	33.8 s	3.45 s	0.278 s	0.323 s
rf & ab	35.46 s	3.38 s	0.278 s	0.309 s
ab & xgb	34.77 s	4.46 s	0.255 s	0.291 s

Some potential directions for the future include the following. The combination of base models could be determined on the basis of predicted probabilities from the meta-level model. In this way, bias and variance could be eliminated to some extent. For imbalanced dataset, more effort could be apply to finding a solution that increase the performance of meta-level model in minority positive instances, rather than overall performance.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] David, O. and Maclin, R. (1999) Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, **11**, 169-198. https://doi.org/10.1613/jair.614

[2] Kuncheva, L.I. (2005) Diversity in Multiple Classifier Systems. *Information Fusion*, **6**, 3-4. https://doi.org/10.1016/j.inffus.2004.04.009

[3] Canuto, A.M.P., *et al.* (2005) Performance and Diversity Evaluation in Hybrid and Non-Hybrid Structures of Ensembles. *Fifth International Conference on Hybrid Intelligent Systems*, Rio de Janeiro, 6-9 November 2005, 6. https://doi.org/10.1109/ICHIS.2005.87

[4] Canuto, A.M.P., *et al.* (2006) Using Weighted Dynamic Classifier Selection Methods in Ensembles with Different Levels of Diversity. *International Journal of Hybrid Intelligent Systems*, **3**, 147-158. https://doi.org/10.3233/HIS-2006-3303

[5] Repeat Buyers Prediction Competition. http://ijcai-15.org/index.php/repeat-buyers-prediction-competition

[6] Archive.ics.uci.edu. (2016) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/

[7] Fan, X., Lung, C.-H. and Ajila, S.A. (2017) An Adaptive Diversity-Based Ensemble

Method for Binary Classification. *IEEE* 41*th Annual International Computers, Software* & *Applications Conference* (*COMPSAC*'17), Torino, 4-8 July 2017. https://doi.org/10.1109/COMPSAC.2017.49

[8] David, B. (2012) Bayesian Reasoning and Machine Learning. Cambridge University Press, New York.

[9] Sariel, H.-P., Roth, D. and Zimak, D. (2002) Constraint Classification: A New Approach to Multiclass Classification. International Conference on Algorithmic Learning Theory. Springer, Berlin Heidelberg.

[10] Ethem, A. (2010) Introduction to Machine Learning. The MIT Press, Cambridge, 249-256.

[11] Friedman, J.H. (2001) Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, **29**, 1189-1232. https://doi.org/10.1214/aos/1013203451

[12] Ron, K. and Provost, F. (1998) Glossary of Terms. *Machine Learning*, **30**, 271-274. https://doi.org/10.1023/A:1007411609915

[13] Tom, F. (2006) An Introduction to ROC Analysis. *Pattern Recognition Letters*, **27**, 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

[14] Wikipedia (2016) Receiver Operating Characteristic. https://en.wikipedia.org/wiki/Receiver_operating_characteristic

[15] William, C., Ravikumar, P. and Fienberg, S. (2003) A Comparison of String Metrics for Matching Names and Records. *Kdd Workshop on Data Cleaning and Object Consolidation*, **3**.

[16] Jesse, D. and Goadrich, M. (2006) The Relationship between Precision-Recall and ROC Curves. *Proceedings of the* 23*rd International Conference on Machine Learning*, Pittsburgh, 25-29 June 2006, 233-240. https://doi.org/10.1145/1143844.1143874

[17] Breiman, L. (1996) Bagging Predictors. *Machine Learning*, **24**, 123-140. https://doi.org/10.1007/BF00058655

[18] Freund, Y., *et al.* (1997) Using and Combining Predictors That Specialize. *Proceedings of the* 29*th Annual ACM Symposium on Theory of Computing*, El Paso, 4-6 May 1997, 334-343. https://doi.org/10.1145/258533.258616

[19] Friedman, N., Geiger, D. and Goldszmidt, M. (1997) Bayesian Network Classifiers. *Machine Learning*, **29**, 131-163.

[20] Chen, T. and Guestrin, C. (2016) Xgboost: A Scalable Tree Boosting System.

[21] Bergstra, J., *et al.* (2006) Aggregate Features and AdaBoost for Music Classification. *Machine Learning*, **65**, 473-484.

[22] Wolpert, D.H. (1992) Stacked Generalization. *Neural Networks*, **5**, 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

[23] Ham, J., *et al.* (2005) Investigation of the Random Forest Framework for Classification of Hyperspectral Data. *IEEE Transactions on Geoscience and Remote Sensing*, **43**, 492-501. https://doi.org/10.1109/TGRS.2004.842481

[24] Atkinson, E.J., *et al.* (2012) Assessing Fracture Risk Using Gradient Boosting Machine (GBM) Models. *Journal of Bone and Mineral Research*, **27**, 1397-1404. https://doi.org/10.1002/jbmr.1577

[25] Ghorbani, A.A. and Owrangh, K. (2001) Stacked Generalization in Neural Networks: Generalization on Statistically Neutral Problems. *Proceedings of the International Joint Conference Neural Networks*, Washington DC, 15-19 July 2001, 1715-1720.

[26] Wang, S.-Q., Yang, J. and Chou, K.-C. (2006) Using Stacked Generalization to Pre-

dict Membrane Protein Types Based on Pseudo-Amino Acid Composition. *Journal of Theoretical Biology*, **242**, 941-946. https://doi.org/10.1016/j.jtbi.2006.05.006

[27] Tsymbal, A., *et al.* (2008) Dynamic Integration of Classifiers for Handling Concept Drift. *Information Fusion*, **9**, 56-68. https://doi.org/10.1016/j.inffus.2006.11.002

[28] Bhatnagar, V., *et al.* (2014) Accuracy-Diversity Based Pruning of Classifier Ensembles. *Progress in Artificial Intelligence*, **2**, 97-111.

[29] Giacinto, G. and Roli, F. (2001) Dynamic Classifier Selection Based on Multiple Classifier Behaviour. *Pattern Recognition*, **34**, 1879-1881. https://doi.org/10.1016/S0031-3203(00)00150-3

[30] Liu, G., *et al.* (2015) Report for Repeated Buyer Prediction Competition by Team 9*STAR*. *Proceedings of the* 1*st International Workshop on Social Influence Analysis Co-Located with* 24*th International Joint Conference on Artificial Intelligence* (*IJCAI* 2015), Buenos Aires, 27 July 2015.

[31] He, B., *et al.* (2015) Repeat Buyers Prediction after Sales Promotion for Tmall Platform. *Proceedings of the* 1*st International Workshop on Social Influence Analysis co-located with* 24*th International Joint Conference on Artificial Intelligence* (*IJCAI* 2015), Buenos Aires, Argentina, 27 July 2015.

[32] Domingos, P. and Pazzani, M. (1997) On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, **29**, 103-130.

[33] Schutt, R. and O'Neil, C. (2013) Doing Data Science: Straight Talk from the Frontline. O'Reilly Media, Inc.