Incorporating Feature Selection in the Improved Stacking Algorithm for Online Learning Analysis and Prediction

Hong Dai, Wenkai Wu, Jiacheng Li, Yangke Yuan

***Abstract*—Online learning is becoming a common learning method in the field of education. The correct classification of online learners plays a vital role in solving the key issues such as low pass rates and high dropout rates. In this paper, we propose a improved ensemble algorithm for classifying learners, which integrates feature selection and the improved Stacking algorithm (Stacking-PMLR). One feature selection algorithm is Mean Decrease Impurity Algorithm based on Random Forest. It is used to investigate the learning behavior factors which contribute to class of learner. It is also used to select the most frequent features and to reduce the dimensions. Analyzing learners' behavior features by the feature selection algorithm, we know that a number of chapters, interaction days, interactions times and video viewed times are the most important factors. Learners' behavior features from feature selection are used as the attribute input of Stacking-PMLR for classifying learners. After that, we use the multilevel improved ensemble algorithm Stacking-PMLR to classify learners. We improve the Stacking algorithms in terms of its hierarchical structure, data features representation, combination strategy and classification algorithm according to its own characteristics. We use the improved Stacking algorithm to construct the classification model. In addition, fifteen real world different type datasets in UCI machine learning repository are applied. The experimental results show that the improved Stacking algorithm has better performance in accuracy, precision and $F_{1.}$ It also shows the feasibility of the Stacking-PMLR. Finally, we use feature selection and the Stacking-PMLR algorithm to classify the public dataset of the edX online learning platform. The experimental results show that the performance of Stacking-PMLR is better. It shows the practical value of the Stacking-PMLR in online learning prediction.**

***Index Terms*—Online learning, ensemble algorithm, feature selection, stacking algorithm**

I. Introduction

THE rapid advancement of Internet technology has promoted the rapid development of the education industry, and the education industry has entered the Internet age. Online learning is a new type of education method that uses the Internet for teaching and learning [1]. The online learning platform provides learners with a variety of high-quality course resources. Users can flexibly choose the study time and content according to their actual conditions [2]-[3]. With its abundant learning resources, convenient learning methods and huge number of learners, online learning has greatly enriched the existing education methods [4]-[5]. And it is of great significance for improving the scientific and cultural quality of the entire people [6]. Although online learning provides learners with a free and convenient way to learn, there are still many bottleneck problems behind them, such as low pass rates and high dropout rates [7]. In response to these bottlenecks that restrict the development of online learning, some researchers have begun to mine and analyze the learning behavior data that generated by learners in the process of online learning. By analyzing the learning behavior data, it provides some scientific basis for online learning platforms in the development of teaching strategies. It enables the online learning platforms to provide learners with efficient learning services and improve the quality of online learning [8]-[9].

Feature selection, one of the data mining technologies, is currently applied to discover the relationships among different learners' features in online learning big data [10]. In this paper, we use a feature selection algorithm (Mean Decrease Impurity Algorithm Based on Random Forest) to gain a clearer understanding of the learning behavior features for different classes of learners as well as improve accuracy of classifying. At the same time, different teaching schemes are implemented according to different types of learners so as to provide scientific basis for efficient teaching of online learning platforms.

The Stacking algorithm is one of the most widely used algorithms in ensemble learning algorithms. Its generalization ability has been greatly improved to compare to other ensemble algorithms. It has been widely used in various fields of industrial production. Similarly, it is often used in major well-known data competitions. According to the current research on the Stacking algorithm, it can be found that the Stacking algorithm has excellent performance by taking the prediction probabilities of the base classifiers as the input features of the meta-classifiers and using Multi-response Linear Regression (MLR) as the meta-classifiers [11]. The paper names it as PMLR and improves the Stacking algorithm based on it. We combine

Manuscipt received January 9, 2020; revised August 19, 2020. This work was supported in part by LiaoNing province natural science foundation guidance plan project under Grant 2019-ZD-0270.

Hong Dai, the corresponding author, is Professor of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing; Anshan China (tel:+086-18642268599; fax: 0412-5929818; e-mail: dear_red9@163.com)

Wenkai Wu is with School of Computer Science and Software Engineering, University of Science and Technology LiaoNing; Anshan, China (e-mail:18678418521@163.com)

Jiacheng Li is with School of Mapping and Geographical Science, Liaoning Technical University; Fuxin, China (e-mail: 2766828497@qq.com)

Yangke Yuan is with School of Computer Science and Software Engineering, University of Science and Technology LiaoNing; Anshan, China (e-mail: 1056741362@qq.com)

different types of ensemble algorithms to reform a new classifying algorithm (Stacking-PMLR). It is also the core of online learners classifying. The algorithm not only improves the consistency and the success rate of classifying, but also reduces the time and the cost of online learning [12].

II. Related Work

In recent years, there are an increasing number of studies that use feature selection algorithms and the stacking algorithms to mine big data. We have a brief literature review in terms of online learning, feature selection algorithm (Mean Decrease Impurity Algorithm Based on Random Forest) and the stacking algorithms in the field of big data.

Most of the scholars' research on online learning analysis initially focused on well-known large-scale online learning platforms such as edX and Coursera. For example, after building the edX platform, Harvard University and MIT have jointly created a data analysis tool called Insights to visually analyze the generated learning behavior data. In the meanwhile, the two universities jointly released part of the processed edX learning platform's learning behavior data for scholars around the world to study [13]-[14]. At present, most research work mainly focuses on two aspects. On the one hand, statistical analysis is made on the features of learning behavior in a single dimension or multiple dimensions to explore the relationship between various types of learning behavior features and learning effects [15]. For example, Hummel analyzed the log information that generated by learners during the platform access process and conducted learning interventions for learners based on the results of the analysis [16]. Laxmisha Rai analyzed various factors that affecting online learning from multiple perspectives based on features such as learners' types, degree of difficulty, learning motivation, learning environment and feedback information [17]. DeBoer analyzed personal information of learners to explore the impact on learning completion [18]. On the other hand, the model is constructed based on the learning behavior data of the learners so as to predict the learning effect and improve the learning efficiency [19]. For example, Ramesh used learners' different learning behavior features to predict learners' probability of taking the test and test scores [20]. Bart Pursel used learners' personal information features and learning behavior features to perform regression analysis with logistic regression method to find out the degree of influence on the learning effects [21]. Balakrishnan used hidden Markov models to predict learners' probability of dropping out of school by using significant learning behavior features in data from an online learning course at the University of Berkeley [22].

Feature selection algorithms have widely applied in data feature extraction of various fields. In the earlier research, feature selection algorithms generally used linear measures such as Mahalanobis distance and correlation coefficients [23]. Mitra et al. proposed two linear measures of minimum power error and maximum information compression index. They applied it to the unsupervised feature selection method and achieved good results [24]. For the non-linear relationship among data, researchers have proposed a variety of non-linear correlation metrics. Among them, metrics based on information theory are considered the most promising metrics. Information and conditional mutual information were applied to feature selection and achieved good results [25]-[26]. Aiming at the time complexity of feature selection, Koller et al. applied Markov blankets to feature selection for the first time [27]. Yu et al. proposed a fast feature selection algorithm based on correlation and defined the basic problems in feature selection [28].

Stacking algorithm has been widely used in different fields. It can construct corresponding Stacking algorithms for different problems. Moudrik et al. used the evolutionary non-linear stacking method in GO Player data [29]. Alvear et al. established a stacking noise reduction auto-encoding classifier for image classification problems [30]. Demir et al. improved Stacking algorithm on the generation, selection and combination of individual classifiers based on network intrusion data [31]. Abawajy et al. researched the Stacking algorithm that has large-scale automatic iterative multilevel classifiers specifically for big data. It greatly improved the classification accuracy [32]. Zhou et al. used the original data to train a neural network for ensemble and replaced the original labels with the predicted labels of the neural network to generate a new dataset [33]. Shunmugapriya et al. optimized the Stacking algorithm through artificial bee colony algorithm [34]. Chen et al. optimized the Stacking algorithm based on ant colony algorithm [35]-[36].

III. Methodology

We organize this section into four main subsections. The first subsection illustrates the feature selection algorithm. The following subsections introduce the theory of Stacking algorithm, the improved Stacking algorithm and the model construction of the improved Stacking algorithm, respectively.

A. Feature selection

The purpose of feature selection is to remove redundant or irrelevant features in the dataset by related methods to select the optimal feature subset, thereby improving the generalization ability of the prediction model and reducing its computational cost. At present, the methods for selecting feature subset include filter, wrapper and embedding. By analyzing the advantages and the disadvantages of the three feature selection methods and the characteristics of the Stacking algorithm, we comprehensively consider using the algorithm that based on the Random Forest (Mean Decrease Impurity Algorithm Based on Random Forest) in the embedded methods. The principle is to calculate the impure reduction value of each feature in the entire decision trees during the training of the decision trees. The mean value of the reduced impurity for each feature is used as feature selection metric. The measure of impurity is the Gini Index, which is defined as follows (formula 1):

$$Gini(D)=1-\sum_{k=1}^{|y|}p_k^2 \tag{1}$$

Where D is the dataset, y is the class label and p_k is the proportion of the k-th class sample in dataset *D*.

B. Stacking overview

Stacking algorithm is one of the most famous algorithms in ensemble learning algorithms [37]-[38]. And it plays an important role in the field of machine learning [39]. The working mechanism of the Stacking algorithm is to generate various base-level learners by training different base-level learning algorithms. Then the output datasets of the base-level learners are used as the training sets of the meta-level learning algorithms. The output of the trained meta-level learner is used as the final prediction result [40]-[41].

During the training of the Stacking algorithm, the input data of the meta-level learners is generated from the output data of the base-level learners, which will cause the risk of overfitting. Therefore, the Stacking algorithm usually uses cross-validation to reduce the risk of overfitting [42]-[43]. Assume that the T base-level learning algorithms are $\varsigma_1,\varsigma_2,...,\varsigma_T$, meta-level learning algorithms is ς . Firstly, the Stacking algorithm needs to divide the dataset into training dataset D and testing dataset D_{test} . Then the training dataset $D=\{(x_1,y_1),(x_2,y_2),...,(x_m,y_m)\}$ is randomly divided into k datasets similar size $D_1,D_2,...,D_k$, $D_i\cap D_j=\varnothing$, . Set D_j and $\bar{D}_j=D\backslash D_j$ as the testing set and training set for the j-th cross-validation [44]. In the k-th iteration, $h_t^{(j)}$ is the base-level learner, which is generated by the t-th base-level learning algorithm ς_t using the j-th training set. $h_t^{(j)}(x_i)$ is the output of $h_t^{(j)}$'s prediction for each sample x_i on testing set D_j . When T base-level learners complete k iterations, we can get the output of sample x_i, that is $x_i'=(h_1^{(j)}(x_i),h_2^{(j)}(x_i),...,h_T^{(j)}(x_i))$. Accordingly, the training set of the meta-level learners generated by the base-level learners are $D'=\{(x_i',y_i)\}_{i=1}^m$. The meta-level learning algorithms are trained by D' to obtain the meta-level learners $h'=\varsigma(D')$. During the training phase of the base-level learners, each base-level classification algorithms ς_t will generate k base-level learners $h_t^{(j)}$ in k iterations, $j=1,...,k$. Therefore, in the testing phase of the base-level learners, k base-level learners will generate k output results $(h_t^{(1)}(x),h_t^{(2)}(x),...,h_t^{(k)}(x))$ on each sample x of the testing dataset D_{test} . The average outputs are described below (formula 2):

$$\bar{h}_t(x)=\sum_{j=1}^{k}h_t^{(j)}(x)/k \tag{2}$$

The average outputs are the testing set of the meta-level learners, and predicts it with the meta-level learners [45]-[46].

C. Improvement of the Stacking algorithm

For various ensemble learning algorithms, their differences lie in the diversity of individual learners, the different training data, their generation methods and the choices of individual combination strategies [47]. According to these features, the paper has made related improvements to the Stacking ensemble algorithm. The first is to process the input data of each level in the Stacking algorithm. The second is to redesign the hierarchical structure of the Stacking algorithm. The third is to construct the base-level learners in each level of the Stacking algorithm by using different ensemble learning algorithms. The fourth is the application of combining strategies.

The improved Stacking algorithm has three-level structure. The first and second levels are base-learner levels, named 0-level and 1-level. The third level is the meta-learner level, which is called 2-level.

i. *Processing of input features*

Assume the training dataset is $D=\{x_i,y_i\mid i=1,...,m\}$, $y_i\in\{c_1,...,c_l\}$, m is the number of samples and l is the number of classes. Generally, the structure of Stacking algorithm is divided into two levels. The first level is the base-learner level and the second level is the meta-learner level. In the base-learner level, T base learning algorithms are used to generate the training set D' of the meta-level learning algorithms on the training set D by k-fold cross-validation. The training set D' is shown as follows (formula 3):

$$\begin{aligned} D' &= \{(x_i',y_i)\mid i=1,...,m\} \\ &= \{z_{i1},z_{i2},...,z_{iT},\ y_i\mid i=1,...,m\} \end{aligned} \tag{3}$$

Meta-level learning algorithms input features x_i' are $(z_{i1},z_{i2},...,z_{iT}),z_{it}=(p^{h_t}(c_1\mid x_i),...,p^{h_t}(c_l\mid x_i))$, where $p^{h_t}(c_j\mid x_i)$ represents the probabilities that samples x_i are predicted to be class *j* by the t-th base-level learners. According to the input vectors of the meta-level learning algorithms, it can be known that the features dimensions of the meta-level learners' training set will increase significantly with the increase of the number of base-level learners. It will multiply the computation time and increase the distance of each sample in the training set of the meta-level learners. Assume that in the classification task of the Stacking algorithm, $p_t(x_1)$ and $p_t(x_2)$ are the class probabilities that predicted by the t-th base classifier for samples x_1 and x_2, respectively. The number of base classifiers is T. The training set features of the meta-level classifier are $P'(x_1)=(p_1(x_1),...,p_T(x_1),...,p_{T_1}(x_1))$, $P'(x_2)=(p_1(x_2),...,p_T(x_2),...,p_{T_1}(x_2))$. The Euclidean distance between them is given as follows (formula 4):

$$L(x_2,x_1)=\|P(x_2)-P(x_1)\|_2 \tag{4}$$

If the number of base-level learners is increased from T to T_1 , the training data features of the meta-level learners will also increase accordingly. That is $P'(x_1)=(p_1(x_1),...,p_T(x_1),...,p_{T_1}(x_1))$, $P'(x_2)=(p_1(x_2)...,p_T(x_2),...,p_{T_1}(x_2))$. The Euclidean distance between them is introduced below (formula 5):

$$L'(x_2,x_1)=\|P'(x_2)-P'(x_1)\|_2 \tag{5}$$

We can see clearly $L'(x_2,x_1)>L(x_2,x_1)$. Therefore, with the increasing number of base classifiers in the Stacking ensemble algorithm, the training data distribution of the meta-level learning algorithms will be sparser. At the same time, the dimensions will increase continuously, which will eventually lead to the algorithm complexity and training cost increasing exponentially. So we need to change the input features of the algorithm to get the reasonable and valid data samples.

Firstly, we process the data features of the 1-level classification algorithm. You can see from the beginning of the part $z_{it} = (p^{h_t}(c_1 \mid x_i), ..., p^{h_t}(c_l \mid x_i))$, where z_{it} is the class probabilities vectors that predicted by the t-th base classifier for sample x_i. $p^{h_t}(c_j \mid x_i)$ is the probability that it belongs to class j. The training set of the meta-level learning algorithms in the Stacking algorithm consists of the class probabilities and class labels that predicted by the base classifiers. In other words, the training set of the meta-level learning algorithms is $\{(z_{i1}, ..., z_{iT}, y_i) \mid i = 1, ..., m\}$. The improvement made in the paper is to add the original samples x_i to the original training set. The improved Stacking algorithm will be used as follows (formula 6):

$$\{(x_i, z_{iT}, y_i) \mid i = 1, ..., m; t = 1, ..., T\} \tag{6}$$

Equation (6) is as the training dataset for the 1-level classification algorithm. The size of the changed dataset and features are $m \times T$ and $l + |x_i|$, respectively. $|x_i|$ is the dimension of x_i.

The Stacking algorithm is suitable for learning tasks with a large amount of data [48]. In order to increase the size of the sample data, the initial training set is first used to train the 0-level classification algorithms. Then the "nearest neighbor" is constructed for the original dataset that based on the output of the 0-level classifier. As a result, the number of datasets and data distribution density can be increased.

In the Stacking algorithm, the class probability of each classifier predicted by the same sample *x* will be different. That means $(p^{h_t}(c_1 \mid x), ..., p^{h_t}(c_l \mid x))$, $t = 1, ..., T$ are different. If a single classifier is used to predict the class of a sample and its outputs are the class probabilities, in that way, the prediction result of the classifier is the class with the maximum probability. Assume that *T* classifiers can accurately predict the class of sample *x*, then the class predicted by each classifier with the maximum probability is same as the real class of sample *x*. That is, the probability vectors of *T* classifiers have similar values for the corresponding elements in the same dimension. Hence the features (x_i, z_{it}), $t = 1, ..., T$, in the training data $((x_i, z_{it}), y_i)$ of 1-level generated from the same sample x_i are relatively similar to each other (based on the Euclidean distance). If we extend the sample x_i in the original dataset D, that is $\%_i = (x_i, c_{i0}, ..., c_{ij}, ..., c_{il})$, $c_{ij} = 1$ represents the class of x_i is c_j, otherwise $c_{ij} = 0$. Then there is a large similarity between the training set $(x_i, p^{h_t}(c_1 \mid x), ..., p^{h_t}(c_l \mid x))$, $t = 1, . T.$ of the 1-level and $(x_i, c_{i0}, ..., c_{ij}, ..., c_{il})$. Since the L_1-norm of each class probability vector is 1, that is $\sum_{j=1}^{l} p^{h_t}(c_j \mid x) = 1$ and $0 \le p^{h_t}(c_j|x) \le 1$. Therefore, the Euclidean distance between the training set $(x_i, p^{h_t}(c_1 \mid x), ..., p^{h_t}(c_l \mid x))$, $t = 1, ..., T$ of T classifiers in the 1-level are mutually less than $\sqrt{l}$. And the Euclidean distance between them and $\%_i = (x_i, c_{i0}, ..., c_{ij}, ..., c_{il})$ is also less than $\sqrt{l}$. The training set of the *T* classifiers can be used as *T* "nearest neighbors" related to $\%$.

Constructing "neighbors" of the original dataset can give the Stacking algorithm several advantages. The first is that the Euclidean distance between samples cannot exceed $\sqrt{l}$ as the number of base classifiers increases. The second is that the training data characteristics dimensions of the 1-level learners are still $l + |x_i|$. $|x_i|$ is the dimension of x_i. It allows the computational cost of the algorithm not to increase significantly. The third is that the predicted probabilities of all classifiers can be retained in the 1-level structure instead of the values after processing them.

Similarly, the features of the original samples are also added to the meta-level, which allow various hidden relationships between the features of the original data and their class probabilities to be preserved.

ii. *Structural hierarchy design*

In the elaboration of the features attributes of the input data, we made relevant assumptions. Assume that all classifiers can correctly distinguish the class labels of each sample. However, in actual learning tasks, the individual classifiers of the ensemble learning algorithms cannot make correct class predictions on all samples. Therefore, there is noisy data in the training data of formula (6). In order to reduce the impact of noise data on the final classification effect, we reform the hierarchical structure based on the two-level structure of the general Stacking algorithm. A base-learner level is added between the base-learner level and the meta-learner level, which is the 1-level named in the opening part of this section. Retrain the data of formula (6) by using 1-level learners. The final results are averaged according to the outputs of the 1-level learners to achieve the effect of reducing noises.

iii. *Construction of the base classifiers*

The base classifiers used in the Boosting algorithms and the Bagging algorithms are usually traditional weak classifiers. These two types of ensemble algorithms can convert weak classifiers into strong classifiers [49]. However, the Stacking algorithm needs to combine learning results through different classification algorithms due to its different structural characteristics [50]. So the choice of classification algorithms is extremely important. According to the theoretical research of ensemble learning, individual learners need to have both diversity and accuracy. In the paper, three ensemble learning algorithms are selected as the base classification algorithms at 0-level and 1-level, namely Random Forest (RF), eXtreme Gradient Boosting Algorithm (XGBoost) and Gradient Boosting Decision Tree (GBDT) [51].

According to the current research, we can know that on the premise of ensuring accuracy of individual classifiers, the prediction diversity of each base classifier can improve the performance of ensemble learning to a certain extent [52]-[53]. The three base classification algorithms RF, XGBoost and GBDT used in the paper are all ensemble learning algorithms. In general, their performance is better than the traditional single algorithm. At the same time, the generation methods of the three ensemble algorithms are also different. They may increase the diversity of the base classifiers. The analysis of the three algorithms shows that the three algorithms have different emphases. Among them, RF pays more attention to reducing variance. XGBoost and GBDT can reduce both bias and variance. Therefore, the paper uses RF, XGBoost and GBDT as the base learning algorithms.

iv. *Application of combining strategies*

The current research on the combination strategies of ensemble learning algorithms shows that different combination strategies have great influence on the generalization ability of ensemble learning algorithms. Some researchers have found that the ensemble learning Stacking algorithm uses Multi-response Linear Regression (MLR) as meta-level learning algorithm and uses class probability as training data. It can achieve better performance.

MLR is a prediction algorithm based on linear regression. It is used as a classifier in a multi-classification problem. We set the number of class to *l*. The class labels are $\{c_1, c_2, ..., c_l\}$. The classification process becomes *l* binary classification problems. That is, for each class label c_l, a linear classifier LR_j, $j = 1, 2, ..., l$ is constructed. Use it to predict a binary variable for each class label c_l. When the class prediction is correct, the binary variable is 1 and vice versa. Firstly, we first need to calculate $LR_j(x)$ for all classes in order to classify a given unclassified sample x. Then the plurality voting method is used to predict the final class. The plurality voting method is shown as follows (formula 7):

$$H(x) = c_{\arg\max_j \sum_{i=1}^{T} h_i^j(x)} \qquad (7)$$

The class predicted by the plurality voting method is the one with the most votes in all class labels among them $h_i^j(x) \in \{0,1\}$. We only need to randomly select a class from them if there are multiple class labels that get the highest number of votes.

The improved Stacking algorithm in the paper used Multi-response Linear Regression (MLR) as the meta-level algorithm. The improved input data features of 1-level and 2-level make the testing sample data x generate T “nearest neighbor” samples after training on the classifiers of 0-level and 1-level. Finally, the MLR performs class prediction on the *T* "nearest neighbor" samples. And the final class is judged by the plurality voting method.

D. *Model construction of the improved Stacking algorithm*

By improving the Stacking algorithm in the previous section, the paper constructs an improved Stacking algorithm model. The structure of the model is divided into three levels. The first and the second levels are the base-learners levels. Meanwhile, RF, XGBoost and GBDT are used as the base learning algorithms. The third level is the meta-learner level and it uses MLR as the final classification algorithm. The paper names the improved Stacking algorithm as the Stacking-PMLR.

In order to clarify the improved Stacking algorithm, namely the Stacking-PMLR, more vividly and clearly, we use the specific input data dimensions and the number of classes to show the Stacking-PMLR algorithm in detail, as shown in Fig.1. Assume that the features dimensions of the data are 4, we need to perform a binary classification task and the number of classifiers in 0-level and 1-level is 3 respectively.

IV. Experiments

A. *Features selection of the edX dataset*

The edX dataset adopted in the paper has some features that are not related to the class prediction of the learners. Therefore, we use the method that decreases impurity based on the random forest to perform feature selection in the preprocessing so as to select the optimal features subset. Features selection experiment results shown in Table I.

According to the results of features selection experiments, the mean decrease Gini impurity of incomplete_flag, roles and course_id are 0.0002, 0.0002 and 0.0009, respectively. Three features have a weak impact on the class prediction of online learners, hence they are eliminated as redundant features. Finally, the improved Stacking algorithm (Stacking-PMLR) is used to perform learner class prediction experiments on the remaining 13 features.

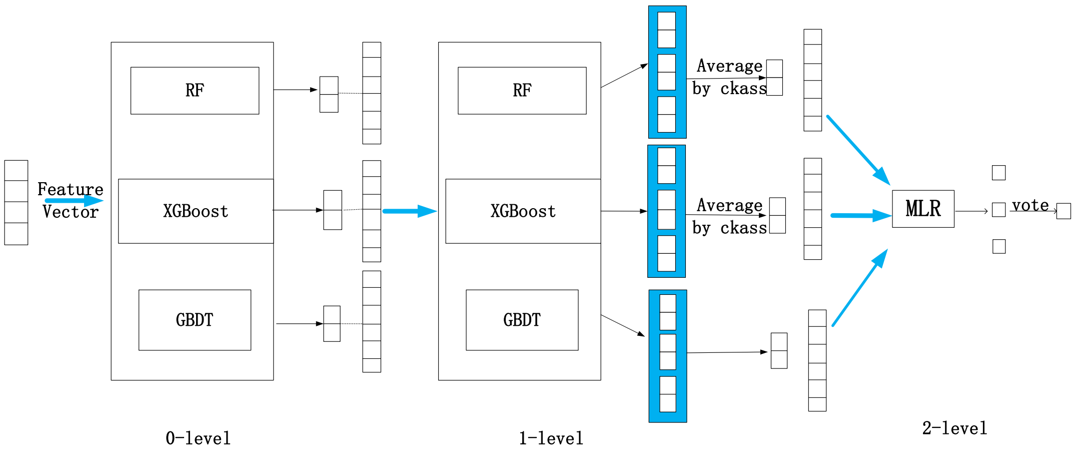

Fig. 1. Structure of the improved Stacking algorithm

B. Experiment design and results analysis of the Stacking-PMLR based on UCI datasets

i. *UCI datasets*

For the purpose of verifying the effectiveness of the improved algorithm in the paper, we use 15 different types of UCI datasets for experimental verification. UCI datasets include eleven unbalanced datasets and four balanced datasets. Table II illustrates the UCI datasets adopted in the experiment.

ii. *Experiment setups*

In order to evaluate the performance of the improved Stacking algorithm, the paper uses 6 algorithms for experimental comparison. The comparison algorithms are PMLR, Stacking-PMLR, ChooseBest, S01F, S3L and S1D. The input features of the PMLR's meta-level classifiers in the Stacking algorithm are the class probabilities distributions and the learning algorithms of the meta-level used multi-response linear regression (MLR). The Stacking-PMLR algorithm is the improved Stacking algorithm in the paper. ChooseBest follows the next key points. The best algorithm selected from the base learning algorithms RF, XGBoost and GBDT, we selected the XGBoost. S01F represents an algorithm that removes the original features attributes from the input features of 1-level and 2-level in Stacking-PMLR. S3L lies in an algorithm that added the same hierarchy as 1-level to the Stacking-PMLR algorithm. S1D represents an algorithm that removed the 1-level from the Stacking-PMLR algorithm.

The experiments used 10-fold cross-validation methods 10 times to evaluate the performance of each algorithm. The random seeds used for each cross-validation are different. The final experimental results are averaged over 10 experimental results.

In the meantime, the base classifiers used in the six comparison algorithms are all the same. In order to evaluate the performance of the six algorithms comprehensively, we used the records of win, tie and lose (W\T\L) to measure each algorithm. Its meaning is the number of wins, ties and losses on the corresponding evaluation indicators of datasets for an algorithm compared to another algorithm.

In the part of the experiment, the number of base-level learners in the Stacking algorithm is 3 and 5, respectively. When we used 3 base classifiers, the number of classifiers for 0-level and 1-level in the Stacking-PMLR algorithm and S01F algorithm was 3, both of which were 1 RF, 1 GBDT and 1 XGBoost. S1D and PMLR algorithm used 6 base-level classifiers, which were 2 RF, 2 GBDT and 2 XGBoost. When we used 5 base classifiers, the number of classifiers for 0-level and 1-level in the Stacking-PMLR and S01F algorithm was 5, both of which were 2 RF, 2 GBDT and 1 XGBoost. S1D and PMLR algorithms used 6 base-level classifiers, which were 4 RF, 4 GBDT and 2 XGBoost.

iii. *Results and analysis*

In this section, we firstly analyze the W\T\L records of each algorithm. The results are shown as Table III and Table IV.

It can be seen from the data in Table III and Table IV that although the number of base-level classifiers of each ensemble algorithm is different, their experimental data shows the commonality.

TABLE Ⅰ
FEATURE SELECT RESULTS

Feature ID	Feature Name	Value Of Gini Impurity Decrease
1	grade	0.5927
2	nchapters	0.5573
3	ndays_act	0.4721
4	nevents	0.3917
5	nplay_video	0.2951
6	LoE_DI	0.2359
7	nforum_posts	0.1524
8	YoB	0.0653
9	gender	0.0284
10	final_cc_cname_DI	0.0057
11	start_time_DI	0.0031
12	last_event_DI	0.0026
13	userid_DI	0.0013
14	course_id	0.0009
15	roles	0.0002
16	incomplete_flag	0.0002

TABLE II
DATASET INFORMATION

Name of the dataset	Number of features	Number of classes	Number of data	Class distribution
transfusion	4	2	748	(177,571)
ionosphere	34	2	351	(225,126)
ecoli	7	8	336	(2,2,5,20,35,52,77,143)
arrhythmia	279	13	452	(2,3,4,5,9,13,15,15,22,25,44,50,245)
beast	9	2	699	(241,458)
pageblocks	10	5	5473	(28,88,115,329,4913)
wine	13	3	178	(48,59,71)
semeion	249	2	1593	(158,1435)
autompg	7	3	398	(70,79,249)
glass	9	6	214	(9,13,17,29,70,76)
pendigits	16	10	7493	(719,719,719,719,720,778,779,,780,780,780)
contraceptive	9	3	1473	(333,511,629)
iris	4	3	150	(50,50,50)
seedsdataset	7	3	210	(70,70,70)
segment	18	7	2310	(330,330,330,330,330,330,330)

Compared with ChooseBest, S01F, S3L and S1D, the PMLR algorithm wins significantly more datasets, which indicates that the PMLR algorithm performs better than the above four algorithms. And the W\T\L records of the improved Stacking algorithm (Stacking-PMLR) in the paper shows that the overall performance of the Stacking-PMLR is better than that of the other 5 algorithms in 15 UCI datasets. At the same time, W\T\L records 9\0\6 and 9\1\5 of the Stacking-PMLR algorithm and the PMLR algorithm in the two tables mentioned above shows that the Stacking-PMLR algorithm performs better. According to the records of the Stacking-PMLR algorithm and the S1D algorithm in the two tables, 11\0\4 and 10\1\4, we can see the importance of improving the algorithm's hierarchical structure, which has a great impact on the improvement of classification accuracy. In addition, on the basis of the records 13\1\1 and 14\0\1 of the Stacking-PMLR algorithm and the S01F algorithm, it can be found that it is necessary to add the original features attributes to the input features of the 1-level and 2-level. Meanwhile, when we evaluate the performance of the Stacking-PMLR algorithm, we need to compare it with ChooseBest, the best classifier that composes it.

It can be seen from the above two tables that the comparison records of the two algorithms are 10\1\4 and 15\0\0. It can be concluded from these data that the ensemble algorithms are correct. The six algorithms are compared in detail using various performance evaluation indicators such as accuracy, precision and F_1. The specific experimental data are shown in Table V, Table VI and Table VII, respectively.

It can be clearly seen from the accuracy in Table V that the Stacking-PMLR algorithm is more than 1% higher than the MLR algorithm in all 5 datasets, less accurate than the PMLR algorithm in the 2 datasets and similar in the 8 datasets.

TABLE III
W\T\L RECORD ON ACCURACY WHEN USED 3 BASE CLASSIFIERS

Algorithm Algorithm	PMLR	ChooseBest	S01F	S3L	S1D	Stacking-PMLR
PMLR	-	8\1\6	12\0\3	13\0\2	12\0\3	6\0\9
ChooseBest	8\1\6	-	12\0\3	13\0\2	9\0\6	4\1\10
S01F	3\0\12	3\0\12	-	12\0\3	5\0\10	1\1\13
S3L	2\0\13	2\0\13	3\0\12	-	3\0\12	1\0\14
S1D	3\0\12	6\0\9	10\0\5	12\0\3	-	4\0\11
Stacking-PMLR	9\0\6	10\1\4	13\1\1	14\0\1	11\0\4	-

TABLE IV
W\T\L RECORD ON ACCURACY WHEN USED 5 BASE CLASSIFIERS

Algorithm Algorithm	PMLR	ChooseBest	S01F	S3L	S1D	Stacking-PMLR
PMLR	-	9\0\6	12\0\3	12\0\1	11\1\3	5\1\9
ChooseBest	6\0\9	-	10\0\5	14\0\1	6\0\9	0\0\15
S01F	3\0\12	5\0\10	-	12\0\3	8\0\7	1\0\14
S3L	1\0\12	1\0\14	3\0\12	-	1\0\14	0\0\15
S1D	3\1\11	9\0\6	7\0\8	14\0\1	-	4\1\10
Stacking-PMLR	9\1\5	15\0\0	14\0\1	15\0\0	10\1\4	-

TABLE V
ACCURACY (%) COMPARISON OF ENSEMBLE ALGORITHMS USING 5 BASE CLASSIFIERS

Algorithm Dataset	ChooseBest	PMLR	S3L	S1D	S01F	**Stacking-PMLR**	Average
transfusion	78.3	81.56	59.52	**81.38**	81.06	81.24	76.36
ionosphere	92.2	92.54	85.53	89.78	91.8	**92.7**	90.37
ecoli	83.05	**85.04**	68.65	84.54	80.79	83.92	80.41
arrhythmia	75.78	74.99	70.02	71.46	74.8	**77.55**	73.41
beast	95.49	94.62	93.07	95.5	96.01	**96.34**	94.94
pageblocks	83.69	82.65	72.15	81.41	80.72	**85.36**	80.12
wine	96.61	**98.36**	96.16	**98.36**	98.2	**98.36**	97.54
semeion	96.11	94.85	96.37	94.73	95.58	**96.67**	95.53
autompg	96.92	96.78	74.85	96.67	96.7	**96.94**	92.38
glass	65.7	68.2	52.58	55.46	64.01	**69.44**	61.19
pendigits	98.41	**98.99**	96.79	98.57	98.35	98.58	98.22
contraceptive	61.43	63.2	58.32	**63.47**	57.42	62.66	60.77
iris	93.5	**94.83**	91.5	94.17	94.5	93.71	93.70
seeds_dataset	91.66	92.83	90.45	92.2	90.93	**93.15**	91.61
segment	96.56	96.34	95.47	96.64	97.21	**97.25**	96.44

The Stacking-PMLR algorithm has lower accuracy than PMLR on both the contraceptive and the pendigits datasets, which are 0.44% and 0.41% lower respectively. The other datasets are higher than PMLR. Meanwhile, the comparison of the accuracy in Table IV shows that the improved Stacking algorithm performs better on the unbalanced datasets than the Stacking algorithm before the improvement. The Stacking-PMLR algorithm is more than 1% higher than the PMLR algorithm in the 4 datasets in Table VI. At the same time, their precision is similar in the 5 datasets. In these datasets, only the wine dataset is equal, and in the remaining datasets, the Stacking-PMLR algorithm is higher.

In terms of F_1, the Stacking-PMLR algorithm performs better than the SMLR algorithm in the 7 datasets, all of which are at least 1% higher than the PMLR algorithm. Moreover, the Stacking-PMLR algorithm is slightly worse than the PMLR in the 2 datasets. The performance of the other datasets is equivalent. The F_1 of the Stacking-PMLR algorithm in the transfusion dataset is 10.06% higher than the PMLR. It increased by 6.91% in the arrhythmia dataset and 6.50% in the semeion dataset. The data in these datasets belong to unbalanced data. The column "average" is the average of the other five algorithms compared on accuracy, precision and F1, respectively. The improved algorithm is higher than the average value of the other five algorithms in accuracy, precision and F1 evaluation indicators respectively. Meanwhile, the improved Stacking-PMLR algorithm has the most number of evaluation indicators in 15 datasets.

TABLE VI
PRECISION (%) COMPARISON OF ENSEMBLE ALGORITHMS USING 5 BASE CLASSIFIERS

Algorithm Dataset	ChooseBest	PMLR	S3L	S1D	S01F	**Stacking-PMLR**	Average
transfusion	45.67	61.86	63.08	**72.92**	42.7	56.64	57.25
ionosphere	92.62	**93.15**	85.28	92.99	91.4	92.96	91.09
ecoli	61.89	**63.67**	52.59	61.06	57.02	62.63	59.25
arrhythmia	73.36	69.13	64.69	64.15	62.67	**76.45**	66.80
beast	96.57	95.74	92.81	97.39	95.49	**98.26**	95.60
pageblocks	80.04	77.25	66.56	74.86	73.98	**80.36**	74.54
wine	97.34	**98.44**	96.27	**98.44**	98.43	**98.44**	97.78
semeion	96.7	97.3	95.8	**97.46**	96.48	97.37	96.75
autompg	88.09	87.25	63.94	87.7	**88.5**	86.15	83.10
glass	43.39	**43.46**	41.39	26.64	32.49	39.91	37.47
pendigits	97.89	**99.1**	97.04	98.67	98.46	98.69	98.23
contraceptive	59.23	61.3	57.68	**63.31**	53.6	61.34	59.02
iris	93.86	94.91	93.56	91.9	94.99	**96.56**	93.84
seeds_dataset	92.46	92.93	91.6	91.66	91.88	**93.25**	92.11
segment	96.7	97.3	95.8	**97.46**	96.48	97.37	96.75

TABLE VII
F_1 (%) COMPARISON OF ENSEMBLE ALGORITHMS USING 5 BASE CLASSIFIERS

Algorithm Dataset	ChooseBest	PMLR	S3L	S1D	S01F	**Stacking-PMLR**	Average
transfusion	35.2	35.9	**57.95**	55.48	47.94	45.96	46.494
ionosphere	91.48	91.89	84.93	**93.77**	91.3	92.17	90.674
ecoli	59.92	**62.14**	59.82	60.84	60.62	61.39	60.668
arrhythmia	71.24	66.37	62.4	60.93	71.81	**73.28**	66.55
beast	96.64	96	93	96.61	95.88	**97.26**	95.626
pageblocks	75.72	75.97	76.51	73.93	75.95	**79.44**	75.616
wine	97.5	**98.6**	96.42	**98.6**	98.59	**98.6**	97.942
semeion	79.15	76.62	**86.9**	86.55	86.69	83.12	83.182
autompg	85.34	83.25	64.25	82.21	81.69	**85.75**	79.348
glass	36.66	36.81	**40.15**	26.8	28.14	36.03	33.712
pendigits	97.8	**99.2**	97.13	98.77	98.56	98.79	98.292
contraceptive	59.02	60.11	57.17	**63.05**	45.51	60.65	56.972
iris	92.98	95.01	91.45	89.46	93.64	**96.35**	92.508
seeds_dataset	92.56	93.03	90.48	91.76	91.09	**93.35**	91.784
segment	96.74	96.71	95.64	**97.54**	96.54	97.45	96.634

According to the experimental data of the three evaluation indicators, it can be found that if the improved Stacking algorithm removes the original feature attributes of the input data in 1-level and 2-level, the classification performance will be reduced as a whole. When the 1-level of the Stacking-PMLR algorithm is removed, the classification performance degrades over multiple datasets. If another base classifier level is added to the Stacking-PMLR algorithm, the classification performance will decrease in general. Meanwhile, the classification performance of the Stacking-PMLR algorithm is better than the PMLR algorithm and the XGBoost algorithm. Furthermore, the Stacking-PMLR algorithm performs better on unbalanced data.

C. *Experiment design and results analysis of the Stacking-PMLR based on edX dataset*

i. *edX dataset*

In this section, we selected 13 features in the edX dataset by feature selection algorithm to verify the performance of the Stacking-PMLR, PMLR, XGBoost, RF and GBDT. The improved Stacking-PMLR algorithm is used to predict classes of learners in the edX online learning behavior dataset, thereby providing scientific basis for the online learning platform to formulate different types of teaching strategies. The original features of the edX dataset are shown in Table VIII.

TABLE VIII
ORIGINAL FEATURES OF THE edX DATASET

Explain / Feature	Type of data	Description of the feature
course_id	string	course and semester
userid_DI	string	user ID
registered	numeric	whether the user is registered
viewed	numeric	only access after registration
explored	numeric	whether the course study exceeds 50%
certified	numeric	whether the user gets a certificate
final_cc_cname_DI	string	nationality of user
LoE_DI	string	educational level of users
YoB	numeric	user's birth year
gender	string	user's gender
grade	numeric	final course grade
start_time_DI	string	user's course registration date
last_event_DI	string	the date of the user's last interaction
nevents	numeric	number of interactive learning
ndays_act	numeric	days of interactive learning
nplay_video	numeric	video learning times
nchapters	numeric	number of chapters learned
nforum_posts	numeric	number of posts in the forum
roles	numeric	learner’s role
incomplete_flag	numeric	whether the data is consistent

ii. *Experiment setups*

The improved Stacking algorithm adopts XGBoost, RF and GBDT algorithm as the base classification algorithms of 0-level and 1-level. At the same time, 3 and 5 base classifiers are used in the base classifier level for experimental verification. When 3 base classifiers are used, the 2 base classifiers levels of the Stacking-PMLR algorithm have 1 RF, 1 GBDT and 1 XGBoost, respectively. Moreover, PMLR uses 6 base classifiers, which are 2 RF, 2 GBDT and 2 XGBoost. The Stacking-PMLR algorithm uses 600 decision trees in order to compare the XGBoost, RF and GBDT algorithm. When 5 base classifiers are used, the 2 base classifiers levels of the Stacking-PMLR algorithm have 2 RF, 2 GBDT and 1 XGBoost, respectively. In addition, PMLR uses 10 base classifiers, which are 4 RF, 4 GBDT and 2 XGBoost. The Stacking-PMLR algorithm uses 1000 decision trees in order to compare the XGBoost, RF and GBDT algorithm. Furthermore, the base classifiers use 100 decision trees in the above two ensemble algorithms.

iii. *Results and analysis*

In this section, we use accuracy, precision and F_1 for comparison. The results and the specific comparative analysis are shown in Table IX, Table X and Fig.2 - Fig.4.

TABLE IX
COMPARISON DATA OF 3 ALGORITHMS

Indicators / Algorithm	Accuracy (%)	Precision (%)	F_1 (%)
RF	90.42	91.37	90.96
PMLR	91.84	91.82	91.75
XGBoost	92.58	92.41	92.33
GBDT	91.73	90.94	90.66
Stacking-PMLR	**93.88**	**94.26**	**93.29**

TABLE X
COMPARISON DATA OF 5 ALGORITHMS

Indicators / Algorithm	Accuracy (%)	Precision (%)	F_1 (%)
RF	92.35	92.77	91.63
PMLR	93.17	93.78	93.13
XGBoost	94.25	93.96	93.65
GBDT	92.57	92.14	92.07
Stacking-PMLR	**95.63**	**95.89**	**95.1**

According to the analysis of the evaluation indicators in Table IX, it can be known that the Stacking-PMLR algorithm used three base classifiers at the base classifier level has great performance in three metrics of accuracy, precision and F_1. And compared with the best performing PMLR algorithm and XGBoost algorithm among the other four algorithms, the Stacking-PMLR algorithm improves 2.04% and 1.3% in accuracy. It increases 2.44% and 1.85% in precision and increases 1.54% and 0.96% in F_1, respectively.

From the analysis of accuracy, precision and F_1 in Table X, it can be found that the Stacking-PMLR algorithm used 5 base classifiers at the 2 base classifier levels compared with the RF, GBDT, XGBoost and the PMLR algorithm. It has significant advantages in three metrics. And compared with the best performing PMLR algorithm and XGBoost algorithm among the other four algorithms, the Stacking-PMLR algorithm improves 2.46% and 1.38% in accuracy, respectively. It increases 2.11% and 1.93% in precision, respectively. In addition, the F_1 increases 1.97% and 1.45%, respectively. The comparative analysis of the experimental data can prove the actual availability of the improved Stacking algorithm in the edX dataset. The performance of the improved the Stacking-PMLR algorithm is verified by data comparison again. It proves the correctness of the improved algorithm in the paper.

We use different numbers of base classifiers in more detail and intuitively in order to compare the performance of ensemble algorithms. We compare different levels of accuracy, precision and F_1 evaluation indicators in the five ensemble algorithms. The specific comparative analysis is shown in Fig.2 - Fig.4.

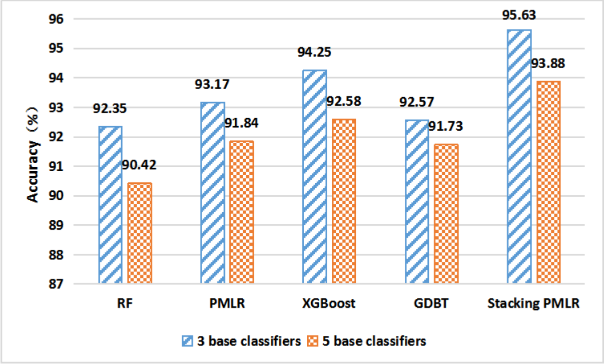

Fig.2. Accuracy (%) comparison

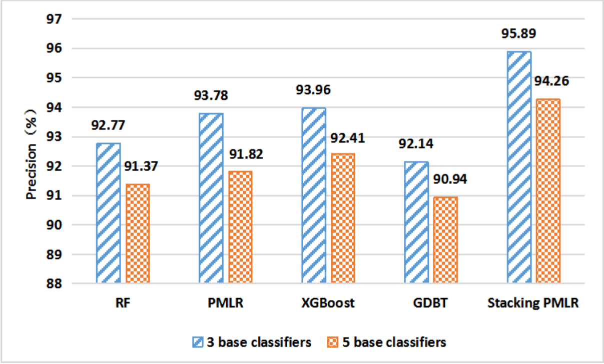

Fig.3. Precision (%) comparison

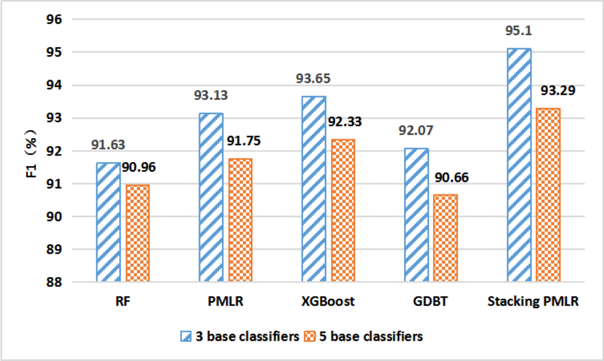

Fig.4. F_1 (%) comparison

The comparison of the three figures' experimental data shows that the Stacking-PMLR algorithm with 5 base classifiers has significant improvement in accuracy, precision and F_1 than the Stacking-PMLR algorithm with 3 base classifiers. Moreover, it can be found that the Stacking-PMLR algorithm using 3 base classifiers is more similar to the PMLR algorithm using 10 base classifiers and the XGBoost algorithm with 1000 decision trees in terms of accuracy, precision and F_1. There has no significant advantage. Therefore, the performance advantage of the Stacking-PMLR algorithm using 5 base classifiers in the edX dataset and the practical availability of the improved algorithm are proved.

V. Conclusion

The paper is corresponding improvements to the Stacking algorithm, including aspects such as feature selection, hierarchical structure, features of input data and combining strategies. The first is to apply the feature selection algorithm (Mean Decrease Impurity Algorithm Based on Random Forest) to select the optimal feature subset of edX dataset. The second is the hierarchical structure of the Stacking algorithm. A base-learner level is added to the two-level structure commonly used to reduce the noise contained in the output data of the first base-learner level. The following is to use the class probability distribution as the output of each level and add the original features to the input features of the second base-learner level and the meta-learner level. Finally, at the meta-learner level, Multi-response Linear Regression is used to learn the input data of the base-learner level to predict the class of sample data. The final class prediction is made by the voting method. By comparing with the PMLR algorithm, the necessity and correctness of the improved Stacking algorithm are verified. Meanwhile, the original features in 15 different types of UCI data sets and the algorithm with the best performance in the basic learning algorithm are compared with the improved algorithm. The advantages of the improved algorithm are verified. Finally, we use the feature selection algorithm and the improved Stacking algorithm to verify their actual effectiveness in the edX dataset.

The following research work of the paper needs to optimize the number and combination of different classifiers at various levels in the improved Stacking algorithm to maximize its performance. The data mining analysis method used in the paper is relatively simple. Therefore, more data mining analysis methods will be studied to build an efficient learning behavior data analysis model in future. In order to protect learners' privacy information, the dataset adopted in the paper is the dataset after data processing. Therefore, the number of datasets is reduced compared with the original datasets. The statistics of some datasets will be slightly affected.

References

[1] Boonlert Watjatrakul, "Online learning adoption: effects of perceived value and the moderating role of neuroticism", *In Proceedings of the 2019 4th International Conference on Distance Education and Learning*, Association for Computing Machinery, New York, NY, USA, pp.40–44, 2019, DOI:https://doi.org/10.1145/3338147.3338176

[2] Huang X., "Construction and application of online course teaching in intelligent learning environment", In: Xu Z., Parizi R., Hammoudeh M., Loyola-González O. (eds) Cyber Security Intelligence and Analytics, CSIA 2020, Advances in Intelligent Systems and Computing, Springer, Cham. vol.1146, pp.702-709, 2020.

[3] Ireri, B. and Wario, R., "An assessment of predictors of learner's attention and their influence to learner's engagement and learning outcomes in a mobile learning classroom", *In proceedings 2017 IST-Africa Week Conferences*, 2017.

[4] Farheen Hassan, Md. Khaled Amin, Tahsina Khan, Md. Mehedi Hasan Emon and Afrina Amin, "Roles of social influence in expediting online learning acceptance: a preliminary study on Bangladeshi learners", *In Proceedings of the International*

Conference on Computing Advancements, Association for Computing Machinery, New York, NY, USA, vol.31, pp.1–6, 2020, DOI:https://doi.org/10.1145/3377049.3377087

[5] Assami, S., Najima, D. and Ajhoun, R., "Ontology-based modeling for a personalized MOOC recommender system", In: Rocha, Á., Serrhini, M. (eds.) *EMENA-ISTL* 2018, SIST, Springer, Switzerland, vol. 111, pp. 21–28, 2019.

[6] Hesham Alomyan and Deborah Green, "Learning theories: implications for online learning design", *In Proceedings of the 2019 3rd International Conference on E-Society, E-Education and E-Technology,* Association for Computing Machinery, New York, NY, USA, pp.126–130, 2019, DOI:https://doi.org/10.1145/3355966.3358412

[7] Gavrilovic, N., Jovanovic, S. and Mishra, A., "Personalized learning system on student behaviour and learning style", *In proceeding of the 8th International Conference on eLearning*, Belgrade, Serbia, 2017.

[8] Meenakshi Sharma, Alka Dwivedi, Anita Sengar, and Manisha Solanki, "Implementing Innovative Online Teaching-learning Practice in Higher Education: Understanding Student Perspective", *In Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning*, Association for Computing Machinery, New York, NY, USA, pp.136–140, 2020, DOI:https://doi.org/10.1145/3377571.3377577

[9] Panchoo S. and Jaillet A., "Content analysis and learning analytics on interactions of unsupervised learners in an online learning environment", In: Serrhini M., Silva C., Aljahdali S. (eds) *Innovation in Information Systems and Technologies to Support Learning Research*. *EMENA-ISTL* 2019. Learning and Analytics in Intelligent Systems, Springer, Cham, vol. 7, 2019.

[10] Ling Zhong, Yantao Wei, Huang Yao, Wei Deng, Zhifeng Wang and Mingwen Tong, "Review of deep learning-based personalized learning recommendation", *In proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning,* Association for Computing Machinery, New York, NY, USA, pp.145–149, 2020, DOI:https://doi.org/10.1145/3377571.3377587

[11] Arroba, P., José L. Risco-Martín, Marina Zapater, José M. Moya and José L. Ayala, "Enhancing regression models for complex systems using evolutionary techniques for feature engineering", *Journal of Grid Computing*, vol.13, no.3, pp.409-423, 2015.

[12] Calvet Liñán, Laura, Juan Pérez and ángel Alejandro, "Educational data mining and learning analytics: differences, similarities, and time evolution", *International Journal of Educational Technology in Higher Education*, vol.12, no.3, pp.98-112, 2015.

[13] HarvardX Insights [Online]. Available: http://harvardx.harvard.edu/harvardx-insights

[14] Ho, A. D., Reich, J., Nesterko, S., Seaton, D. T. and Mullaney, T., et al., "HarvardX and MITx: the first year of open online courses, Fall 2012-Summer 2013", *Social Science Electronic Publishing*, 2014.

[15] Bari M. G., Salekin S. and Zhang J. "A robust and efficient feature selection algorithm for microarray data", *Molecular Informatics*, vol.36, no.4, 2017.

[16] Hummel K. A. and Hlavacs H., "Anytime, anywhere learning behavior using a web-based platform for a university lecture", *In proceedings of the SSGRR 2003 Winter Conference*, L'Aquila, Italy, 2003.

[17] Rai, L. and Chunrao, D., "Influencing factors of success and failure in MOOC and general analysis of learner behavior", *International Journal of Information and Education Technology*, vol.6, no.4, pp.262-268, 2016.

[18] DeBoer J., Stump G. S. and Seaton D., "Diversity in MOOC students' backgrounds and behaviors in relationship to performance in 6.002 x", *In proceedings of the sixth learning international networks consortium conference*, vol.4, pp.16-19, 2013.

[19] Hill P., "Emerging student patterns in MOOCs: a graphical view", Available: http://mfeldstein.com/emerging_student_patterns_in_moocs_graphical_view, 2013.

[20] Ramesh A., Goldwasser D. and Huang B., "Modeling learner engagement in MOOCs using probabilistic soft logic", *NIPS Workshop on Data Driven Education*, vol.21, pp.1-7, 2013.

[21] Pursel, B. K., Zhang, L. and Jablokow, K. W., "Understanding MOOC students: motivations and behaviours indicative of MOOC completion", *Journal of Computer Assisted Learning*, vol.32, no.3, pp.202-217, 2016.

[22] Balakrishnan G. and Coetzee D., "Predicting student retention in massive open online courses using hidden markov models", *Electrical Engineering and Computer Sciences University of California at Berkeley*, vol.53, pp.57-58, 2013.

[23] Guyon I., Elisseeff and André, "An introduction to variable and feature selection", *Journal of Machine Learning Research*, vol.3, no.6, pp.1157-1182, 2003.

[24] Mitra P., Murthy C. A. and Pal S. K., "Unsupervised feature selection using feature similarity", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.3, pp.301-312, 2002.

[25] Vinh L. T., Lee S. and Park Y. T., "A novel selection method based on normalized mutual feature nformation", *Applied Intelligence*, vol.37, no.1, pp.100-120, 2012.

[26] François Fleuret, "Fast binary feature selection with conditional mutual information", *Journal of Machine Learning Research*, vol.5, no.4941, pp.1531-1555, 2004.

[27] Koller D., "Toward optimal feature selection", *In proceedings of 13th International Conference on Machine Learning*, Morgan Kaufmann, vol.28, no.4, pp.184-292, 1996.

[28] Yu L. and Liu H., "Efficient feature selection via analysis of relevance and redundancy", *Journal of Machine Learning Research*, vol.5, no.12, pp.1205-1224, 2004.

[29] Moudrik J. and Neruda R., "Evolving non-linear stacking ensembles for prediction of go player attributes", *Computational Intelligence, 2015 IEEE Symposium*, pp.1673-1680, 2015.

[30] Alvear-Sandoval R.F. and Figueiras-Vidal A.R., "On building ensembles of stacked denoising auto-encoding classifiers and their further improvement", *Information Fusion*, vol.39, pp.41-52, 2017.

[31] Demir N. and Dalkılıç G., "Modified stacking ensemble approach to detect network intrusion", *Turkish Journal of Electrical Engineering & Computer Sciences*, vol.26, no.1, pp.418-433, 2018.

[32] Abawajy J.H, Kelarev A. and Chowdhury M., "Large iterative multitier ensemble classifiers for security of big data", *Emerging Topics in Computing IEEE Transactions on*, vol.2, no.3, pp.352-363, 2014.

[33] Zhou Z.H. and Jiang Y., "NeC4.5: neural ensemble based C4.5", *IEEE Transactions on Knowledge & Data Engineering*, vol.16, no.6, pp.770-773, 2004.

[34] Shunmugapriya P. and Kanmani S., "Optimization of stacking ensemble configurations through artificial bee colony algorithm", *Swarm & Evolutionary Computation*, vol.12, no.12, pp.24-32, 2013.

[35] Chen Y.J. and Man L.W. "An ant colony optimization approach for stacking ensemble", *In proceedings - 2010 2nd World Congress on Nature and Biologically Inspired Computing*, pp.146-151, 2010.

[36] Chen Y.J., Wong M.L. and Li H., "Applying ant colony optimization to configuring stacking ensembles for data mining", *Expert Systems with Applications*, vol.41, no.6, pp.2688-2702, 2014.

[37] Delibašić B., Radovanović S., Jovanović M., Bohanec M. and Suknović M., "Integrating knowledge from DEX hierarchies into a logistic regression stacking model for predicting ski injuries", *Journal of Decision Systems*, vol.27, pp.201-208, 2018, DOI:10.1080/12460125.2018.1460164

[38] Qunzhong Liu, Wei Luo and Tao Shi, "Classification method for imbalanced data set based on EKC stacking algorithm", *In Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, Association for Computing Machinery, New York, NY, USA, pp.51–56, 2019, DOI:https://doi.org/10.1145/3375998.3376002

[39] Chandrashekar G. and Sahin F., "A survey on feature selection methods", *Computers & Electrical Engineering*, vol.1, pp.16-28, 2014.

[40] Zhang S., Liu M. and Zhang J., "An academic achievement prediction model enhanced by stacking network", In: Zhai G., Zhou J., Yang H., An P., Yang X. (eds) *Digital TV and Wireless Multimedia Communication*, *IFTC* 2019, Communications in Computer and Information Science, Springer, Singapore, vol.1181, 2020.

[41] Malmasi, Shervin Dras and Mark, "Native language identification with classifier stacking and ensembles", *Computational Linguistics*, vol.44, no.3, pp.403–446, 2018, DOI:https://doi.org/10.1162/coli_a_00323

[42] Rasaq Otunba, Raimi A. Rufai and Jessica Lin, "Deep Stacked Ensemble Recommender", *In Proceedings of the 31st International Conference on Scientific and Statistical Database Management*, Association for Computing Machinery, New York, NY, USA, pp.197–201, 2019, DOI:https://doi.org/10.1145/3335783.3335809

[43] Alvear-Sandoval R.F. and Figueiras-Vidal A.R., "On building ensembles of stacked denoising auto-encoding classifiers and their further improvement", *Information Fusion*, vol.39, pp.41-52, 2019.

[44] Martin Gjoreski, Mitja Lustrek and Matjaz Gams, "Multi-Task Ensemble Learning for Affect Recognition", *In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers,* Association for Computing Machinery, New York, NY, USA, pp.553–558, 2019, DOI:https://doi.org/10.1145/3267305.3267308

[45] Yang Y. and Liu X., “A robust semi-supervised learning approach via mixture of label information”, *Pattern Recognition Letters*, vol.68, pp. 15-21, 2015.

[46] Choubin B., Darabi H. and Rahmati O., “River suspended sediment modeling using the CART model: a comparative study of machine learning techniques”, *Science of the Total Environment*, pp.272-281, 2018.

[47] Datao You, Xiangyu Yao, Xudong Geng, Xuyang Fang and Shenming Qu, “Stock index prediction method based on dynamic weighted ensemble learning”, *In Proceedings of the 2019 International Conference on Robotics Systems and Vehicle Technology*, Association for Computing Machinery, New York, NY, USA, pp.41–46, 2019, DOI:https://doi.org/10.1145/3366715.3366727

[48] Shamshirband S., Nodoushan E. J. and Adolf J. E., “Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters”, *Engineering Applications of Computational Fluid Mechanics*, vol.13, no.1, pp. 91-101, 2019.

[49] Pham B. T., Prakash I. and Singh S. K., “Landslide susceptibility modeling using reduced error pruning trees and different ensemble techniques: hybrid machine learning approaches”, *Catena*, pp.203-218, 2019.

[50] Jidong Duan, Kun Ma and Runyuan Sun, “Unbalanced data sentiment classification method based on ensemble learning”, *In Proceedings of the 2nd International Conference on Big Data Technologies* , Association for Computing Machinery, New York, NY, USA, pp.34–38, 2019, DOI:https://doi.org/10.1145/3358528.3358597

[51] Wang Q., Xu M. and Hussain A., “Large-scale ensemble model for customer churn prediction in search ads”, *Cognitive Computation*, vol.11, no.2, pp. 262-270, 2019.

[52] Asif Ahmed Neloy, H. M. Sadman Haque and Md. Mahmud Ul Islam, “Ensemble learning based rental apartment price prediction model by categorical features factoring”, *In Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, Association for Computing Machinery, New York, NY, USA, pp.350–356, 2019, DOI:https://doi.org/10.1145/3318299.3318377

[53] Chen T. and Guestrin C., “Xgboost: a scalable tree boosting system”, *In proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp.785-794, 2016.

Hong Dai (LiaoNing province of China, 1975-01), Master of computer application technology in 2005, graduated from Anshan University of science and technology, LiaoNing, China. Her research interests include computer network and network security. She is working at University of Science and Technology LiaoNing.