Received November 21, 2020, accepted December 8, 2020, date of publication December 16, 2020,
date of current version December 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3045157

MOOC Dropout Prediction Using FWTS-CNN Model Based on Fused Feature Weighting and Time Series

YAFENG ZHENG[1], ZHANGHAO GAO[1], YIHANG WANG[2], AND QIAN FU[3]
[1]Computer and Information Engineering College, Henan University of Economics and Law, Zhengzhou 450000, China
[2]Artificial Intelligence and Computer Science College, Jiangnan University, Wuxi 214122, China
[3]School of Educational Technology, Beijing Normal University, Beijing 100875, China

Corresponding author: Qian Fu (fredqian@bnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61907011, and in part by the Key Technology Research and Development Program Foundation of Henan under Grant 202102310314.

ABSTRACT High dropout rates have been a major problem affecting the development of Massive Open Online Courses (MOOCs). Student dropout prediction can help teachers identify students who are tending to fail and provide extra help in a timely manner, helping to improve the effectiveness of online learning. In recent years, the use of convolutional neural networks for dropout prediction has yielded good results. However, traditional convolutional neural networks use automatic feature extraction, which does not consider the importance of the learner's behavior features and the effect of the time series of behavior on dropout, so it is difficult to guarantee the final prediction effect. To solve this problem, this article proposes a convolutional neural network model FWTS-CNN that integrates feature weighting and behavioral time series. It extracts continuous behavioral features from the learner's log of learning activities, filters key features and ranks them by importance based on the decision tree, then weights the continuous behavioral features based on importance, and finally builds a convolutional neural network model based on behavioral time series and weighted features. Experiments on the KDD Cup 2015 dataset show that the FWTS-CNN dropout prediction model has a high accuracy, which can reach more than 87%, an improvement of about 2% over using the CNN algorithm alone. The FWTS-CNN model integrates the effects of behavioral features and behavior time on dropout, effectively improving the accuracy of dropout prediction.

INDEX TERMS Dropout prediction, time series prediction, feature engineering, convolutional neural networks, accuracy.

I. INTRODUCTION

Massive Open Online Courses (MOOCs) are a new product of the Internet application innovation and open educational resources movement [1]. With the rise of online learning sites such as edX, Coursera and Udacity, MOOCs are gaining more and more attention worldwide [2]. Although the number of MOOC learners is expanding, low completion rates and high dropout rates are one of the most prominent issues in the sustainability of MOOCs [3], [4]. Numerous studies have reported MOOC course completion rates as low as 4-10% and dropout rates as high as 80-95%, which means that most students who participate in MOOCs abandon their studies before the end of the course [5].

The high dropout rate reflects the fact that most students who take MOOCs are not fully engaged in the online course and have difficulty learning from the online course. In addition, persistently high dropout rates can lead to a massive loss of online course participants, resulting in a high number of MOOC registrants but a low number of students actually completing the course. This phenomenon affects the quality of teaching and learning in online courses as well as the implementation of courses. Identifying students at high risk of dropping out early can help prevent dropout by providing them with access to additional course resources, peer support, and teacher guidance. Therefore, the extremely important research question in the field of online education is to obtain

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

the factors that cause students to drop out and find ways to reduce the dropout rate in MOOCs as a way to increase the final pass rate of online courses. This plays a crucial role in the development and construction of MOOC platforms [6].

On the MOOC platform, a large amount of learning behavior information is stored, such as student clickstream data, assignment submission, grade information, forum discussion participation and interaction information [7], [8]. Researchers usually analyze and extract the key factors that affect students' final learning effect from these data, in order to further use these data to predict dropout. Among them, clickstream data is the most widely used dropout prediction information due to the richness of the data [9]. Traditional clickstream-based dropout prediction is performed using two main types of data mining methods. One class views dropout prediction as a simple bicategorical problem, uses traditional machine learning algorithms for prediction, such as logistic regression [10], support vector machines [11], and decision trees [12]. Another class focuses on the time-series features of clickstream data, views the prediction as a typical time-series classification problem, which uses sequence mining models, such as the Hidden Markov method [13].

With the development of current deep learning algorithms, some deep learning algorithms are also used in dropout prediction, such as RNN [14], CNN [15], LSTM [16], etc., which greatly improves the prediction effect. In particular, CNN algorithm applications have become a current research hotspot and have been successful in many prediction areas such as image processing [17], natural language processing [18], and medical imaging [19]. The advantage of CNN algorithm is that it can map low-dimensional shallow features to high-dimensional deep features, and use the local correlation of the data to process tasks in various domains, which greatly improves the prediction effect. Because of its unique advantages, the CNN algorithm is also being used in the field of education.

However, there are some limitations in the application of CNN methods for education. First, CNN's automatic feature extraction method processes the data using local correlation. In image processing, neighboring pixels are likely to be the most correlated, but for numerical data, the neighboring data are not necessarily the most correlated. Second, for educational prediction applications, there is a strong correlation between students' final performance and their learning behavior features. Thus, the accuracy of feature recognition will greatly affect the effectiveness of the predictive model. However, the automatic feature extraction methods used by CNN cannot clearly distinguish the importance of different behavioral data. In addition, a large number of educational practice studies have shown that the sequence of behaviors has a great impact on students' learning performance and final grades [20]. However, current CNN studies rarely consider time as a key factor in model building.

In order to overcome the above shortcomings, this study obtains the importance ranking of various learner behavioral characteristics affecting dropout outcomes by constructing a decision tree model, then weights the features based on their importance, then constructs a two-dimensional time series matrix for the weighted features, and finally uses the CNN model to predict the MOOC dropout rate. Thus, a fusion feature-weighted and time-series convolutional neural network-based MOOC dropout prediction model (FWTS-CNN) is proposed to provide effective support for teacher intervention, educational decision, instructional design and platform support in large-scale online education. Further, the main contributions of this study can be summarized in the following three points:

1) We propose a feature-weighting method based on the importance of learner behavioral features on dropout outcomes, thereby highlighting the dominant role of key behavioral features in the MOOC dropout model.

2) We designed a time series matrix that combines learner behavior time information and learner behavioral feature data, which greatly improves the accuracy of the MOOC dropout prediction model.

3) We designed a comparative experiment to compare our FWTS-CNN model with related research models, which confirmed the effectiveness of the model in this article, and proved that it is superior to related research models under the same data set.

The rest of this article is organized as follows. Section 2 introduces related research work. Section 3 introduces the overall research framework and data set. Section 4 introduces the FWTS-CNN prediction model in detail. Section 5 introduces the results and discussion of the experiment. Finally, Section 6 summarizes the work of this article.

II. RELATED RESEARCH WORK

A. MACHINE LEARNING-BASED STUDY OF MOOC DROPOUT PREDICTION

Machine learning-based dropout prediction methods have been developed rapidly in recent years. By building models of interpretable algorithms, these methods focus on operationalizing prediction as a classification problem, using logistic regression, SVM, and decision trees to establish operational models, thereby improving the accuracy of prediction. Compared with the early classification prediction method, the prediction effect is significantly higher than the average level [21]. For example, Kloft *et al.* [22] proposed an SVM classification model for dropout prediction using clickstream data as input for the high dropout rate of MOOCs. Similarly, Lu *et al.* constructed an SVM model to predict whether MOOC learners would earn a certificate at the end of the semester. Liang used data from 39 courses collected on the XuetangX platform, mined user behavior log data, and built a gradient-enhanced decision tree model for dropout rate prediction with an accuracy of 89% [23]. Amnueyporsakul *et al.* used the attempted submission behavior and interaction features with course components, which mined from learner clickstream data, as effective predictors to construct a predictive model of learner dropout using SVM [24].

It is also evident from existing research that clickstream data is the most widely used data in MOOC dropouts [25]. However, it has also been found that the large amount of clickstream data, with numerous features and redundant information, not only increases the complexity of machine learning-based dropout prediction models, but also affects the accuracy of the models and the performance of the algorithms. For example, Chen *et al.* [26] proposed a decision-tree-based MOOC dropout prediction method, applying user clickstream data from MOOC platforms for dropout prediction. But due to the large number of features in the data and the complexity of the model, it is impossible to obtain better prediction results. Traditional machine learning is sensitive to feature engineering, and the feature quality of the input data directly affects the model predictions [27]. Thus, how to extract and identify an effective feature set through feature engineering methods becomes the key to research.

Different researchers have used different feature engineering methods to extract effective input feature sets depending on the specific research situation. For example, Qiu *et al.* [28] proposed a feature selection method named FSPred for clickstream log data, which scores the input features by ensemble feature selection method, and then uses forward feature extraction method in combination with the logistic regression classification model to find out the optimal feature subset, which effectively improves the classification model prediction performance and reduces the computational complexity. Gelman *et al.* [29] proposed a feature engineering approach based on non-negative matrix factorization (NMF) to analyze changes in the importance of learners' behavioral features in MOOCs over time, ultimately concluding that there are eight unique behavioral features that have a large impact on MOOC dropout outcomes on a sustained basis. Bote-Lorenzo and Gómez-Sánchez [30], in exploring the impact of learner behavioral features on dropout outcomes in each curriculum section, proposed a feature selection method based on feature correlation (CFS), which not only finds features with high predictive power but also identifies features with low redundancy.

B. DEEP LEARNING-BASED STUDY TO MOOC DROPOUT PREDICTION

In recent years, with the rapid development of MOOC platforms, online learning data has grown very rapidly, while data modeling issues such as high latitude, dynamics, and non-linear correlations have become increasingly prominent. Traditional machine learning algorithms face bottlenecks, and deep learning-based prediction methods have become a new hotspot for research. For example, Tang *et al.* [31] proposed a prediction model based on a recurrent neural network with short-term memory unit (RNN), which made the AUC value of the prediction result reach 88.1%. However, the RNN prediction model suffers from long-term dependency problems, making it difficult to conduct joint analysis of relevant data with a certain time span. Therefore, the use of RNN as a prediction model for MOOC dropout may not adequately capture the association between learning behaviors at different stages, resulting in limited prediction accuracy. To solve this problem, Wang and Wang [16] proposed an E-LSTM model that can process both event information and time information, this model separately weighted long-term interval data and short-term interval data, so that it can analyze the contextual information as well as combine the time information for dropout prediction. This model greatly improves the accuracy of MOOC dropout prediction and to some extent solves the problem of long-term dependence of learning behavior.

In addition to RNN and LSTM, CNN (Convolutional Neural Networks, CNN) algorithm is also a hot application in the field of deep learning algorithm in recent years. CNN is a multi-layer supervised learning neural network. The neurons in CNN only need to extract features based on correlation for local features, and then combine the output of all the neurons at higher levels to obtain global features, which avoids the complex feature extraction and data reconstruction process in traditional deep learning algorithms. CNN has achieved better prediction results in various fields such as image processing, natural language processing, and medical information. For example, in the field of image processing, CNN, by virtue of its unique fine-grained feature extraction method, has reached a level close to human eye recognition in the accuracy of image processing [17]. In the field of natural language processing, thanks to the structure in which text can be converted into a two-dimensional matrix through word vectors, CNN's efficient feature representation can be combined with natural language, thus demonstrating its powerful algorithmic performance in machine translation, information extraction, and question-and-answer systems. For example, He *et al.* [18] used the CNN algorithm to implement a high similarity text-matching model, which was used in the development of a question-and-answer system and greatly improved the usability of the question-and-answer system. As CNN algorithms continue to be affirmed in various prediction fields, some researchers also gradually starting to adopt CNN algorithms in exploring problems in the field of education. For example, Qiu *et al.* [15] proposed a convolutional neural network (CNN)-based prediction model that incorporates feature extraction and selection as well as classification of learners and uses 7 behavioral features from the XuetangX platform for prediction. The model prediction accuracy rate reached 86.75%.

C. TIME SERIES-BASED STUDY TO MOOC DROPOUT PREDICTION

Time-series data is a type of data that records changes in behavioral data over time. Time series based predictive models are widely used for modeling and forecasting in energy, finance and healthcare. For example, Chen *et al.* [32] proposed a deep learning time series prediction model based on LSTM, which overcomes the weak generalization ability and poor robustness of classification prediction models in the face of diverse data, and effectively improves the prediction performance of wind speed prediction models. Sidra et al [33] proposed a CNN deep learning regression model based on time series in predicting the future trend of stock prices. The model utilizes the time series data in weeks as model input data and achieves a correlation coefficient of 99% in predicting the stock price trend for the coming week. Further, the time series is combined with a variety of deep learning methods, enhancing the model's ability for potential feature extraction based on the time series. For example, Sahoo *et al.* [34] proposed a time series-based long and short-term memory recurrent neural network (LSTM-RNN), this model uses the combination of time series and LSTM-RNN to learn the data at a specific point in time, and the learned features depend not only on the current value of the observable object, but also on the past values, making it an effective solution to the problem of hydrological prediction during dry periods.

Student learning behavior data have typical time-series features. Numerous studies have shown that students' behavior at different times has a significant impact on their future academic performance. For example, Xing *et al.* [2] pointed out that time information is an important factor in analyzing the behavior of MOOC users. MOOC clickstream data are typically time-series data, and each clickstream operation of a student is accompanied by a timestamp. As a result, education prediction based on time series data has become a new research hotspot. By focusing on the time-series features of the data, some researchers propose to use nonlinear state-space models for MOOC dropout prediction. For example, Wang *et al.* A Nonlinear State Space Model for Identifying At-Risk Students in Open Online Courses applied the NSSM nonlinear state space model to predict student dropout rates by combining clickstream data from different weeks. Considering the good performance of deep learning, some researchers have used time series data as input data for deep learning to improve the accuracy of dropout prediction. For example, Wen *et al.* [36] designed a two-dimensional matrix based on time series as an input to the CNN model, combining time information with the learner's behavioral features to solve the problem of local correlation of behavioral features using time series. Qiu *et al.* [15] proposed an end-to-end dropout prediction model based on convolutional neural networks that integrates feature extraction and classification into a single framework. It transforms the raw timestamped data according to different time windows and automatically extracts features to achieve better dropout prediction. Overall, by using time-series data with time information in combination with classification prediction models, the problem of data sparsity can be addressed and the impact of the time interval between data on classification results can be fully considered. However, there are few current studies combining time series and deep learning methods for dropout prediction, and most of these studies adopt automatic feature extraction as feature engineering for dropout prediction models, ignoring the influence of the importance of different features on dropout prediction results.

Based on this, this study proposes a FWTS-CNN prediction model that integrates feature engineering and time series. We perform dropout prediction experiments by constructing a decision tree model for feature selection and weighting in feature engineering, using the time-series matrix with a weekly cycle as the input to the CNN model, and combining the key factor selection and weighting based on feature engineering with deep learning prediction to construct a deep neural network-based prediction model to strengthen the importance of key features and improve the prediction accuracy.

III. METHOD

A. PROBLEM DEFINITION

The dropout prediction problem in this study was defined as: tracking a learner's activity record over a five-week period, and the learner was considered to have dropped out of school if no learning activity occurred for 10 consecutive days. As a dichotomous prediction problem, it uses two different labels: dropout is labeled as 0 and non-dropout is labeled as 1.

B. DATASETS

The KDD CUP 2015 open dataset comes from "XuetangX", the largest MOOC platform in China, and is widely used in MOOC dropout prediction research [28]. In this article, the KDD CUP 2015 public dataset is used as the experimental dataset. The dataset records 120,542 activity logs of 79,186 students enrolled in 39 courses, each lasting five weeks, from 27 October 2013 to 1 August 2014. In this study, we randomly selected a total of 60,000 pieces of data from 13,081 learners from 7 categories of features generated during their course of study to construct a predictive model. In this dataset, the learner behavior information contains 7 learning behaviors of accessing objects, discussing, navigating courses, closing pages, trying to solve problems, watching videos, and browsing wikis, as shown in Table 1. In addition, a student may participate in multiple courses, and each student has a marker in each week's course indicating whether the student has dropped out or completed the course (0 for dropped courses).

TABLE 1. Types of events in the click stream.

Events	Description
problem	do homework
video	watch video
access	access to course objects other than videos and homeworks
wiki	view the course on Wikipedia
discussion	forum discussion
navigate	browse the rest of the course
page_close	close page

C. PROCESS OVERVIEW

The study proposes a method for dropout prediction using a FWTS-CNN model that incorporates feature engineering and time series. The overall process of the method consists of four main steps: data pre-processing, feature selection and weighting, model prediction, and indicator evaluation, as shown in detail in Figure 1. The study begins with data pre-processing of the KDD CUP 2015 public dataset. Then, in the feature selection and weighting stage, we perform decision tree analysis on the behavioral features of the pre-processed dataset to obtain the importance of each feature, weight the features according to their importance and generate the weighted time series matrix. In the model prediction stage, we use the CNN model with two-dimensional convolutional layers to extract the time series matrix of the first stage output, and then apply interconnected layer data functions to generate the final prediction results. This is a complete end-to-end system with all parameters co-trained by back propagation. The final performance was evaluated by Accuracy, Precision, Recall and F1-Score evaluation metrics.

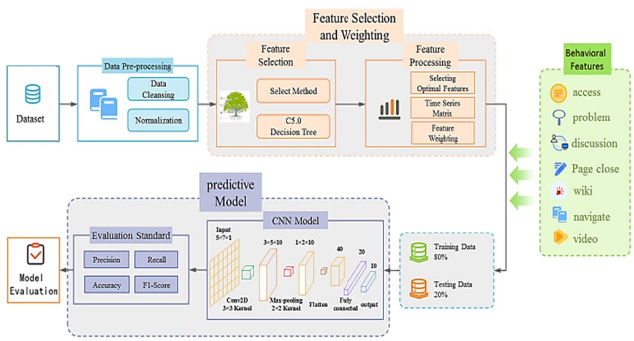

FIGURE 1. Process Implementation Diagram.

D. DATA PRE-PROCESSING

The study first performed feature frequency processing on the clickstream data in the KDD CUP 2015 raw dataset. Then, based on student id, course id, and event timestamp, we performed frequency counts on a weekly basis for accessing objects, discussing, navigating courses, closing pages, trying to solve problems, watching videos, and browsing wikis clickstream events in the Server and Browser of the raw dataset. The pre-processing procedure used is shown in Algorithm 1.

IV. FWTS-CNN MODEL

In this section, the study will focus on the core components of the FWTS-CNN model, from feature selection to feature weighting and finally to inputting the time series matrix into the CNN model. In this article, the specific steps in the implementation of the FWTS-CNN model are presented in two parts: feature selection and weighting and CNN model. The model framework is shown in Figure 2.

A. FEATURE ENGINEERING AND TIME SERIES MATRICES (FWTS)

The selection of learning behavioral features is the basis for accurate prediction of academic performance, thus the selection of key behavioral features that influence dropout plays

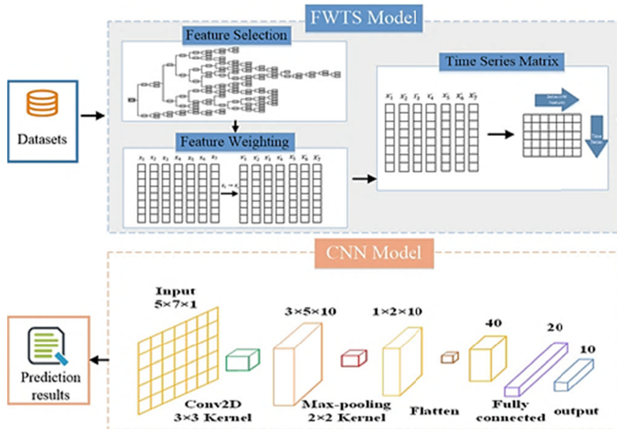

FIGURE 2. FWTS-CNN Model Framework Diagram.

Algorithm 1 Feature Pre-Processing

Input: courses set C, students set S, events set E, time period set T and number of events set N
Output: the dataset F with generation features

$F \leftarrow \emptyset$
Get time period set $T^i \in T$
Get number of events set $N^j \in N$
Get events set $E^k \in E$
for each course $c_n \in C$ do
 Get students set $S^m \in S$ about c_n
  for $m = 1$ to S do
   for $i = 1$ to T do
    for $j = 1$ to N do
     $E_{m,i,j} = E_j$
     $F_{m,i,j} = E_{m,i,j}$
    end
   end
  end
end

an important role in predicting MOOC dropout. In order to clarify the role of 7 learning behavior features in determining dropout prediction and to select the most important features, the study constructed a C5.0 decision tree model with these 7 behaviors and dropout labels for each student to reveal the extent to which learning behavioral features influence student dropout. The feature selection is shown in Algorithm 2.

In this study, 7 learning behavioral features (accessing objects, discussing, navigating courses, closing pages, trying to solve problems, watching videos, and browsing wikis) were introduced into the C5.0 decision tree model as independent variables and the dropout label as the dependent variable. The trained decision tree model is used to obtain the degree of influence of each variable on the dropout, so that we can select the optimal combination of behavioral features as the key feature behaviors for predicting MOOC dropout.

Based on the importance of the feature variables obtained from the C5.0 decision tree, the 7 features were ranked as

Algorithm 2 Feature Selection

Input: n instance $\{(x_i,y_i)\}$, $y_i \in \{0, 1\}$, feature ranking set TF, the number of points required for moving average α, the threshold value of difference β, the threshold value of times γ

Output: all feature subset T

$x^f \leftarrow$ generation features using Algorithm I
$T \leftarrow \emptyset$
Count $\leftarrow 0$
Position $\leftarrow 0$
for $i = 1$ to n do
  $T \leftarrow T \cup \{TF_i\}$
  $x'' \leftarrow$ extract all feature of T and the corresponding values in x^f
  Training the selected classifier with the decision tree to obtain the evaluation score S_i corresponding to the current feature subset
  if $i >= \alpha$ then
    $p_{Ti} \leftarrow \frac{1}{\alpha}\sum_{j=0}^{\alpha-1} S_{i\_j}$
  end if
  if $i > \alpha$ then
    $d_{T_i} \leftarrow p_{T_i} \leftarrow p_{T_{i-1}}$
    if $d_{T_i} < \beta$ then
      Count $\leftarrow$ Count $+ 1$
      if Count $= \gamma$ then
        Position $\leftarrow$ the position corresponding to the maximum value of the evaluation scores from $i - \gamma - \alpha$ to $i - \gamma$
        break
      end if
    end if
    if $d_{T_i} > \beta$ then
      Count $\leftarrow 0$
    end if
  end if
end for
for $i = 1$ to Position do
  $T \leftarrow T \cup \{TF_i\}$
end for

accessing objects, trying to solve problems, discussing, closing pages, browsing wikis, navigating courses, and watching videos, with corresponding feature weights of 0.54, 0.29, 0.1, 0.05, 0.02, 0, and 0. Using the weighting of each behavioral feature, the study highlights the dominant role of key features in the prediction model, enabling the model to more accurately predict MOOC dropout. Therefore, each behavioral feature is weighted separately on the basis of the raw data. The weighting method is as follows:

$$X_i' = X_i \cdot (1 + Q) \quad i \leq 7 \tag{1}$$

where X_i is the original value of the behavioral feature, X_i' is the weighted value of the behavioral feature, and Q is the weight. Each learning behavioral feature is assigned a corresponding weight to obtain the weighted feature. The weighted learning behavioral features are used as raw feature data for the FWTS-CNN model.

By analyzing the features of the learner's learning behavior in the raw data, this article proposes the design of a two-dimensional time series matrix. This study performs time series matrix construction on the weighted dataset. In the KDD dataset, each course was taught over a period of 5 weeks and the learners had 7 learning behavior features. Therefore, this study decided to take 5 weeks as a time period in terms of the time dimension, and evenly divided the time period into five-time dimensions by week; in the behavioral features dimension, we divided 7 dimensions and counted the frequency of the 7 learning behavioral features weighted to construct the final 5×7 time series matrix. Each matrix records a learner's weighted frequency of 7 learning behavioral features for each week of a course. For each learner, we extracted i frequencies of learning behavioral features over t weeks, expressed as a vector $x'_{t,i}t \leq 5, i \leq 7$. The resulting time series matrix is shown in Figure 3. In the next section of the FWTS-CNN predictive model, the study uses the time series matrix as a data input for the next step of the study.

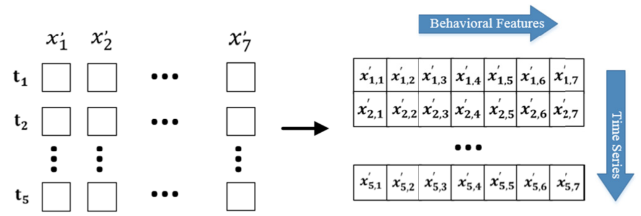

FIGURE 3. Time series matrix.

B. CNN MODEL

After processing through feature engineering and time series matrix stages. The output dataset is a five-week feature matrix of the student's time series behavior, which is entered as input to the CNN model in the convolution phase.

As shown in Figure 4, our attempt to utilize as few hidden layers as possible while achieving high classification performance. The model starts with an input layer, connected to a convolutional layer, a pooling layer, and two fully connected layers. The logistic classifier is finally used to output the classification results to which the student behavioral characteristics belong.

The data of the CNN model in this article is in the form of matrices as input, therefore, the different features of the input data are extracted from the convolution of 2D convolutional kernels in the convolutional layer to generate feature maps.

$$C_i^l = conv2\left(D, K_i^l\right) + b_i^l \tag{2}$$

In equation(2), C_i^l denotes the ith feature extracted from the input data by the lth layer convolution, $conv2()$ denotes a convolutional operation, D is the input matrix, K_i^l denotes the convolutional kernel weight of the ith feature of the lth layer

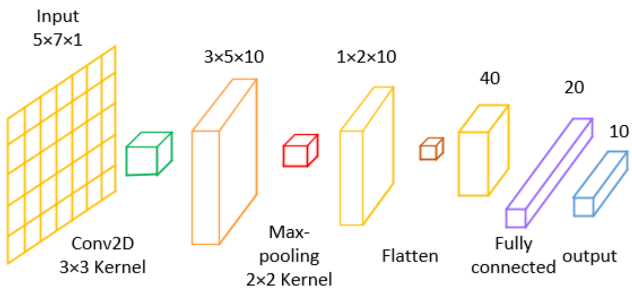

FIGURE 4. CNN Framework Diagram.

convolution. b_i^l denotes the deviation of the ith feature of the lth layer convolution.

$$a_i^l = f\left(C_i^l\right) = f\left(\sum\nolimits_{j=1}^{N_{l-1}} conv2\left(a_j^{l-1}, K_{ij}^l\right) + b_{ij}^l\right) \quad (3)$$

In equation(3), a_i^l denotes the ith feature map extracted from the input data by the lth layer convolution, K_{ij}^l denotes the kernel weight of the convolution of the ith feature map connected to the jth feature map of the lth layer convolution, $f()$ denotes the activation function of the convolutional layer. In this article, a nonlinear activation function, ReLU, is used to effectively prevent gradient disappearance and speed up the training of the model. The formula is shown below.

$$f(\alpha) = \max(0,\alpha) \quad (4)$$

In order to reduce the number of parameters in the model and prevent overfitting, in this article, the max-pooling layer is connected after the convolutional layer, and the output of the convolutional layer is processed to generate pool maps. The formula for the max-pooling layer is as follows.

$$S_i^l = \beta_i^l down\left(a_i^l\right) + b_i^l \quad (5)$$

$$a_i^l = f\left(S_i^l\right) = f\left(\beta_i^l down\left(a_i^{l-1}\right) + b_i^l\right) \quad (6)$$

In equations (5) and (6), S_i^l denotes the ith feature maps extracted from the convolutional layer by the lth layer pooling window. β_i^l denotes the scalar parameter of the ith feature map of the lth layer pooling window, $down()$ denotes downsampled operation.

After the pooling layer, the first fully connected layer accepts the output from the pooling layer and generates the input for the second fully connected layer. The second fully connected layer generates inputs for the logistic classifier. The formula for the fully connected layer is as follows.

$$a_i^l = f\left(\sum\nolimits_{i=1}^{N_{l-1}} \beta_i^l down\left(a_i^{l-1}\right) + b_i^l\right) \quad (7)$$

In the above formula, a_i^l is the output of the first fully connected layer and the input of the second fully connected layer, $f()$ denotes the activation function of the convolutional layer. In the first fully connected layer, he activation functions for both the same convolutional and pooling layers are ReLU functions. However, in the second fully connected layer, we use the logistic function as the activation function of this layer to output the classification results to which the student's behavioral characteristics belong. The formula for the logistic function is as follows.

$$logistic(\theta) = \frac{1}{1+e^{-\theta}} \quad (8)$$

In calculating the gradients of the model parameters, considering that the input data in this article is matrix form. To ensure the stability of matrix and vector calculations, reduce the fluctuation factor of parameter updates. In this article, the backpropagation algorithm is used to calculate the gradient of the model parameters and a small batch of randomly and uniformly selected training samples is used to calculate the gradient to ensure a more stable convergence of the model parameters.

In the fully connected layer, the backpropagation computation is performed in the same way as for ordinary neural networks. The formula is as follows.

$$\delta^l\left(a^{l-1}\right)^T = \frac{\partial C}{\partial K^l} \quad (9)$$

$$\delta^l = \frac{\partial C}{\partial \gamma^l} \quad (10)$$

In equations (9) and (10), δ^l is the residual value, a^{l-1} is the output of the l-1th layer convolutional neural network, K^l is denoted as the weight of layer l, γ^l is denoted as $K^l a^{l-1} + b^l$, C denotes the cost function of the model. In this model, we use cross-entropy as a cost function. The formula is as follows.

$$C = -\frac{1}{n}\sum\nolimits_{j=1}^{n}\left(y_j \ln a_j^l + (1-y_j)\ln(1-a_j^l)\right) \quad (11)$$

In the convolutional layer, the δ of each neuron in layer l is related only to the relevant neuron connected to the pooled layer $l+1$. The layer l convolutional layer to the layer l+1 pooling layer does a downsampled operation that reduces the matrix dimension. δ_i^{l+1} uses the upsampling operation to restore the matrix dimension of the convolutional layer. The formula is as follows.

$$\delta_i^l = \beta_i^{l+1}\left(a\left(C_i^l\right)\circ up\left(\delta_i^{l+1}\right)\right) \quad (12)$$

$$\frac{\partial C}{\partial b_i^l} = \sum\nolimits_{s,t}(\delta_i)_{st} \quad (13)$$

$$\frac{\partial C}{\partial K_{ij}^l} = \sum\nolimits_{s,t}\left(\delta_i^l\right)_{st}\left(P_j^{l-1}\right)_{st} \quad (14)$$

In the above formulas, $\circ$ denotes element-wise multiplication, $()_{st}$ denotes iterate through all elements, $\left(P_j^{l-1}\right)_{st}$ is a matrix of related elements of a_i^{l-1} in layer $l-1$ that δ_i^l connects to.

In the pooling layer, the lth pooling layer to the l+1th convolution layer does the operation to make the convolution narrower so that the matrix dimension is reduced. δ_i^{l+1} needs to extend the dimension of the matrix with the corresponding convolution kernel for the operation that widens the convolution. The formula is as follows.

$$\delta_i^l = \sum\nolimits_{j=1}^{N_l} a_i^l \circ conv2\left(a_j^{l+1}, K_{ji}^l\right) \tag{15}$$

$$\frac{\partial C}{\partial b_i^l} = \sum\nolimits_{s,t} \left(\delta_i^l\right)_{st} \tag{16}$$

$$\frac{\partial C}{\partial \beta_{ij}^l} = \sum\nolimits_{s,t} \left(\delta_i^l \circ d_i^{l-1}\right)_{st} \tag{17}$$

In the above formulas, $d_i^{l-1} = down(a_i^{l-1})$.

V. RESULTS AND EVALUATION

A. EVALUATION STANDARD

In order to assess the generalization performance of the FWTS-CNN prediction model developed in this article, it is necessary to rely on evaluation metrics that measure the generalization ability of the model. This study uses Accuracy, Precision, Recall, and F1-Score as evaluation metrics, which are commonly used to assess the performance of binary classification [37]. Precision indicates the accuracy of samples predicted to be positive, i.e., the proportion of samples predicted to be positive that are actually positive; Recall indicates the proportion of positive samples that are correctly predicted, i.e., the proportion of all positive samples that are predicted to be positive; And F1 indicates that the classifier performance can be evaluated comprehensively after reconciling the correct rates and recall rates.

For a binary classification problem, classifying an instance as either a positive or a negative will have four scenarios, where TP (True Positive) denotes the number of correctly classified positive classes; TN (True Negative) denotes the number of correctly classified negative classes; FP (False Positive) denotes the number of misclassified positive classes; and FN (False Negative) denotes the number of misclassified negative classes.

From this, we can calculate the various evaluation indicators of the model, and the specific formulae for each evaluation indicator are shown below.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{19}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{20}$$

$$\text{F1} = \frac{2 \cdot \text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{21}$$

All 3 models use the same training set of data to train the model and use the same test set to evaluate the model.

B. BASELINE MODEL

In order to provide a reference point for the results of the FWTS-CNN model, we used 5 traditional machine learning models that are more commonly used in existing research, namely Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT), as the baseline model. Considering that the above 5 traditional machine learning models are commonly used in different machine learning applications, we only briefly introduce them here.

1) Logistic regression (LR) is a supervised learning classification method based on generalized linear regression analysis. It estimates the probability of an event occurring based on the regression coefficients of one or more characteristic variable [38].

2) Naive Bayes (NB) is a quasiclinear classification algorithm in the supervised learning method. The naive Bayes model calculates the frequency of each category in the training set data and the conditional probability of each category divided according to each feature value, simplifies the modeling of a joint probability distribution to model each eigenvalue. This completes the construction of the naive Bayes model [39].

3) Random Forest (RF) is a machine learning method that builds a bag ensemble based on decision trees, which introduces random attribute selection in the decision tree training process to improve the generalization performance of the integrated classifier [40].

4) Support Vector Machine (SVM) is a supervised machine learning method commonly used for binary classification applications. The support vector machine approach is based on the VC dimension of statistical learning theory and the principle of minimal st structure risk. The support vector machine model can solve the small-sample, nonlinear, high-dimensional pattern recognition problem. It mimics the nonlinearity of the input space to the high-dimensional feature space, which is different from the dimensionality reduction method commonly used in traditional pattern recognition, allowing the linear unclassifiable in the low-dimensional space to be linearly separated in the high-dimensional space [41].

5) Decision tree (DT) is based on tree structure for decision making. Each non-leaf node in the tree represents a test of a feature attribute, each branch represents the output of that feature attribute within a certain range, and each leaf node stores a category. The decision process starts from the root node, tests the corresponding feature attribute of the item to be classified, selects the output branch based on its value until it reaches the leaf node, and the category stored by the leaf node is the result of the decision. Information gain, gain rate and Gini index can be used as a basis for optimal attribute classification [42].

To facilitate comparison and analysis, the study first checked the completeness of the data and treated missing data to eliminate the effect of incomplete data on the results; we then normalize the data so that the variables in each dimension fall into the interval [0, 1]; we then use the proposed feature generation method to generate features from the raw data; finally, we use the generated features for learning and prediction of the baseline model, using 80% of the data as a training set and 20% as a test set. The experiments were all iterated 10 times. The purpose of normalization is to make the features numerically comparable between different dimensions and to speed up the descent of the gradient to the optimal solution.

C. HYPERPARAMETER SETTING

The model used the behavioral features of each student every 5 weeks as input matrices, each with a size of 5 × 7. The specific parameters are shown in Table 2. The first hidden layer is the CNN layer. In this article, 128 filters are used to generate 128 feature mappings, all of which are 5 × 7 in size. In the 2nd hidden maximum pooling layer, the pooling size is set to 2×2. The 3rd hidden layer is a fully connected layer that generates 7 vectors for every 5 weeks and outputs 7 vectors of size 128 dimensions. To prevent overfitting, the Dropout in the model takes a value of 0.2. The final output of the model is a 0-1 number that indicates the probability of the student dropping out of the corresponding course, with "0" indicating "no dropout" and "1" indicating "dropout".

TABLE 2. Model hyperparameters.

Parameters	Values
kernel size	20
pooling size	2×2
filters	40
dropout	0.2
epochs	20

This article implements the FWTS-CNN model using Keras, a Python library for implementing deep learning methods. In the experiments, we use a log loss function to train the parameters. The optimization function takes an adaptive learning rate approach and runs the model for 20 cycles.

D. EXPERIMENTAL RESULTS

The KDD CUP 2015 dataset is available as a public dataset and has been used by a number of researchers to conduct dropout studies of MOOCs using deep learning models. Therefore, this study first used the FWTS-CNN model to compare the modeling with the baseline model designed in the study. The test results of the different algorithms in the experiment are shown in Table 3. The best values of the experimental results are shown in bold. Here, the input data for the baseline model were not processed using feature engineering and 2D matrix input.

TABLE 3. Comparison of results with the baseline model.

Model	Accuracy	Precision	Recall	F1-Score
LR	0.671	0.665	0.671	0.643
NB	0.673	0.669	0.673	0.670
RF	0.658	0.650	0.658	0.651
DT	0.622	0.624	0.622	0.623
SVM	0.681	0.674	0.681	0.675
CNN	0.853	0.843	0.847	0.845
FWTS-CNN	**0.871**	**0.863**	**0.865**	**0.864**

As shown in Table 3, the overall performance of the FWTS-CNN model is higher than the other models. The FWTS-CNN model is higher than the other seven models in terms of the four assessment values. The assessment metrics of each model are shown in Figure 5. Compared with the LR, NB, RF, DT, and SVM models, the overall performance of the FWTS-CNN model improves by about 20%, indicating that the FWTS-CNN model is substantially better than traditional machine learning models in classifying the MOOCs dropout problem. Compared with the CNN model, the Accuracy, Precision, Recall, and F1-Score of the FWTS-CNN model are respectively higher by 1.8%, 2%, 1.8%, and 1.9%. The results show that the FWTS-CNN model, after utilizing the feature engineering and 2D matrix input processing designed in this study, has a significant improvement in the classification performance of MOOCs dropout problem. The above results suggest that the FWTS-CNN model can better predict the dropout problem in MOOCs.

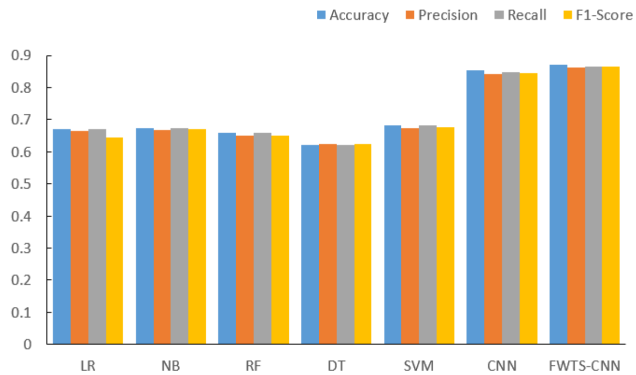

FIGURE 5. Indicator Diagram for Model Assessment.

Table 4 shows the effect of size of time series matrix. The FT-CNN model has the best performance when using the 5 × 7 time series matrix. Looking at Figure 6, it can be seen that the FT-CNN model has the worst classification performance when using the 1 × 7 time series matrix as the input data, with an accuracy of only 65.4%. The reason for this result is that during the first week of the course, the students had fewer course tasks making the behavioral characteristics produced by the students relatively little. As the course continued, the number of characteristic behaviors produced by the students increased. When utilizing the 2 × 7 time series matrix as input data, the performance of the model improved by approximately 12% compared to the 1 × 7 time series matrix. By looking at the results using the 4 × 7 time series matrix and the 5 × 7 time series matrix as input data, respectively, it can be seen that the difference between the two experiments is only about 1.7%. This phenomenon is due to the fact that the course duration is only five weeks, and the number of behavioral characteristics of some students drops significantly after the course, leading to partial data missing in the fifth week, which has a certain impact on the performance of the model.

From the recently published literature, this article selected five representative MOOC dropout prediction models for comparison tests with the FWTS-CNN model as follows: the

TABLE 4. Effect of size of time series matrix.

Framework	Accuracy	Precision	Recall	F1-Score
FWTS-CNN with 1×7 matrix	0.654	0.652	0.648	0.643
FWTS-CNN with 2×7 matrix	0.776	0.726	0.722	0.724
FWTS-CNN with 3×7 matrix	0.817	0.745	0.750	0.748
FWTS-CNN with 4×7 matrix	0.854	0.841	0.849	0.845
FWTS-CNN with 5×7 matrix	**0.871**	**0.863**	**0.865**	**0.864**

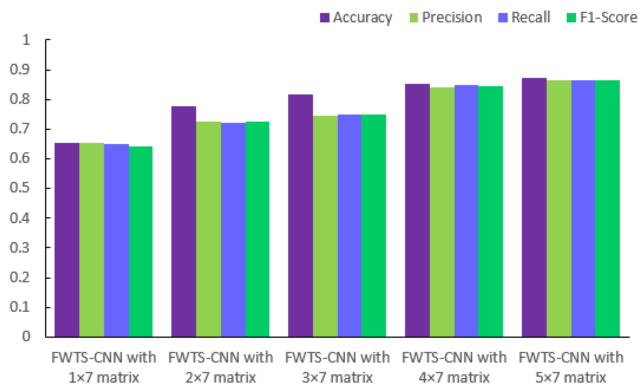

FIGURE 6. Assessment of different time series matrices.

logistic regression-based machine learning MOOC dropout prediction model proposed by Qiu *et al.* [28], the support vector machine based machine learning MOOC dropout prediction model proposed by Kloft *et al.* [22], the LSTM neural network MOOC dropout prediction model proposed by Wang *et al.* [43], the CNN combined with LSTM MOOC dropout prediction model proposed by Sun *et al.* [44], and the time series-based CNN MOOC dropout prediction model proposed by Qiu *et al.* [15]. The KDD CUP 2015 dataset was used as the original dataset for all models in the experiment. The final evaluation results of the models are shown in Table 5.

TABLE 5. Comparison of results with other models.

Model	ACC	Precision	Recall	F1-Score
LR[28]	-	0.857	0.863	0.847
SVM[22]	-	0.801	0.793	0.784
LSTM[43]	0.805	0.801	0.792	0.784
CNN-LSTM[44]	0.848	-	-	0.853
DP-CNN[15]	-	0.842	0.849	0.837
FWTS-CNN	0.871	0.863	0.865	0.864

As can be seen in Table 5, the model proposed in this study has some performance advantages when compared with deep learning-based dropout models of MOOCs designed by other researchers. The results illustrate that among the many preprocessing methods and deep learning models, the feature engineering and two-dimensional matrix input methods proposed in this study are more suitable for predicting the dropout problem in MOOCs.

The results show that:

1) The FWTS-CNN model proposed in this article can effectively solve the dropout prediction problem in the large-data MOOC, and achieves better results than the baseline method.

2) In comparing the MOOC dropout prediction models of other researchers, the feature engineering and two-dimensional matrix input methods proposed in this article can better help the CNN classification model for dropout prediction.

3) Since the FWTS-CNN model can achieve an accuracy of 87.1%, a model trained using clickstream data from the KDD CUP 2015 dataset can be used to predict dropout rates for new courses.

VI. CONCLUSION

In this study, we propose a dropout prediction model (FWTS-CNN) based on a two-dimensional convolutional neural network to address the problem of dropout prediction in MOOCs. This model can directly process the clickstream data, filter the behavioral features of the data to obtain the most important features, and then weight the original behavioral feature data using the behavioral feature weights. After that, it uses the completed weighted behavior feature data to construct a time series matrix as the input of the CNN dropout prediction model to predict the dropout of students in the MOOC. This study compares the FWTS-CNN model to the baseline model using a dataset of 39 courses collected by XuetangX. The experimental results show that with sufficient data, the FWTS-CNN prediction model achieves better results than the correlation baseline model by feature weighting supplemented with a two-dimensional time series matrix as input, which not only improves the prediction performance but also reduces the computational complexity. Therefore, based on our findings, the model proposed in this article can help instructors to further understand students' learning status based on their recent behavioral characteristics in large-scale online open courses. Teachers can analyze student status to enhance monitoring and personalized tutoring for students with potential dropout risk, reduce dropout rates, and improve the quality of online courses. However, there are some shortcomings in the study. Some of the students in the dataset used in this study do not have a learning period of five weeks, resulting in insufficient valid data that can be put into the experiment.

In future work, we will investigate how other factors from multiple sources can be combined to enhance the dropout prediction of MOOCs, as well as collect more data on the learning behavior features of MOOCs learners to refine the experimental dataset.

REFERENCES

[1] Tang, Jeff K T, H. Xie, and T. L. Wong, "A big data framework for early identification of Dropout students in MOOC," in *Proc. ICTE*, Hong Kong, Jul. 2015, pp. 127–132.

[2] W. Xing, X. Chen, J. Stein, and M. Marcinkowski, "Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization," *Comput. Hum. Behav.*, vol. 58, pp. 119–129, May 2016.
[3] K. Jordan, "Initial trends in enrolment and completion of massive open online courses," *Int. Rev. Res. Open Distrib. Learn.*, vol. 15, no. 1, pp. 133–160, Jan. 2014.
[4] I. Nawrot and A. Doucet, "Building engagement for MOOC students: Introducing support for time management on online learning platforms," in *Proc. 23rd Int. Conf. World Wide Web - WWW Companion*, 2014, pp. 1077–1082.
[5] P. Parslow, "Dropout: MOOC participants' perspective," in *Proc. EMOOCs*, Lausanne, Switzerland, Feb. 2014, pp. 95–100.
[6] S. Li and L. Qian, "Analysis and thinking of MOOC's high Dropout rate," *China Educ. Technol. Equipment.*, vol. 22, no. 17, pp. 132–133, 2015.
[7] D. Yang, M. Wen, A. Kumar, E. P. Xing, and C. P. Rose, "Towards an integration of text and graph clustering methods as a lens for studying social interaction in MOOCs," *Int. Rev. Res. Open Distrib. Learn.*, vol. 15, no. 5, pp. 214–234, Oct. 2014.
[8] S. Halawa, "Dropout prediction in MOOCs using learner activity features," *eLearn. Papers.*, vol. 37, no. 1, pp. 1–10, Mar. 2014.
[9] J.-L. Hung, M. C. Wang, S. Wang, M. Abdelrasoul, Y. Li, and W. He, "Identifying at-risk students for early interventions—A time-series clustering approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 45–55, Jan. 2017.
[10] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg, "Your click decides your fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions," *ACM Trans. Comput-Hum Int.*, vol. 14, no. 16, pp. 3–14, Oct. 2014.
[11] T. A. Cardona and E. A. Cudney, "Your click decides your fate: Inferring information processing and attrition Behavior from MOOC video clickstream interactions," *Procedia Manuf.*, vol. 39, pp. 1827–1833, Aug. 2019.
[12] K. P. Shaleena and S. Paul, "Data mining techniques for predicting student performance," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, Mar. 2015, pp. 1–3.
[13] G. Wang, Y. Tang, J. Li, and X. Hu, "Modeling student learning Behaviors in ALEKS: A two-layer hidden Markov modeling approach," in *Proc. 19th AIED*, London, U.K., Jun. 2018, pp. 27–30.
[14] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 256–263.
[15] L. Qiu, Y. Liu, Q. Hu, and Y. Liu, "Student dropout prediction in massive open online courses by convolutional neural networks," *Soft Comput.*, vol. 23, no. 20, pp. 10287–10301, Oct. 2018.
[16] L. Wang and H. Wang, "Learning Behavior analysis and dropout rate prediction based on MOOCs data," in *Proc. 10th Int. Conf. Inf. Technol. Med. Edu. (ITME)*, Aug. 2019, pp. 419–423.
[17] M. J. Nordin, O. W. Xin, and N. Aziz, "Food image recognition for price calculation using convolutional neural network," in *Proc. 3rd Int. Conf. Digit. Signal Process. ICDSP*, 2019, pp. 80–85.
[18] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1576–1586.
[19] A. Prasoon, K. Petersen, C. Igel, F. Lauze, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," *Med. Image Comput. Comput.-Assisted Intervent.*, vol. 16, no. 2, pp. 246–253, Sep. 2013.
[20] Q. Chen, X. Yue, X. Plantaz, Y. Chen, C. Shi, T.-C. Pong, and H. Qu, "ViSeq: Visual analytics of learning sequence in massive open online courses," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 3, pp. 1622–1636, Mar. 2020.
[21] M. Sharkey and R. Sanders, "A process for predicting MOOC attrition," in *Proc. EMNLP Workshop Anal. Large Scale Social Interact. MOOCs*, Oct. 2014, pp. 50–54.
[22] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, "Predicting MOOC dropout over weeks using machine learning methods," in *Proc. EMNLP Workshop Anal. Large Scale Social Interact. MOOCs*, Oct. 2014, pp. 60–65.
[23] J. Liang, C. Li, and L. Zheng, "Machine learning application in MOOCs: Dropout prediction," in *Proc. 11th Int. Conf. Comput. Sci. Edu. (ICCSE)*, Aug. 2016, pp. 52–57.
[24] S. Goel, A. S. Sabitha, T. Choudhury, and I. S. Mehta, "Analytical Analysis of Learners' Dropout Rate with Data Mining Techniques," in *Proc. ICETEAS*, Nov. 2018, pp. 583–592.
[25] C. G. Brinton and M. Chiang, "MOOC performance prediction via clickstream data and social learning networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2299–2307.
[26] J. Chen, J. Feng, X. Sun, N. Wu, Z. Yang, and S. Chen, "MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine," *Math. Problems Eng.*, vol. 2019, pp. 1–11, Mar. 2019.
[27] H. Li, C. Lynch, and T. Barnes, "Early prediction of course grades: Models and feature selection," in *Proc. 11th Int. Conf. Educ. Data Mining (EDM)*, Dec. 2018, pp. 492–495.
[28] L. Qiu, Y. Liu, and Y. Liu, "An integrated framework with feature selection for Dropout prediction in massive open online courses," *IEEE Access*, vol. 6, pp. 71474–71484, 2018.
[29] B. Gelman, M. Revelle, and C. Domeniconi, "Acting the same differently: A cross-course comparison of user Behavior in MOOCs," in *Proc. Int. Conf. Educ. Data Mining*, Raleigh, NC, USA, Jun. 2016, pp. 376–381.
[30] M. L. Bote-Lorenzo and E. Gómez-Sánchez, "Predicting the decrease of engagement indicators in a MOOC," in *Proc. 7th Int. Learn. Anal. Knowl. Conf.*, Mar. 2017, pp. 143–147.
[31] C. Tang, Y. Ouyang, W. Rong, J. Zhang, and Z. Xiong, "Time series model for predicting Dropout in massive open online courses," in *Proc. AIED*, London, U.K., Jun. 2018, pp. 353–357.
[32] J. Chen, G.-Q. Zeng, W. Zhou, W. Du, and K.-D. Lu, "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization," *Energy Convers. Manage.*, vol. 165, pp. 681–695, Jun. 2018.
[33] S. Mehtab and J. Sen, "Stock price prediction using convolutional neural networks on a multivariate timeseries," in *Proc. Nat. Conf. Mach. Learn. Artif. Intell.*, New Delhi, India, Jan. 2020, pp. 1–7.
[34] B. B. Sahoo, R. Jha, A. Singh, and D. Kumar, "Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting," *Acta Geophys.*, vol. 67, no. 5, pp. 1471–1481, Jul. 2019.
[35] F. Wang and L. Chen, "A nonlinear state space model for identifying at-risk students in open online courses," in *Proc. Int. Conf. Educ. Data Mining*, Raleigh, NC, USA, Jun. 2016, pp. 527–532.
[36] 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 Sci. Technol.*, vol. 25, no. 3, pp. 336–347, Jun. 2020.
[37] H. Cui, V. O. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," in *Proc. Conf. DBLP*, Jul. 2006, pp. 1265–1270.
[38] Q. Cheng, P. K. Varshney, and M. K. Arora, "Logistic regression for feature selection and soft classification of remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 491–494, Oct. 2006.
[39] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multilabel naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, Sep. 2009.
[40] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 67, pp. 93–104, Jan. 2012.
[41] Y. Liu and Y. F. Zheng, "One-against-all multi-class SVM classification using reliability measures," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2005, pp. 849–854.
[42] X. Chen, "Research on machine learning algorithms based on the background of big data," *Comput. Products Circulat.*, vol. 3, p. 85, Mar. 2020.
[43] W. Wang, H. Yu, and C. Miao, "Deep model for dropout prediction in MOOCs," in *Proc. 2nd Int. Conf. Crowd Sci. Eng. ICCSE*, Jul. 2017, pp. 1–7.
[44] X. Sun, N. Wu, L. Zhang, C. J. Chen, and J. Feng, "Deep learning-based dropout rate prediction method for MOOCs," *Comput. Eng. Sci.*, vol. 41, no. 5, pp. 893–899, May 2019.

YAFENG ZHENG was born in Luoyang, Henan, China, in 1979. She received the B.S. degree in computer education from Henan Normal University, in July 2002, and the M.S. degree in computer applications from Northwestern University, in June 2005. She is currently an Associate Professor. Since July 2005, she has been engaged in teaching and research of computer science. Her research interests include educational data mining and analysis, and data visualization.

ZHANGHAO GAO was born in Baoji, Shaanxi, China, in 1997. He received the B.S. degree in information management and information systems from the Southwest University of Science and Technology, in June 2019, and the master's degree. He studied computer applications at the Henan University of Finance and Economics and Law, in September 2019. His research interests include data mining and analysis and data visualization.

YIHANG WANG was born in Xuchang, Henan, China, in 1997. He received the bachelor's degree in software engineering from the Henan University of Finance and Economics and Law, in June 2019, and the master's degree. He studied at Jiangnan University, in September 2019, with a major in software engineering. His research interest includes image recognition.

QIAN FU was born in Jinhua, Zhejiang, China, in 1978. He is currently an Associate Professor with the Department of Education, Beijing Normal University, where he is also the Head of the Creators Education Laboratory. He is also the Head of the Research and Development Team, Mixly and MixIO, open source creators tools.