Research Article

Grade Prediction in Blended Learning Using Multisource Data

Ling-qing Chen ¹, Mei-ting Wu,¹ Li-fang Pan ^{2,3} and Ru-bin Zheng¹

¹School of Computer Engineering, Jimei University, Xiamen 361021, China

²School of Science, Jimei University, Xiamen 361021, China

³Digital Fujian Big Data Modeling and Intelligent Computing Institute, Jimei University, Xiamen 361021, China

Correspondence should be addressed to Li-fang Pan; xiaopan@jmu.edu.cn

Received 10 June 2021; Accepted 30 August 2021; Published 14 September 2021

Academic Editor: KunHong Liu

Copyright © 2021 Ling-qing Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Today, blended learning is widely carried out in many colleges. Different online learning platforms have accumulated a large number of fine granularity records of students' learning behavior, which provides us with an excellent opportunity to analyze students' learning behavior. In this paper, based on the behavior log data in four consecutive years of blended learning in a college's programming course, we propose a novel multiclassification frame to predict students' learning outcomes. First, the data obtained from diverse platforms, i.e., MOOC, Cnblogs, Programming Teaching Assistant (PTA) system, and Rain Classroom, are integrated and preprocessed. Second, a novel error-correcting output codes (ECOC) multiclassification framework, based on genetic algorithm (GA) and ternary bitwise calculator, is designed to effectively predict the grade levels of students by optimizing the code-matrix, feature subset, and binary classifiers of ECOC. Experimental results show that the proposed algorithm in this paper significantly outperforms other alternatives in predicting students' grades. In addition, the performance of the algorithm can be further improved by adding the grades of prerequisite courses.

1. Introduction

In recent years, online education platforms have increased with the rapid development of Internet technology; and the mixed teaching mode combining offline face-to-face teaching and online learning has also emerged. Mixed teaching, aided with computers, mobile phones, and other electronic equipment, guides students to carry out personalized and adaptive learning according to teachers' preset learning goals and driving tasks. Under the mixed teaching mode, students self-regulate the learning process, and teachers supervise students' learning progress as well as efficiency in real time. Integrating online evaluation with offline evaluation, the students' learning autonomy increases while learning efficiency is guaranteed. At the same time, combining with the analysis of students' online learning behavior trajectory and offline classroom performance, we can infer the students' phased learning situation to adjust the teaching pace in time and further improve the teaching quality. Therefore, it is worth studying how to transform students' learning data such as their online learning behavior

track and offline classroom performance into their phased learning situation and take corresponding measures to adjust and guide their personalized learning to achieve better teaching effect.

The structure of this paper is organized as follows. Section 2 reviews related literature. Section 3 describes the setting of the curriculum and data from the platforms. Section 4 provides a brief introduction to error-correcting output codes (ECOC) and genetic algorithms (GA) and then further illustrates the design and implementation of the TBCGA-ECOC (Ternary Bitwise Calculator based GA for ECOC) algorithm. Section 5 describes the experimental parameters and analyzes the experimental results of various algorithms. Section 6 summarizes the work.

2. Literature Review

Under the upsurge of mixed teaching, many scholars have conducted researches on related fields. Yousef and Sumner conducted detailed analysis and discussion on more than 200 articles on MOOC. They proposed possible future

research directions according to the main achievements and MOOC research direction in recent years [1]. For emerging ways of education big data, Ang et al. introduced and discussed six aspects including overview and classification of education big data, different data analysis methods, social challenges and technical challenges, technical problems, systematic data sources, data mining, and database [2]. To explore the underlying mechanisms behind the features, Xie conducted a survival analysis of the viewing duration of large-scale open online courses, in which two stochastic differential equations were used to describe the growth of two classes. The research helps in understanding the role of memory in the complexity of learning behavior [3]. Based on SPOC and flipped classes, Xu et al. also launched research on online platforms. They created experiments on the computer network mixed classes, stating that students' online behavior data can be used to predict their results and the results are more stable and reliable as classes advance [4]. Guided by Moore's trading distance, Hew et al. defined MOOC success as student satisfaction with the course [5]. They proposed a unique methodology framework (supervised machine learning and sentiment analysis) to examine user review generated datasets and identify specific learners level and course level factors. It can predict MOOC learners' satisfaction and estimate their relative impact. The research extended the theoretical understanding of the factors affecting learner MOOC satisfaction and presented specific opinions for the design of MOOC. To adjust the video curriculum and arrangement of MOOC better, Hu et al. put forward the analysis method of students' video viewing behavior based on the Spark platform and verified it with the data of the cauX platform. The experimental results show that the method can analyze the characteristics of the viewing behavior quickly and accurately [6]. Yu et al. have noted that SPL-based methods can be used to establish a practical MOOC learning analysis framework, which increased the complexity and reusability of learning analysis [7]. Zhang et al. constructed a SPOC-based flipped classroom teaching model based on the OBE concept and verified its effectiveness [8]. In order to build a better MOOC teaching platform and provide theoretical guidance and policy advice for college managers, Wang et al. combined the technology acceptance model (TAM) and plan behavior theory (TPB) to construct a theoretical model to analyze the mechanism behind MOOC learning performance [9]. It forms a scientific basis for the MOOC teaching setting and provides theoretical guidance and policy advice for college managers. Meanwhile Gardner and Brooks classified MOOC studies by predictors, predictions, and underlying theoretical models [10]. Then they critically investigated the work of each category and provided data on raw data sources, feature engineering, statistical models, evaluation methods, prediction architectures, and other aspects of these experiments, revealing several critical approach gaps and exploring different features and modeling technology spaces. The results make valuable contributions to construct accuracy, operability, and student success models for theoretical construction. A document clustering model based on weighted word embeddings was developed by Onan and Toçoğlu to

identify the post problem on MOOC forums [11]. Additionally, Toçoğlu and Onan designed a long short-term memory network (LSTM) which can classify around 70,000 MOOC reviews more accurately [12]. They also investigated the sentiment analysis of Twitter data and student evaluation data [13–15].

In addition to the research on MOOC data and teaching mode, many scholars focused on students' behavior models, researched student performance prediction, and explored factors affecting student study. Xu et al. developed a two-layer structure model consisting of multiple underlying predictors and cascading integrated predictors [16]. They also proposed a data-driven approach based on latent factor models and probability matrix decomposition to discover the curriculum relevance to construct more efficient underlying predictors. It is demonstrated that the proposed method performs the benchmark method. In addition, Ulloa-Cazarez et al. proposed genetic planning (GP) algorithm to predict whether students can pass the final exam for early warning of their learning status [17]. Xu and Yang designed classification algorithms based on support vector machine (SVM) to summarize students' learning motivation through their MOOC activity log to predict whether they can obtain a certificate [18]. Similar work was being done by Meier et al. [19]. They predicted possible student performance in subsequent learning through historical teaching data from the course and took instructional intervention when necessary. Qiu et al. proposed the latent dynamic factor graph (LadFG) to predict student homework completion and whether students can get a certificate successfully [20]. Based on the MOOC data established by Peking University on Coursera, Zhang et al. for the first time classified and analyzed student behavior characteristics in Chinese MOOC and predicted learning results with three classification models: linear discriminant analysis (LDA), logical regression (LR), and linear nuclear support vector machine (l-SVM) [21]. Yu et al. identified seven cognitive participation models based on students' video clickstream logs. Also, they designed classification algorithms based on K-nearest neighbors (KNN), SVM, and artificial neural networks (ANN) to predict whether students can pass the course exam [22]. Mubarak et al. employed a dual analysis method combining visual and predictive analysis to visualize the data and model the behavior with an RNN-LSTM model, which improved the prediction accuracy of learning performance [23]. In addition, Wen et al. proposed a new simple feature matrix to maintain local correlation information with learning behavior and a novel Convolutional Neural Network (CNN) model to improve the completion rate of MOOC [24]. Hussain et al. analyzed data recorded in digital electronic Education and Design Suite (Contract) using five machine learning algorithms: artificial neural network (ANN), support vector machine (SVM), logical regression, naive Bayesian classifier, and decision tree [25]. ViSeq, an interactive visual analysis system available to visualize the learning order of different groups of learners proposed by Chen et al., helped users explore the learning sequence of MOOC from multiple granularity levels [26]. It includes four-link maps: projection views of

learner-identifying groups, pattern views showing the overall sequence patterns within the selected group, sequence views describing transformations between ongoing events, and individual views with an extended sequence chain. Cobo and Ruiz-Garcia presented the edX-LIS Learning Intervention System, which provides intervention strategies to improve learning performance, positively affecting learning motivation, persistence, and participation [27]. García-Molina et al. presented an automatic scoring algorithm based on the data dimension and correlation of learners' contributions [28]. It provides teachers with visual and digital representation for each learner, offering opinions for adjusting the expected behavior in the MOOC forum. In response to the rapid change in knowledge in the IT domain, Chen et al. proposed a framework for in-based expression through decision tree expression to evaluate the relationship between the curriculum and employment [29]. They designed the algorithm recommended by the course group based on the decision to provide the basis for adjusting the course configuration. The results indicate that the designed algorithms and recommended curriculum groups have a significant role in promoting student employment.

In essence, these studies are the integration of multiple classification problems. They analyzed the collected data (behavioral trajectory data) with different models or algorithm frameworks and finally offered predicting outcomes. Many studies have also proved the feasibility and effectiveness of relevant algorithms and models in result prediction, which provided a solid theoretical basis for the application and improvement of algorithms. This paper mainly collects students' learning behavior track data and divides their grades into excellent, qualified, and unqualified grades. We add feature selection based on GA and based classifier selection to the ECOC algorithm. We also optimize the ECOC coding matrix, feature subset, and choice of binary classifiers, taking accuracy as the evaluation index. By screening the learning behavior characteristics of raw data in the data preprocessing stage, the multiclassified accuracy can be improved. Next, we get the learning behavior features that influenced student performance to analyze through the fine-grained results of feature selection and correlation analysis. Finally, according to the character selection and correlation analysis of fine-grained results, the teaching methods can be improved and the teaching rhythm can be adjusted, which will be helpful in guiding the students' personalized learning and improve their learning performance.

3. Data Description

3.1. Course Settings. The datasets for this analysis were collected from a blended course of *Java Language Programming* in a college for four consecutive years. To make students understand the basic knowledge of Java and master the Java programming technology, this course integrates the characteristics of Java and introduces the programming foundation, object-oriented programming, GUI programming, and so forth; and it is scheduled for 68 class hours. Unlike traditional classes, it adopts the current popular online learning platforms such as rain classroom and

MOOC for auxiliary teaching. In addition to participating in conventional offline classes, students also need to use online platforms to preview, review, and complete tests or do some other online learning activities and take the final exam offline in the end.

3.2. Data Description. The course data are the learning track and data records collected from a blended course of *Java Language Programming* in a college for four consecutive years (the four different grades are represented by grade-1, grade-2, grade-3, and grade-4). The data source consists of four popular online education platforms: MOOC Platform, Rain Classroom, Programming Teaching Assistant (PTA), and Cnblogs.

Four grades (grade-1, grade-2, grade-3, and grade-4) have 94, 127, 130, and 122 people participating in the courses, respectively. Auxiliary teaching platforms for each grade are depicted in Figure 1.

3.2.1. Data of MOOC. The MOOC Platform data mainly includes the viewing of 102 teaching videos, the scores of 6 mutual-evaluation homework and 8 chapter tests, the cumulative number of discussions in the current semester, the learning times of each chapter, and the online test results as well as submission time. The data field is summarized in Table 1.

3.2.2. Data of Rain Classroom. Rain Classroom is an intelligent online teaching platform covering a variety of preclass-classroom-after-class teaching scenes. The collected data mainly include preclass preview situations and classroom performance. Tables 2 and 3, respectively, list the data composition and field description.

3.2.3. Data of Cnblogs. Cnblogs is a knowledge-sharing community for developers where students can consolidate knowledge and regularly make a summary by writing blogs. At the same time, teachers and teaching assistants can review students' blogs to understand their phased learning results and current situations. The main components of the scores include chapter knowledge summary, PTA exercise score situation, PTA exercise algorithm analysis, code reading, and learning feelings. Figure 2 portrays the composition of the Cnblogs' score.

3.2.4. Data Integration. In addition to the three categories of data sources, PTA online tests were added to integrating data. When processing the data, we need to align the data from different sources with the student ID as the standard and then combine the data to get four complete datasets. At the same time, the student's grades are divided into three grades, excellent, qualified, and unqualified, for the subsequent grades and grade prediction.

After integration, the pretreatment of the data (filtering, feature conversion) was conducted to obtain the final datasets. Table 4 lists the four datasets.

FIGURE 1: The source and composition of the data.

TABLE 1: The data of MOOC.

Feature types	Number	Data description	Examples
Teaching video viewing situation	102	The completion of watching each teaching video is presented in percentage	100%, 55%
Student information	7	School number, name, and other identifying information	
Discussion	4	The number of postdiscussions and reply discussions are presented by the number of replies	0, 5
Job scores	6	Job score, a total score of 5	0, 4.5, 5
Job completion status	6	Complete or not	Yes/no
Job mutual-evaluation status	6	Rate each other or not	Yes/no
Scores on tests	8	Online test scores, out of 100	0, 90.0, 100
Test completion time	8	Test completion time in YYYY-MM-DD, hh:mm:ss	2019-12-28, 00:24:09
Watching time duration	1	Student cumulative time for the video, in minutes	721 minutes
Accumulated learning times	1	Number of times students open a learning platform to learn	324
Progress of task point completion	4	Progress of video viewing and completion of the test, in scores	120/120, 1/8
Certificate issuance	1	The MOOC Platform can apply for certification after completing the specified indicators	Yes/no
The score of online platforms	2	Regular results and total results on the teaching platform, out of 100	90, 100
Five levels	1	Convert scores to grade system	A, B, C, D, E

TABLE 2: Data composition of the Rain Classroom.

Semester/grade	Preclass preview	Classroom performance
Fall semester 2017-2018 (grade-1)	25	9
Fall semester 2018-2019 (grade-2)	27	16
Fall semester 2019-2020 (grade-3)	26	15
Fall semester 2020-2021 (grade-4)	26	11

4. The ECOC Framework Based on GA and TBC

The multiple classification problem in the field of machine learning is often divided into multiple double classification problems. Finally, the multiple classifications are accomplished by integrating all the results. This paper focuses on ECOC classification-based algorithm by splitting multiple classification problems and combining multiple two classifiers.

4.1. Introduction to ECOC. ECOC algorithm, originated in the field of digital communication, is an ensemble learning method framework for multiclass classification. However, it is widely used in face recognition, traffic signal recognition, microarray data analysis, and other fields nowadays. In the multiclassification problem, by designing proper coding for each category, the ECOC ensemble system can achieve the error correction function by using the classifier to classify the samples into the correct category [30].

TABLE 3: Description of the rain classroom data.

Type	Field	Data description	Example
Essential information	Name	Student information	San Zhang
	Student ID	Student information	20XX21123XXX
	Total points	Total results in rain classroom (full marks are different)	100
	Sign-in times	Total number of classroom check-ins	11
	Access rate	Class check-in rate (percentage)	100%
Preview before class	Number of announcements read	Number of readings of published announcements	15
	Total number of pages viewed	Number of viewing pages of the pushed courseware	10
	Total viewing time length	Watching time of the pushed courseware XX h, XX m, XX s	00 h, 00 m, 00 s
	Finish time	Time to complete the courseware preview YYYY:MM:DD HH:MM:SS or completion status	Time/no preview/not completely finish
	Answer score	The score of the pushed courseware exercises	10
Classroom performance	Sign-in mode	The score of the pushed courseware exercises	Not signed in/scan the code to sign in
	Barrage releases	Number of barrage releases	0
	Accumulated score	Integration of classroom performance scores	10

FIGURE 2: Composition of the Cnblogs' score.

The ECOC algorithm mainly includes three basic steps [31]: encoding, training, and decoding. The process of constructing a coding matrix is called coding. The coding method is roughly divided into two categories: data-dependent and data-independent. The former refers to the model or distribution of the datasets during the coding process, typically including ECOC-ONE, DECOC (Discriminant ECOOC), and Data-Driven ECOC. The data-independent coding process only considers the number of categories. It does not analyze the model and distribution of the data, typically including OVO (One Versus One) [32], OVR (One Versus Rest), OVA (One Versus All), DR (Dense Random) [33], SR (Sparse Random) [34], and other coding methods [35]. The coding matrix consists of “+1,” “−1,” or “0.” Each coding matrix represents a category, and each column represents a binary classifier. Two classifiers train only the non-0 encoding corresponding category data and take “+1,” encoding corresponding category

training data as positive classes and “−1” corresponding category data as negative classes. Figure 3 portrays a classical one-to-many encoding (One vs All, OVA) for the encoding matrix. The ECOC algorithm is essentially an integration framework that improves model performance by integrating different classifiers and achieves classifier complementarity [36, 37].

The process opposite to coding is called decoding. The key to decoding is to determine the final category of the unknown sample based on the decision results of each base classifier. Decoding mainly consists of three types: decoding strategies based on output coding and target coding distance such as Hamming distance decoding, probability-based decoding such as Bayesian criterion, and mode space-based decoding strategy.

4.2. Feature Selection Method. Feature selection is a primary data processing method in machine learning [38, 39]. It improves the accuracy of prediction models and constructs

TABLE 4: Description of integration data summary.

Semester/grade	Data dimensions	Number of pieces of data
Fall semester 2017-2018 (grade 1)	119	82
Fall semester 2018-2019 (grade 2)	130	117
Fall semester 2019-2020 (grade 3)	130	115
Fall semester 2020-2021 (grade 4)	131	119

FIGURE 3: Schematic diagram of the OVA encoding matrix.

faster and less consuming prediction models and gives the model for better understanding and interpretation by selecting some features from all features for training models, enhancing the generalization ability of the model, highlighting the essence of problems, and simplifying the mathematical model.

The origins of GA date back to the early 1960s. GA searches for the optimal solution by simulating a natural evolutionary process. The problem-solving process is converted to the crossover and variation of chromosome genes in biological evolution through computer simulation operation. The algorithm has three basic operators: selection, crossover, and variation. Compared with traditional algorithms, GA is based on biological evolution with good convergence, less computational time, and high robustness in computational accuracy requirements. Meanwhile, the algorithm has scalability and is easy to combine with other algorithms [40].

4.3. The ECOC Framework Based on GA and TBC. The classical ECOC algorithm takes all the features as the input variables and uses a collection of base classifiers to train and predict the model. In this paper, based on ECOC and GA for performance grade prediction, we design a multi-classification algorithm framework. The algorithm performs

the optimization process to exchange their information by the vector operator computing code words between a pair of encoded columns; and it employs a collection of heterogeneous classifiers via GA for better solutions.

4.3.1. Gene. Gene is a feature in the solution of the problem. In this paper, a gene can guide encoding columns to computation to generate a new encoding column. The gene design mainly includes the index of the two selected matrix columns, the feature selection sequence, and two columns for calculation. Gene structure is illustrated in Figure 4.

4.3.2. Individual. An individual is consisted of a series of genes. It is a unit that needs to evolve or a solution to the original problem. In this algorithm, a new ECOC coding matrix can finally be generated through an individual by integrating new generated coding columns of genes; and the number of two classifiers determines the number of genes in the individual. The design of the individual is shown in Figure 5.

4.3.3. Operator. The algorithm has five types of operators. It is mainly used to operate the selected characteristic matrix column according to the chosen operator and generate a new characteristic matrix.

4.3.4. Legality Check. The ECOC encoding matrix has its unique legitimacy constraints. Illegal cases mainly include duplicate rows, reverse columns, and rows with only 0 symbols. For column levels, it means that the column contains only 0 or contains only +1 or -1 or that +1 or -1 is missing.

It is necessary to conduct legality testing before performing the evaluation operation, correct illegal operations, and discard some illegal columns.

4.3.5. Crossover. Crossover and mutation are the basic steps in genetic algorithms. The primary purpose of variation is to increase the diversity of the population, thus avoiding local optimality due to gene monopoly after superior individuals. The algorithm requires the exchange of genes or parts between individuals with a certain crossover probability. The process of gene crossing is shown in Figures 6 and 7: select parent individuals to cross first and then exchange the same type of bits in two selected gene individuals.

4.3.6. Mutation. Mutations are used to maintain and introduce diversity in genetic groups; and the mutation probability determines the number of genes in the

FIGURE 4: Gene structure.

FIGURE 5: The representation of individual.

FIGURE 6: Select parent individuals to cross.

FIGURE 7: Exchange the same type of bits in two selected gene individuals.

individuals selected for the mutation. Each gene has a low mutant mutation probability. For the chosen gene, each of them will change randomly with a mutation probability.

If the matrix index is selected to change, the original number will be changed within all the indexes of the encoding matrix pool. If the column index in the selection matrix is changed, do so within the number of columns. If the feature selection sequence is selected, some of the bits are selected in the digital sequence for mutation representing the feature selection index mutation of the classification model. If the operator is selected, change it in the five operator index ranges.

Figure 8 describes the example of mutation.

4.3.7. Elite Retention. The algorithm adopts an elite retention strategy to ensure that the classification effect rises during the iteration process and to avoid the optimal individual being destroyed by crossover and mutation. Suppose that the individuals with the worst adaptability in the new generation population are worse than the optimal individuals in the previous generation population. In that case, the worst individuals in the new generation will be replaced with the optimal individuals of the last generation to ensure the upward trend of evolution. In the evolution of genetic algorithms, the optimal individuals appearing to date are not lost or destroyed by the selection, crossover, and variation operations. Elite retention strategies play a significant role in improving the global convergence capabilities of standard genetic algorithms.

4.3.8. Optimization Method of the Binary Classifiers. The algorithm uses a hybrid classifier that places multiple classifiers into the chromosome and jointly iterates with the population to select the optimal base classifier corresponding to each column of each matrix. Each column is trained with the most optimum base classifier and combines the corresponding matrix column into a new matrix to obtain an optimal matrix.

4.3.9. Flowchart of the Proposed Algorithm

- (1) Relevant parameter setting: initialize all kinds of parameters, such as population iteration number,

crossover probability, mutation probability, the number of individuals in each generation of population, and the number of the initial coding matrices in the pool of coding matrix.

- (2) Population: A population is a collection of several individuals. It is the unit of iteration in the algorithm; and the evolution of the final result is conducted in population units.
- (3) Population initialization: The population contains several ECOC encoding matrix individuals. When generating a primary population, generate illegitimate ECOC coding matrices as primary individuals according to the constraint law of the ECOC matrix randomly.
- (4) Assess the new individual: The model takes accuracy as the index to calculate the classification effect of the model and performs the “survival of the fittest” operation.
- (5) Termination condition: Terminate the algorithm when the number of iterations reaches the preset value. The solution is now the optimal solution.

The algorithm is verified by the 5-fold cross and based on ECOC for model training and prediction. The average accuracy is used as the fitness function to measure the performance improvement of the multiclassification problem. In the iterative process of the algorithm, we select the more adaptable feature matrix and base classifier sequence for subsequent iterations. The final individual is the optimal individual. The algorithm’s flowchart is shown in Figure 9.

5. Experimental Results and Analysis

5.1. Experimental Settings. Parameter settings of ECOC algorithm are as follows: OVO, DECOC as encoding method, LR (logistic regression), SVM (support vector machine), Bayes as base classifier, and soft decoding as decoding method. The remaining parts adopt the default parameter settings.

Parameter settings of the GA are as follows: The number of individuals in the population is 80. The crossover probability is 0.8. The mutation probability is 0.1; and the maximum number of iterations is 50 generations. Parameter

FIGURE 8: Choose feature selection sequence to mutate.

FIGURE 9: Flowchart of the proposed TBCGA-ECOC algorithm.

settings for both Random Forest and XGBoost use the default parameters in the scikit-learn library.

All the experiments employ the 5-fold crossover validation method. The average accuracy value is used as the evaluation index of the classification performance and the fitness in the GA.

5.2. Experimental Results

5.2.1. Comparison with the Classical Ensemble Learning Algorithms. TBCGA-ECOC proposed in this paper is

compared with Random Forest and XGBoost, two classical ensemble learning algorithms in machine learning. As shown in Figure 10, the TBCGA-ECOC significantly improves the accuracy of predicting students' performance compared with the two classic algorithms. Therefore, we conclude that the TBCGA-ECOC offers better performance compared to the traditional ensemble learning algorithms.

5.2.2. Comparison with the ECOC Algorithms. On the basis of the ECOC algorithm, the TBCGA-ECOC algorithm utilizes the GA-based feature selection method and base

FIGURE 10: Comparison among TBCGA-ECOC and classic ensemble learning algorithms.

classifier selection to improve the prediction performance of the algorithm further. The comparative experimental results of the two classes of algorithms will be offered.

Table 5 reports the experimental results on different datasets when the OVO coding method is employed. As for grade-1 dataset: the accuracy of the TBCGA-ECOC algorithm increases about 3.53% and 3.58% over the results of the ECOC (LR) and ECOC (Bayes) algorithms, respectively; and it is pretty close to that obtained from the ECOC (SVM) algorithm. For the other three datasets, the TBCGA-ECOC algorithm results are better than those of ECOC (SVM), ECOC (LR), and ECOC (Bayes) algorithms. Among them, the algorithm results of the grade-2 dataset have increased by about 2.01%, 1.43%, and 7.11% based on the three algorithm results of ECOC (SVM), ECOC (LR), and ECOC (Bayes). The grade-3 dataset's accuracy is enhanced by 0.29%, 3.82%, and 13.15%. As for the grade-4 dataset, it also has an improvement of 1.42%, 2.53%, and 9.85%, respectively. The above results demonstrate that the proposed algorithm outperforms classical ECOC algorithms on all four datasets and shows significant improvement.

Similar results are also observed in experiments by using the DECOE encoding method. As shown in Table 6, the experimental results on different datasets are as follows.

Compared with the three algorithm results of ECOC (SVM), ECOC (LR), and ECOC (Bayes), the overall result of the TBCGA-ECOC algorithm has been improved. For example, the results of the grade-1 dataset are optimized by 1.97%, 4.42%, and 1.23%; and the grade-2 dataset's results are improved by about 3.17%, 3.17%, and 7.66%, respectively, based on other algorithms. For the two datasets, grade-3 and grade-4, the algorithm's results slightly rise on the basis of ECOC (SVM) and ECOC (LR) by 0.29% and 2.64% as well as 0.84% and 2.56%, respectively. Meanwhile, comparing the ECOC (Bayes) algorithm results, there is a significant enhancement, with an increase of 11.97% and 10.43%, respectively.

TABLE 5: Comparison of TBCGA-ECOC and ECOC (OVO)

	SVM	Logi	Bayes	TBCGA-ECOC
Grade 1	0.662 ± 0.330	0.622 ± 0.049	0.622 ± 0.053	0.657 ± 0.035
Grade 2	0.701 ± 0.022	0.707 ± 0.032	0.650 ± 0.071	0.721 ± 0.034
Grade 3	0.667 ± 0.022	0.631 ± 0.061	0.538 ± 0.032	0.667 ± 0.014
Grade 4	0.683 ± 0.032	0.672 ± 0.036	0.599 ± 0.033	0.697 ± 0.022

TABLE 6: Comparison of TBCGA-ECOC and ECOC (DECOE).

	SVM	Logi	Bayes	TBCGA-ECOC
Grade 1	0.638 ± 0.017	0.613 ± 0.031	0.645 ± 0.061	0.657 ± 0.035
Grade 2	0.690 ± 0.011	0.690 ± 0.028	0.645 ± 0.079	0.721 ± 0.034
Grade 3	0.667 ± 0.013	0.643 ± 0.034	0.550 ± 0.040	0.670 ± 0.014
Grade 4	0.689 ± 0.029	0.672 ± 0.050	0.593 ± 0.040	0.697 ± 0.022

Furthermore, it is observed that the accuracy of the algorithm on the grade-1 dataset is slightly low. The possible reasons are that the characteristic dimensions of the grade-1 dataset are fewer than those of the other three datasets, and the data have fewer dimensions. So the evolution is excessive under the same algorithm and parameters, and the overfitting phenomenon appears. Thus the accuracy is reduced to some extent. But, overall, TBCGA-ECOC improves to various degrees over ECOC accuracy on all four datasets.

These results conclude that the accuracy obtained from TBCGA-ECOC using different datasets or coding methods is higher than that from ECOC algorithms. Thus, the TBCGA algorithm based on GA feature selection and classifier traverse has excellent performance.

5.2.3. Comparison of Different Methods considering Prerequisite Courses. We tried to add the results of the prerequisite course to the dataset as a new feature; and, after data alignment, which can be achieved by eliminating the data missing features of prerequisite courses, we generated three new datasets with prerequisite courses and three without prerequisite courses corresponding to levels grade-2, grade-3, and grade-4.

To begin with, we utilize Random Forest and XGBoost, two traditional ensemble algorithms, to analyze and predict the six datasets with and without prerequisite courses. Figure 11 plots the prediction results.

There are six cases in total, and the prediction methods with prerequisite courses can obtain better results than those without prerequisite courses in 5 cases of 6 cases.

The two algorithms correspond to three grades, with six examples in total. In the five samples, the predicted results of the dataset with prerequisite courses were higher than or equal to those without prerequisite courses.

FIGURE 11: Feature contribution analysis on prerequisite courses for Random Forest and XGBoost.

FIGURE 12: Feature contribution analysis on prerequisite courses for the ECOC (OVO) method.

In addition, the classical ECOC algorithm and TBCGA-ECOC algorithm are tested in the two kinds of datasets, and the results are listed in Figures 12–14.

When adopting the ECOC algorithm with the OVO coding method, three classifiers (SVM, Logi, and Bayes) correspond to three different datasets, for 9 cases in total. The results of the dataset incorporating the characteristics of prerequisite courses win or flat in 8 samples, as shown in Figure 12. A similar phenomenon is found when using the ECOC algorithm with the DECOC encoding method, with 8 of the 9 samples winning, as shown in Figure 13. Furthermore, when using the TBCGA-ECOC algorithm, the

dataset with prerequisite courses can get better results in all the cases, shown in Figure 14.

In general, the algorithm that adds the prerequisite courses as a new feature cannot consistently achieve better results compared to the original algorithm. Still, the performance is usually better than the average performance of the original algorithm, and it has good robustness. Therefore, it can be seen that the characteristics of the prerequisite course are helpful to the prediction of grades and have a better classification effect. Moreover, as shown in Figure 15, the prediction accuracy of the TBCGA-ECOC algorithm with the features of prerequisite

FIGURE 13: Feature contribution analysis on prerequisite courses for the ECOC (DECOC) method.

FIGURE 14: Feature contribution analysis on prerequisite courses for the TBCGA-ECOC.

courses added is also superior to those of the other algorithms on the whole.

5.2.4. Time Complexity. The running times of TBCGA-ECOC for grade-1, grade-2, grade-3, and grade-4 levels are 1826 s, 2122 s, 2025 s, and 2778 s, respectively. Based on the population iteration with high algorithm complexity, the GA algorithm consumes much time, while Random Forest and XGBoost, the classical ensemble learning algorithm models, are relatively simple and can complete training within 5 s. Although ECOC employs

different base classifiers and different coding methods, its algorithm's complexity is also low.

The numbers of samples and features in the four datasets are different, so the computation times the algorithms need are different. The higher the feature dimension, the greater the solution space available for search. The grade-4 dataset with the highest dimension of features (131) and biggest number of samples (119) needs the longest running time, while the grade-1 dataset with the lowest dimension of features (110) and the smallest number of sample (82) needs the shortest running time. Finally, the grade-2 and grade-3 datasets need identical running times with similar

FIGURE 15: Comparison of TBCGA-ECOC and other methods with prerequisite courses feature.

dimensions and sample sizes. So the running time is less than that of the grade-4 dataset but more than that of the grade-1 dataset.

6. Conclusion

In this paper, we propose a novel ECOC multiclassification framework to predict students' grades. We try to add the prerequisite course as the new feature to improve the performance of the algorithm. The findings can be used to evaluate students' performance, providing valuable information for improving teaching in order to optimize individualized teaching, improve student performance, implement instructional interventions, and adjust the pace of teaching when necessary. The main conclusions are as follows:

- (1) Experimental results show that the proposed TBCGA-ECOC algorithm can obtain higher accuracy compared to several classical ECOC algorithms and the traditional ensemble learning algorithms, i.e., XGBoost and the Random Forest.
- (2) The prediction performance of various methods has been slightly or significantly improved by adding the grade of the prerequisites courses, i.e., *C language* and *Data Structure*, as the new features. On the contrary, the proposed method can provide valuable advice for curriculum and teaching planning. According to the predicted results, we can recommend suitable prerequisites to students.

In future work, we can follow the line of this work. As for the dataset, we can collect other multisource learning track data or add other prerequisite courses. As for the method, the ECOC algorithm can be improved, especially its adaptability with high-dimensional small sample data; or the

selection scheme of hybrid binary classifiers can be optimized so that the better binary classifiers can be selected to enhance accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Ling-qing Chen and Mei-ting Wu contributed equally to this work and should be regarded as joint first authors.

Acknowledgments

This work was supported in part by the Scientific Research Program of Fujian Bureau of Education, China (no. JAT200266), the Opening Fund of Digital Fujian Big Data Modeling and Intelligent Computing Institute, and the Scientific Research Foundation of Jimei University, China (no. F020803).

References

- [1] A. M. F. Yousef and T. Sumner, "Reflections on the last decade of MOOC research," *Computer Applications in Engineering Education*, vol. 29, no. 4, pp. 1–18, 2020.
- [2] K. L.-M. Ang, F. L. Ge, and K. P. Seng, "Big educational data & analytics: survey, architecture and challenges," *IEEE Access*, vol. 8, pp. 116392–116414, 2020.

- [3] Z. Xie, "Modelling the dropout patterns of MOOC learners," *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 313–324, 2020.
- [4] Z. Xu, H. Yuan, and Q. Liu, "Student performance prediction based on blended learning," *IEEE Transactions on Education*, vol. 64, no. 1, 2020.
- [5] K. F. Hew, X. Hu, C. Qiao, and Y. Tang, "What predicts student satisfaction with MOOCs: a gradient boosting trees supervised machine learning and sentiment analysis approach," *Computers & Education*, vol. 145, Article ID 103724, 2020.
- [6] H. Hu, G. Zhang, W. Gao, and M. Wang, "Big data analytics for MOOC video watching behavior based on spark," *Neural Computing & Applications*, vol. 32, no. 11, pp. 6481–6489, 2019.
- [7] C.-H. Yu, J. Wu, M.-C. Liu, and A.-C. Liu, "Adopting software product lines to implement an efficient learning analytics framework in MOOCs," *Journal of Information Science and Engineering*, vol. 37, no. 1, pp. 139–155, 2021.
- [8] L. Zhang, Y. Xuan, and H. Zhang, "Construction and application of SPOC-based flipped classroom teaching mode in installation engineering cost curriculum based on OBE concept," *Computer Applications in Engineering Education*, vol. 28, no. 6, pp. 1503–1519, 2020.
- [9] Y. Wang, C. Dong, and X. Zhang, "Improving MOOC learning performance in China: an analysis of factors from the TAM and TPB," *Computer Applications in Engineering Education*, vol. 28, no. 6, pp. 1421–1433, 2020.
- [10] J. Gardner and C. Brooks, "Student success prediction in MOOCs," *User Modeling and User-Adapted Interaction*, vol. 28, no. 2, pp. 127–203, 2018.
- [11] A. Onan and M. A. Toçoğlu, "Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts," *Computer Applications in Engineering Education*, vol. 29, no. 4, 2020.
- [12] M. A. Toçoğlu and A. Onan, "Sentiment analysis on students' evaluation of higher' educational institutions," *Advances in Intelligent Systems and Computing*, Springer, Germany, pp. 1693–1700, 2021.
- [13] O. Aytuğ, "Sentiment analysis on Twitter based on ensemble of psychological and linguistic feature sets," *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 69–77, 2018.
- [14] A. Onan, "Mining opinions from instructor evaluation reviews: a deep learning approach," *Computer Applications in Engineering Education*, vol. 28, no. 1, pp. 117–138, 2020.
- [15] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency and Computation: Practice and Experience*, Article ID e5909, 2020.
- [16] J. Xu, K. H. Moon, and M. Van Der Schaar, "A machine learning approach for tracking and predicting student performance in degree programs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 5, pp. 742–753, 2017.
- [17] R. L. Ulloa-Cazarez, C. López-Martín, A. Abran, and C. Yáñez-Márquez, "Prediction of online students performance by means of genetic programming," *Applied Artificial Intelligence*, vol. 32, no. 9–10, pp. 858–881, 2018.
- [18] B. Xu and D. Yang, "Motivation classification and grade prediction for MOOCs learners," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 2174613, 7 pages, 2016.
- [19] Y. Meier, J. Xu, O. Atan, and M. Van der Schaar, "Predicting grades," *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 959–972, 2015.
- [20] J. Qiu, J. Tang, T. X. Liu et al., "Modeling and predicting learning behavior in MOOCs," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 93–102, New York, NY, USA, February 2016.
- [21] M. Zhang, J. Zhu, Z. Wang, and Y. Chen, "Providing personalized learning guidance in MOOCs by multi-source data analysis," *World Wide Web*, vol. 22, no. 3, pp. 1189–1219, 2019.
- [22] C.-H. Yu, J. Wu, and A.-C. Liu, "Predicting learning outcomes with MOOC clickstreams," *Education Sciences*, vol. 9, no. 2, p. 104, 2019.
- [23] A. A. Mubarak, H. Cao, W. Zhang, and W. Zhang, "Visual analytics of video-clickstream data and prediction of learners' performance using deep learning models in MOOCs' courses," *Computer Applications in Engineering Education*, vol. 29, no. 4, pp. 710–732, 2020.
- [24] Y. Wen, Y. Tian, B. Wen, Q. Zhou, G. Cai, and S. Liu, "Consideration of the local correlation of learning behaviors to predict dropouts from MOOCs," *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 336–347, 2020.
- [25] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Using machine learning to predict student difficulties from learning session data," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 381–407, 2018.
- [26] Q. Chen, X. Yue, X. Plantaz et al., "ViSeq: visual analytics of learning sequence in massive open online courses," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 3, pp. 1622–1636, 2018.
- [27] R. Cobos and J. C. Ruiz-Garcia, "Improving learner engagement in MOOCs using a learning intervention system: a research study in engineering education," *Computer Applications in Engineering Education*, vol. 29, no. 4, 2020.
- [28] S. García-Molina, C. Alario-Hoyos, P. M. Moreno-Marcos, P. J. Muñoz-Merino, I. Estévez-Ayres, and C. Delgado Kloos, "An algorithm and a tool for the automatic grading of MOOC learners from their contributions in the discussion forum," *Applied Sciences*, vol. 11, no. 1, p. 95, 2021.
- [29] X. Chen, J. Zheng, Y. Du, and M. Tang, "Intelligent course plan recommendation for higher education: a framework of decision tree," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 7140797, 11 pages, 2020.
- [30] M. Sun, K. Liu, Q. Wu, Q. Hong, B. Wang, and H. Zhang, "A novel ECOC algorithm for multiclass microarray data classification based on data complexity analysis," *Pattern Recognition*, vol. 90, pp. 346–362, 2019.
- [31] T. G. Dietterich and G. Bakiri, "Solving multi-class learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1994.
- [32] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "NMC: nearest matrix classification - a new combination model for pruning one-vs-one ensembles by transforming the aggregation problem," *Information Fusion*, vol. 36, pp. 26–51, 2017.
- [33] K. Crammer and Y. Singer, "On the learnability and design of output codes for multi-class problems," *Machine Learning*, vol. 47, no. 2, pp. 201–233, 2002.
- [34] L. Zhou, Q. Wang, and H. Fujita, "One versus one multi-class classification fusion using optimizing decision directed acyclic graph for predicting listing status of companies," *Information Fusion*, vol. 36, pp. 80–89, 2017.
- [35] F. Masulli and G. Valentini, "Effectiveness of error correcting output coding methods in ensemble and monolithic learning machines," *Formal Pattern Analysis & Applications*, vol. 6, no. 4, pp. 285–300, 2004.

- [36] X.-N. Ye, K.-H. Liu, and S.-T. Liong, "A ternary bitwise calculator based genetic algorithm for improving error correcting output codes," *Information Sciences*, vol. 537, pp. 485–510, 2020.
- [37] Y.-P. Zhang, X.-N. Ye, K.-H. Liu, and J.-F. Yao, "A novel multi-objective genetic algorithm based error correcting output codes," *Swarm and Evolutionary Computation*, vol. 57, Article ID 100709, 2020.
- [38] M. A. Bagheri, Q. Gao, and S. Escalera, "A genetic-based subspace analysis method for improving error-correcting output coding," *Pattern Recognition*, vol. 46, no. 10, pp. 2830–2839, 2013.
- [39] L. I. Kuncheva and L. C. Jain, "Designing classifier fusion systems by genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 327–336, 2000.
- [40] D. Dutta, J. Sil, and P. Dutta, "Automatic clustering by multi-objective genetic algorithm with numeric and categorical features," *Expert Systems with Applications*, vol. 137, pp. 357–379, 2019.