An Improved Stacking Performance Classification Prediction Model in Blended Teaching of Advanced Mathematics

Xiaoxu Xia, Meixue Liu*, Si Yuan, Lingna Li, Zhouyu Deng
School of Sciences, Southwest Petroleum University, Chengdu, Sichuan, China
**Corresponding Author*

Abstract: Examination results not only directly reflect students' learning outcomes but also indirectly indicate the effectiveness of teaching methods. However, traditional performance analysis primarily relies on summative evaluations of final exams, which are inherently delayed and insufficient to support individualized instruction. Consequently, performance prediction has become a prominent research focus in educational data mining. Students from six classes of the "Advanced Mathematics" course under a blended teaching model at a university are taken as research subjects in this study. A performance classification model based on hybrid sampling and an improved Stacking algorithm is proposed, which categorizes student performance into "Pass" or "Fail". Experimental results demonstrate that, compared with the traditional Stacking model(T-Stk), the improved model achieves increases of 0.43% in accuracy, 0.48% in precision, 0.23% in recall, and 1.57% in AUC.

Keywords: Performance Prediction; Stacking Fusion; Hybrid Sampling; Blended Teaching; Advanced Mathematics

1. Introduction

The steps of performance prediction are divided into data collection, data preprocessing, feature engineering, model establishment, model prediction, and model evaluation, six phases in total. The data sources for performance prediction are diverse. Harackiewicz et al. collected data on students' motivation, emotions, and interests during the study of the "Introduction to Psychology" course through questionnaire surveys, thereby achieving the goal of predicting students' performance [1]. Paulo and Alice predicted students' performance by collecting data on their educational background, family situation, social relationships, and historical grades, and ultimately, they found that the factor most influencing the current predicted performance was historical grades [2]. Amra et al. predicted students' performance by collecting basic information, educational background, parents' participation level, and social relationships of students, and the study concluded that social relationships were the key factor affecting study performance [3]. Zhang et al. collected students' behavioral characteristics, such as learning resources, forum exchanges, and homework tests, from an online learning platform and established a performance prediction model [4]. Younas [5], Ouajdouni [6], and Bossman [7] studied the relationship between students' satisfaction with online learning and academic performance during the COVID-19 period. Barnabás et al. analyzed the learning logs of teaching platforms and found that good learning performance was related to online learning duration, screen on-time, and sleep time [8]. From the perspective of data collection, most studies collect data from the basic social relationships of learners, classroom emotional feedback, and teaching platform records. These perspectives are single and lack multi-dimensional mixed scenarios of data. As for the feature engineering step, since the data for performance prediction is mostly numerical data, feature selection can be directly carried out. Zhang et al. employed principal component analysis to reduce the nine quantifiable indicators to three principal component factors. Consequently, two prediction models were developed. The initial model posited that the weighted sum of principal component factors was employed to obtain the prediction results. The second model involved the establishment of a logistic regression model. The results showed that both kinds of performance prediction models achieved good results [9]. From the perspective of model establishment and improvement, Qiu et al. proposed an online learning behavior classification model based on the online learning process - Process Behaviors Classification (PBC) model. The results showed that the PBC model outperformed traditional classification methods in terms of learning performance and prediction [10]. Hou et al. used an open dataset containing students' online learning behaviors to build a student learning behavior analysis model. The experimental results showed that students were divided into two learning types, namely the interactive learning type and autonomous learning type, and by combining the interaction between learning behaviors, the prediction ability of the model could be improved [11]. Existing experiments have shown that multiple prediction algorithms can achieve performance prediction, including classification prediction and regression prediction. Puarungroj et al. used the C4.5 decision tree algorithm to predict students' English grades [12]. Although the decision tree is simple and can visualize the classification process, it has the possibility of poor prediction performance for some multi-dimensional complex data. Wu et al. used the Big Five Personality Model as the main theoretical framework to explore the participation of students with different personality traits in online learning [13]. Ebiemi avoided using classification algorithms and instead employed clustering algorithms to group together low-performing students with similar attributes within a cluster. It identified the main attributes of these students and looked for individuals with those characteristics among future students, to provide precise assistance to students with similar traits [14]. Clustering can achieve the effect of grouping similar data points into the same cluster. Although the number of clusters can be specified, this approach results in a model with low accuracy, lacking the robustness and sustainability of classification models.

Based on the study of the above research, we found that most of the research did not change the original distribution of the data, which would lead to the false impression that the performance evaluation of the classification prediction model is generally good, while the prediction effect for the minority class is poor, yet the prediction effect for the majority class is good. Therefore, we adopt a new resampling method “Edited Nearest Neighbors (ENN) + Adaptive Synthetic Sampling (ADASYN)” to reconstruct the data distribution and balance the number of majority class samples and minority class samples. In addition, we use the Stacking integration framework to integrate multiple ensemble models and improve them based on the inherent defects of the learning framework, further enhancing the classification prediction performance of the final classification prediction model.

2. Performance Classification Prediction Model Based on Improved Stacking

2.1 Mixed Sampling Technique

In general, the number of learners with qualified grades is usually much larger than that of those with unqualified grades. Therefore, the collected data sets are mostly imbalanced datasets, which need to undergo resampling processing to alleviate the imbalance between the majority and minority classes. However, in most studies, no resampling is performed, and only a few studies perform Synthetic Minority Over-sampling Technique (SMOTE) over-sampling on the data sets without considering the impact of various over-sampling, under-sampling, and mixed sampling methods on the data sets. SMOTE over-sampling may generate a large amount of noise, resulting in a lower possibility of samples being judged as accurate during the prediction process, and further reducing the performance of the model on the test set.

ADASYN is superior to SMOTE oversampling to a certain extent. It is similar to the SMOTE oversampling idea, which means both methods are synthetic multiple minority class samples to achieve class balance. However, the difference between the two methods lies in the fact that the SMOTE algorithm requires the synthesis of new samples for each minority class sample, and the number of samples synthesized for each minority class sample is fixed. In contrast, the ADASYN algorithm whose greater emphasis on the minority class samples at the boundary, automatically adjusts based on the sparsity of the minority class samples, which means more new samples are generated for such minority samples. This approach can suppress the generation of noise in order to effectively focus on the sample regions with high classification difficulty.

The process of ADASYN is as follows: first, based on the number of samples of different classes in the dataset, the initial imbalance degree of the samples d can be calculated. The calculation formula is as Eq(1), m_1 where represents the number of majority class samples, and m_s represents the number of minority class samples.

$$d = \frac{m_1}{m_s} \tag{1}$$

The number of new minority class samples G that need to be synthesized in the entire sample set can be calculated as Eq(2). In Eq(2), the meanings of m_1 and m_s are not changed, just like the former one, β ($\beta \in [0,1]$)represents the balance target parameter. When $\beta = 1$ the ratio of samples of different classes after oversampling is 1:1.

$$G = (m_1 - m_s) * \beta \tag{2}$$

Then, we calculate the distribution density of each minority category sample. Compute the K-nearest neighbors for each minority class sample using Euclidean distance, and calculate the proportion of majority class samples among the K-nearest neighbors. The calculation formula is

$$r = \frac{a}{k}, r \in [0,1] \tag{3}$$

Where a represents the number of the majority class samples in the K-nearest neighbor samples. For each minority class sample x_i , the normalization of their r_i is performed to obtain λ_i , representing the contribution of each sample to the generation of new samples, as Eq(4):

$$\lambda_i = \frac{r_i}{\sum_{i=1}^{s} r_i} \tag{4}$$

When multiplying λ_i by G , the result is the new number of synthetic samples required for each minority class sample point, as Eq(5):

$$g_i = \lambda_i * G \tag{5}$$

Finally, we can select one minority class sample x_j from the K-nearest neighbors of x_i and generate a new sample x_{new} , as Eq(6). Repeat this operation until a certain number of minority class samples g_i is reached.

$| x_i - x_j |$ represents the Euclidean distance between x_i and x_j, and $rand(0,1)$ represents a random value within the range of [0, 1].

$$x_{new} = x_i + rand(0,1) * | x_i - x_j | \tag{6}$$

ENN is a method of undersampling. The basic idea is to remove a certain number of majority class samples according to certain rules to alleviate the imbalance between different classes. For each majority class sample x_i , if more than half of its K nearest neighbor samples do not belong to this class, then it is likely to be noise or an outlier at the boundary. During oversampling, this sample should be removed. If the number of samples of this class in the K nearest neighbor samples is less than half, it is considered safe and should be retained. The ENN algorithm can also set the proportion of majority class samples to be removed based on the real-time situation during the sampling process; therefore, this method has high flexibility and credibility.

The principle of mixed sampling involves first oversampling to increase the number of samples of the minority class, then undersampling to eliminate the noise or outliers that do not meet the requirements. This approach combines the advantages of both methods, and to a certain extent, compensates for their shortcomings. In practical applications, mixed sampling often yields better results than undersampling and oversampling. For ADASYN + ENN mixed sampling, it is necessary to first perform ADASYN oversampling and then perform ENN undersampling.

2.2 Stacking Classification Prediction Model

Stacking is an ensemble learning technique that combines multiple individual models to enhance the performance of the final model. The specific implementation method is to output the prediction results of multiple first-level base learners and use them as new features for the second-level meta-learner. The final prediction result is obtained through the meta-learner. Stacking fusion can also be regarded as obtaining the final prediction effect by weighting and integrating the prediction results of different types of base learners. The main steps of the improved Stacking model(I-Stk) ensemble fusion are as follows:

Step 1: Train the primary learners using the training set. To obtain multiple differentiable features that can be fused, it is usually necessary to train several primary learners with significant differences.

Step 2: Use the primary learners trained in the first step to make predictions on the test dataset, and use the predicted results of multiple primary learners as new features. Use the prediction results of the primary learners on the training set and the test set to construct new training and test sets as the input for the second-level meta-learner.

Step 3: Train the meta-learner using the new training set.

Step 4: Use the trained meta-learner to make predictions on the new test set and obtain the final prediction result on the test set.

By using the Stacking method to integrate other models, the prediction performance of the final score prediction model can be enhanced. However, the Stacking integration model has two drawbacks. Firstly, it does not input the original features into the subsequent learners, which may discard important information in the data; secondly, the T-Stk only has a two-layer structure, which may prevent it from having the opportunity to capture the complex relationships in the dataset. Further speaking, these two shortcomings may result in the prediction ability of the Stacking integration model not reaching the maximum. Therefore, in this paper, the Stacking method is used to integrate multiple single models, and improvements are made to address the above two major deficiencies of the T-Stk, hoping to further enhance the prediction performance of the model.

2.2.1 Traditional stacking model for performance prediction classification

The specific steps for the T-Stk ensemble learning model for academic performance prediction are as follows:

Step 1: Select Adaptive Boosting (AdaBoost) and eXtreme Gradient Boosting (XGBoost) as the primary learners for the first layer.

Step 2: Divide the training set (Train) into 10 equal parts, denoted as [Train1, Train2, ..., Train10].

Step 3: In the first-level prediction model, train the two primary learners using each of these 10 parts as the training set. For the AdaBoost primary learner: select one of Train1, Train2, ..., Train10 as the validation set and the remaining 9 as the training set. Perform ten-fold cross-validation in sequence, and successively obtain 10 validation set prediction results [a11, a12, ..., a110]. Merge them vertically to obtain the AdaBoost prediction result, denoted as A1. Use the trained AdaBoost model to predict the test set Test, obtaining 10 prediction results b11, b12, ..., b110, and average them to obtain the AdaBoost prediction result, denoted as B1.

Step 4: Train the two primary learners, AdaBoost and XGBoost, separately according to the process described in Step 3, completing the training and prediction, and obtaining two sets of prediction values, A1 and A2, for the training sets. Merge A1 and A2 horizontally and then input them as a new training set (A1, A2) into the secondary learner Light Gradient Boosting Machine (LightGBM) for training, obtaining the final prediction result of the Stacking model training set. Similarly, use the trained AdaBoost and XGBoost models according to the process described in Step 5 to predict the test set, obtaining two sets of prediction values, B1 and B2, and merge them horizontally as (B1, B2), which is used as the first-layer output result of the test set. Use (B1, B2) as the new feature of the test set and input it into the trained second-layer meta-learner LightGBM for prediction, obtaining the final prediction result of the test set, that is, whether the students can pass the final exam on the test set (1 indicates passing, 0 indicates failing).

2.2.2 Improved stacking model for performance prediction classification

The specific construction process of the three-layer Stacking model is as follows:

Step 1: Select AdaBoost and XGBoost as the first-level primary learners.

Step 2: Divide the training set (Train) into 10 equal parts, represented as [Train1, Train2,..., Train10].

Step 3: In the first-level prediction model, train the two primary learners using these 10 equal parts of the training set, respectively, obtaining the prediction results of the first-level primary learners on the training set (A11, A12). Take A11 as an example, where A represents the prediction value of the training set, the first 1 indicates the first primary learner in the first layer, and the second 1 indicates the first individual learner in the first layer. Use the trained AdaBoost and XGBoost models to test the test set, obtaining the prediction results as (B11, B12).

Step 4: Concatenate the original feature X row-wise with the output features of the first layer horizontally, then the input of the second-level primary learner's training set is (A11, A12, X), and the input of the test set is (B11, B12, X).

Step 5: Select AdaBoost and XGBoost as the second-level primary learners.

Step 6: Input (A11, A12, X) as the new training set to the second-level primary learner. In this layer, each learner still uses ten-fold cross-validation for training. The output of the second-level primary learner on the training set is (A21, A22). Use the trained AdaBoost and XGBoost to predict the test set (B11, B12, X), obtaining the prediction results as (B21, B22).

Step 7: Follow the same principle as Step 6 for feature concatenation. The new training set of the third-level meta-learner is input as (A21, A22, X), and the new test set is input as (B21, B22, X).

Step 8: Input (A21, A22, X) to the meta-learner LightGBM for training, obtaining the trained LightGBM model. Input (B21, B22, X) to the trained LightGBM model to obtain the final LightGBM prediction result, that is, whether the student can pass the exam on the test set (1 indicates passing, 0 indicates not passing).

3. Experiment and Results

3.1 Data Preparation

The subjects of this experiment were students from 6 teaching classes of the "Advanced Mathematics" course at a certain university, totaling 785 individuals. We adopted a blended teaching model combining online and offline methods in this course. A complete classroom cycle consists of three stages: pre-class, in-class, and post-class. The pre-class stage is online teaching, conducted on MOOC platforms. The teacher posts learning content and tasks on the MOOC platform in advance, and students complete the relevant content on the MOOC platform, including watching the teaching videos released on the MOOC platform, answering the questions in the videos, and completing the pre-class assignments and other preparatory tasks. The in-class stage is offline teaching, conducted face-to-face with students in the classroom. The teacher teaches the students the content they pre-studied on the MOOC platform and helps them build a complete knowledge system, answering students' difficult points. During offline teaching, the teacher issues a teaching sign-in task on the MOOC platform to record the attendance of students. At the same time, students complete classroom exercises posted on MOOC and Superstar learning platforms, and their mastery of knowledge points is monitored at any time. The post-class stage is the students' review stage. Students complete the post-class exercises and submit them through the Superstar Learning Platform. Regular unit tests and mid-term exams are conducted on the MOOC platform to consolidate the learned knowledge. The feature set of the data set is shown in Table 1.

Table 1. Prediction Characteristics for Blended Teaching Performance

	feature	meaning	source
Basic information	SID	Student ID	
Before Class	VIVD	Video viewing duration	MOOC
	ASPA	Average score of the preview assignments	Chaoxing Learning
	ATPA	Average submission time for preview assignments	Chaoxing Learning
During Class	ATR	attendance rate	MOOC
	QRP	Quick-response points	Chaoxing Learning
	NICD	Number of in-class discussions	MOOC and Chaoxing Learning
After Class	NED	Number of extracurricular discussions	Chaoxing Learning and QQ
	MTES	Mid-term exam scores	MOOC
	ASUT	The average score of the unit test	MOOC
	LSTP	Learning situation of task points	Chaoxing Learning
	ASOA	Average score of the offline assignments	Chaoxing Learning
	ATOA	Average submission time for offline assignments	Chaoxing Learning
	TPBP	Test paper bonus points	Paper document
	NTPA	The number of times students participate in Q&A sessions through both online and offline channels	Paper document
Goal	FC	final exam grade	Educational administration system

The data set was cleaned. Since these records included students who had retaken the course, and these students lacked their regular grades, the records of the retaking students were removed, and only the records containing complete information were retained, totaling 763. The training set (Train) and the test set (Test) were divided in a 7:3 ratio, and the training set and test set were standardized, respectively. Due to the different scales of the features, there was a significant difference in the calculation process, so the data set was standardized to remove the influence of scale and obtain clean data.

3.2 Data Resampling

From the table below, one can observe the distribution of samples before and after resampling. For details, please refer to Table 2. After dividing the dataset into training and testing sets, the ratio of passers to non-passers in the test set is approximately 10:1, and the same ratio exists in the original training set. This indicates an imbalance between the majority and minority class samples. The application of ADASYN oversampling to the training set resulted in a balanced 1:1 class distribution. However, after ENN undersampling, the number of samples in the passing and non-passing classes still shows a significant disparity, with a ratio of approximately 8:1. The ratio of samples after ADASYN + ENN processing is 1.35:1. Subsequently, the various resampled training sets will be used to train the models to identify which resampling method yields the best overall predictive performance.

Table 2. Data Distribution

.			
	Original	Original	ADASYN + ENN
Passing	207	484	482
Failure	22	50	355

3.3 Model Evaluations

In this study, the accuracy, precision, recall, F1-score, and AUC of the five models are computed and compared. The results for the three base classifiers (AdaBoost, XGBoost, LightGBM) and the two fusion models (T-Stk, I-Stk) are tabulated in Table 3. Furthermore, these metrics are visualized in Figure 1 using a bar chart, with the evaluation metrics of the models on the horizontal axis and the corresponding models on the vertical axis for clear performance assessment. From Figure 1, it can be seen that among all the models, I-Stk achieved the highest accuracy of 0.8777, while T-Stk achieved the second-highest accuracy of 0.8734. Overall, the accuracy of the Stacking fusion models is higher than that of the ensemble learning models. Comparing the precision of the models in the text, the precision of LightGBM is the highest, at 0.9609, followed by AdaBoost, with a precision of 0.9581. Overall, the precision of the ensemble learning models is higher than that of the Stacking model, but there is no significant improvement. The precision of the Stacking model also remains at a relatively good level, around 0.95. Comparing the recall of the models in the text, the recall rates of T-Stk and I-Stk are the same, at 0.9082, and both are higher than those of the other ensemble learning models. Comparing the F1-score of the models in the text, the fused model has a higher F1-score than the ensemble learning model. Among them, I-Stk achieved the highest F1-score of 0.9307. Overall, the F1-score of the Stacking model is higher than that of the ensemble learning model. Comparing the AUC values of the models in the text, the AUC value of the fused model is higher than that of the ensemble learning model. Among them, I-Stk achieved the highest AUC value of 0.8663, followed by T-Stk, at 0.8506.

In conclusion, the two improved methods proposed based on the T-Stk can improve the model performance to a certain extent. Compared to T-Stk, I-Stk maintains the same recall rate while improving accuracy, precision, F1-score, and AUC values by 0.43%, 0.48%, 0.23%, and 1.57%.

4. Conclusion

We collect multi-stage and multi-dimensional data from blended teaching classrooms, thereby providing a more comprehensive and objective prediction of students' final grades. For model development, two key improvements are proposed for the traditional Stacking algorithm. Firstly, original features are added when inputting the subsequent layer models. Secondly, the number of model layers is increased. Based on these two improvements, the traditional two-layer Stacking prediction model (T-Stk) is improved into a three-layer Stacking prediction model (I-Stk). The study concluded that both T-Stk and I-Stk exhibited superior prediction performance in comparison to the prediction performance of the ensemble learning model. Comparing the two Stacking fusion models, the overall prediction performance of I-Stk is superior to that of T-Stk. This indicates that adding original features and increasing the number of learner layers can effectively improve the prediction performance of the Stacking model. However, it is worth noting that the prediction ability of the added base learners cannot be too weak; otherwise, it may reduce the prediction ability of the original Stacking model. So, a three-layer Stacking model is constructed for final prediction. Both the first and second layers incorporate AdaBoost and XGBoost. LightGBM is then used in the third layer, and the original features are additionally included. This three-layer Stacking model is used as the final prediction model for students' "Advanced Mathematics" grades classification. Experiments demonstrate that this model achieves superior predictive performance compared to previous models.

Table 3. Evaluation Indicators of the Model

Model	Accuracy	Precision	Recall	F1-score	AUC
I-Stk	0.8777	0.9543	0.9082	0.9307	0.8663
T-Stk	0.8734	0.9495	0.9082	0.9284	0.8506
LightGBM	0.8166	0.9609	0.8309	0.8912	0.8474
XGBoost	0.8384	0.9570	0.8599	0.9059	0.8419
AdaBoost	0.8603	0.9581	0.8841	0.9196	0.8443

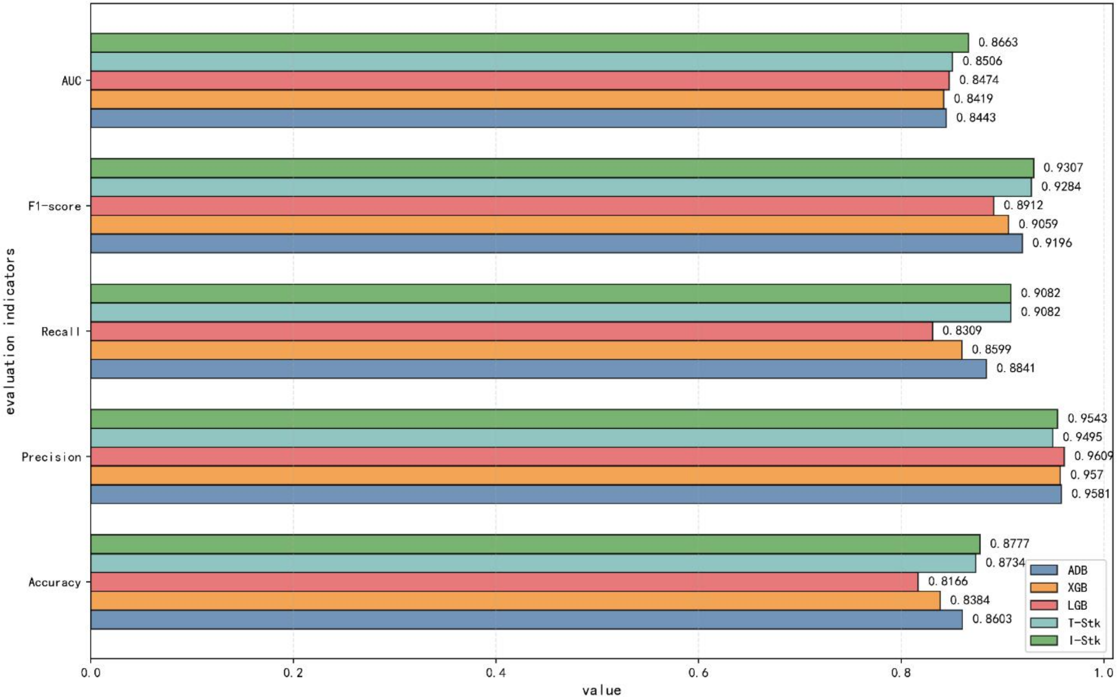

Figure 1. The Comparison Chart of Evaluation Indicators for Models

Acknowledgments

The authors are thankful to the esteemed reviewers for their valuable comments that significantly improved the quality of this article, and also extend gratitude for the project support. This work is supported by the projects "Research and Practice on the Reform of Blended Online and Offline Teaching in Advanced Mathematics Courses" [Grant No. 231101984035149], "Exploration and Practice of Teaching Reform in Public Mathematics Course Clusters Empowered by Digital Intelligence"[Grant No. JG2024-0512] and "Exploration of Reform and Innovation in University Mathematics Foundation Courses Based on Intelligent Technologies"[Grant No. GJS-2025-ZD-13].

References

[1] Harackiewicz J M, Durik A M, Barron K E, et al. The role of achievement goals in the development of interest: Reciprocal relations between achievement goals, interest, and performance. Journal of Educational Psychology, 2008, 100(1): 105.

[2] Cortez P, Silva A M G. Using data mining to predict secondary school student performance. 2008.

[3] Amra I A A, Maghari A Y A. Students performance prediction using KNN and

Naïve Bayesian//2017 8th international conference on information technology (ICIT). IEEE, 2017: 909-913.

[4] Zhang W, Zhou Y, Yi B. An interpretable online learner's performance prediction model based on learning analytics//Proceedings of the 11th International Conference on Education Technology and Computers. 2019: 148-154.

[5] Younas M, Noor U, Zhou X, et al. COVID-19, students satisfaction about e-learning and academic achievement: Mediating analysis of online influencing factors. Frontiers in psychology, 2022, 13: 948061.

[6] Ouajdouni A, Chafik K, Boubker O. Measuring e-learning systems success: Data from students of higher education institutions in Morocco. Data in Brief, 2021, 35: 106807.

[7] Bossman A, Agyei S K. Technology and instructor dimensions, e-learning satisfaction, and academic performance of distance students in Ghana. Heliyon, 2022, 8(4).

[8] Holicza B, Kiss A. Predicting and Comparing Students' Online and Offline Academic Performance Using Machine Learning Algorithms. Behavioral Sciences, 2023, 13(4): 289.

[9] Zhang W, Qin S, Yi B, et al. Study on learning effect prediction models based on principal component analysis in MOOCs. Cluster Computing, 2019, 22(6): 15347-15356.

[10]Qiu F, Zhang G, Sheng X, et al.Predicting students' performance in e-learning using learning process and behaviour data. Scientific Reports, 2022, 12(1): 453.

[11]Hou J, Wen Y. Prediction of learners' academic performance using factorization machine and decision tree//2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, 2019: 1-8.

[12]Puarungroj W, Boonsirisumpun N,Pongpatrakant P, et al. Application ofdata mining techniques for predicting student success in English exit exam//proceedings of the 12th international conference on ubiquitous information management and communication. 2018: 1-6.

[13]Wu R, Yu Z. Relationship between university students' personalities and e-learning engagement mediated by achievement emotions and adaptability. Education and Information Technologies, 2024, 29(9): 10821-10850.

[14]Ekubo E A. Attributes of low performing students in e-learning systemusing clustering technique. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2019, 5(3).