Research Article

Dropout Rate Prediction of Massive Open Online Courses Based on Convolutional Neural Networks and Long Short-Term Memory Network

Xingqiu Tang,1 Hao Zhang,1,2 Ni Zhang,1 Huan Yan,3 Fangfang Tang,1 and Wei Zhang4

1Institute of Education Guizhou Normal University, Guiyang 550025, China
2Guizhou Provincial Educational Governance Modernization Research Center, Guiyang 550025, China
3Baiyun District Vocational and Technical School, Guiyang 550000, China
4The Chinese University of Hong Kong, China

Correspondence should be addressed to Hao Zhang; zh@gznu.edu.cn

Received 23 March 2022; Revised 17 April 2022; Accepted 22 April 2022; Published 16 May 2022

Academic Editor: Mian Ahmad Jan

Copyright © 2022 Xingqiu Tang 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.

Massive open online courses (MOOC) is characterized by large scale, openness, autonomy, and personalization, attracting increasingly students to participate in learning and gaining recognition from more and more people. This paper proposes a network model based on convolutional neural networks and long short-term memory network (CNN-LSTM) for MOOC dropout prediction task. The model selects 43-dimensional behavioral features as input from students’ learning activity logs and adopts the CNN model to automatically extract continuous features over a period of time from students’ learning activity logs. At the same time, considering the time sequence of students’ learning behavior characteristics, a MOOC dropout prediction model was established by using long short-term memory network to obtain students’ learning status at different time steps. The algorithm proposed in this chapter was trained and evaluated on the public dataset provided by the KDD Cup 2015 competition. Compared with the dropout prediction methods based on LSTM and CNN-RNN, the model improved the AUC by 2.7% and 1.4%, respectively. The result in this paper is a good predictor of dropout rates and is expected to provide teaching aid to teachers.

1. Introduction

Education is no longer a one-time event, but a lifetime experience. In this context, digital learning e-learning develops rapidly. Among them, MOOC is the most representative online education platform. Since its birth in 2012, it has attracted nearly ten million learners around the world to participate in various courses on the platform [1]. MOOC is characterized by large scale, openness, autonomy, and personalization, attracting increasingly students to participate in learning and gaining recognition from more and more people. The education and teaching reform brought by MOOC has become the general trend of the deep integration of information technology and education and teaching. More and more students choose MOOC for learning due to the outdated and unguaranteed quality of the courses in the traditional teaching mode. It can not only gather high-quality educational resources, provide an open and sharing platform, and enhance the value of higher education but also complement flipped classroom teaching method to greatly improve students’ learning experience. There is still much to be done about one of the main unresolved issues surrounding MOOC: student dropouts. MOOC, which are online and open, are usually large and, in theory, have no limit on how many people can sign up, allowing anyone to join or leave without penalty. Therefore, the dropout rate reaches more than 90% [2]. While high dropout rates are often cited as a scale trade-off, it is a potential impediment to the growth of MOOC.
2. Literature Review

Recently, with the storage of student behavior data in online education platforms, the disclosure of relevant datasets and the rise of machine learning (ML), researchers’ research enthusiasm is greatly stimulated, and the research on MOOC dropout prediction task based on machine learning algorithm is also promoted. Researchers determine different behavioral characteristics based on prior knowledge in the field of education and then build various dropout prediction models based on behavioral characteristics.

For example, in 2014, Jiang et al. [3] used logistic regression (LR) as a classifier to predict whether students could complete MOOC courses based on the homework performance of students in the first week and social interaction in MOOC. Lakkaraju et al. [4] proposed a machine learning framework to identify students who are likely to fail to finish their high school studies on time, mainly using several classification algorithms and evaluating them with relevant indicators that school administrators attach importance to. Amnueypornsakul et al. [5] used the information mined from clickstream data to study the prediction model of learner dropout in a given week, mainly using support vector machine (SVM) to build the model, and the results showed that relevant characteristics of attempted submission behavior and interaction characteristics of various course components were valid predictors for a given week. Sinha et al. [6] took students’ video lecture clickstream operation as a cognitively credible advanced behavior and constructed a quantitative information processing index to help teachers better analyze the reasons for learners’ dropout behavior and unsatisfactory academic performance. Taylor et al. [7] designed multiple classifiers for 4 types of learners (passive collaborators, wiki contributors, forum contributors, and full collaborators) with 27 features, including LR, SVM, decision tree, DT, and hidden Markov model; and the 14-week behavior data of the four types of learners were trained and evaluated, respectively. Kloft et al. [8] used SVM to the extracted features to predict whether students actively participated in the last week of the course. Sharkey and Sanders [9] used the data and logs generated by courses to predict the dropout situation of MOOC and analyzed 15 different predictive feature subsets and their respective weight values. Bailey et al. [10] modified LR by adding regularization terms to reduce the difference in dropout probability among consecutive weeks. Liang et al. [11] used user behavior log data and GBDT model to predict the possibility of students dropping out in the next ten days. Feng and Li [12] employed a nonlinear state space model to predict whether students drop out or not by combining click stream data of different weeks. Lu et al. [13] used several methods of ML to build a sliding window model for predicting the probability of students dropping out of school in a certain process. Hagedoorn and Spanakis [14] conducted an in-depth study on MOOC dropout by using LR, random forest (RF), and AdaBoost methods in combination with students’ static and dynamic behavior characteristics. Carmen et al. [15] used Bayesian networks (BNs) to predict dropout. BNs provide a concise way to express knowledge and an adequate method to interpret and contextualize data without advanced statistical knowledge. BNs are considered to be an appropriate model for use in the context of learning analysis; Yi et al. [16] predicted student performance through multicore support vector machine combined with optimization algorithm. All in all, there are plenty of machine learning methods that have proven effective in predicting dropout rates [17–20].

The successful application of deep learning (DL) also provides some new study ideas for MOOC dropout prediction. In terms of dropout prediction, Fei and Yeung [21] employed RNN models to predict the dropout rates. In 2016, Tang et al. [22] used RNN model with LSTM units to predict the next interaction of learners. Wei et al. [23] used convolutional neural network (CNN)-based RNN model, in which CNN realizes automatic extraction of features, and RNN takes into account the influence of different time factors on dropout. The proposed method achieves 87.42% AUC. Feng et al. [24] found the high correlation between dropout rates of different courses and the influence of friends’ dropout behaviors based on statistical data.

In terms of MOOC dropout prediction, existing research methods generally rely on manual feature extraction with high labor and time costs. Feature extraction strategies depend on the characteristics of datasets. Feature extraction strategies that are effective for one dataset may be ineffective for another. This makes the MOOC dropout prediction model unstable and have poor performance. Therefore, finding a more general and automatic feature extraction method that does not rely on domain knowledge to obtain feature sets can make the MOOC dropout prediction results more stable and accurate.

Considering that CNN has the feature of automatic feature extraction, the complex process of manual feature extraction is avoided. LSTM model is more suitable for handling with the field of time series. Therefore, based on CNN and LSTM, two deep learning networks, this paper conducts research on MOOC dropout prediction model.

The innovations of this paper can be concluded as follows: based on the analysis and summary of the current research status of dropout prediction, this paper focuses on the learning behavior records of learners in the MOOC learning environment, mainly studying the automatic extraction of effective features including students’ behaviors from online education datasets and then using the deep learning network model to achieve the dropout prediction task. The input of this method is the behavioral characteristics of online education, and the output is whether each student drops out in different time steps. The method consists of the following two steps: feature extraction and dropout prediction. The data collected from the online education platform are first preprocessed and then processed into a form of input that can be recognized by the network. Finally, each student’s learning status (dropping out or completing the course) is predicted in different time steps.

3. Method

3.1. Dropout Prediction Algorithm Based on CNN-LSTM Network Model. This chapter proposes a dropout prediction
model based on convolutional neural network and long short-term memory (CNN-LSTM) network, aiming at the existing MOOC dropout prediction model which relies on feature engineering to extract features and has some problems such as unstable prediction effect and poor performance. The convolutional neural network has the end-to-end characteristics, which is a more general and simple automatic feature extraction method. The main innovation of LSTM is its storage unit $C_t$, which basically acts as an accumulator for state information. The cells are accessed, written, and cleared by several self-parameterized control gates. Whenever there is a new input, if the input gate is activated, its information is accumulated into the cell. Therefore, the use of CNN and LSTM can make the prediction of MOOC dropout problem more stable and accurate.

3.2. Construction of CNN-LSTM Network Model for Dropout Prediction. This chapter uses CNN-LSTM network model to predict and analyze the dropout problem. The model includes a layer of convolution and a layer of pooling for automatic feature extraction. Considering the high efficiency of CNN in processing high-dimensional features and the effectiveness of feature extraction, this paper adopts single-layer CNN to automatically extract learners’ behavioral features, train the optimal parameters of the model, and automatically screen out effective behavioral features. The LSTM network is used as a dropout prediction model to determine whether students drop out in different time steps. Figure 1 introduces the structure of the CNN-LSTM network.

3.2.1. Representation of Learning Behavior Characteristics. The weekly original behavior records of students are taken as the input of CNN-LSTM network. To convert the original behavior records into a vector form that can be recognized by the network, this chapter carries out feature representation of learning behaviors. Set the length of the weekly original behavior record of students to a fixed value $n$, and then, the original behavior record of students $S$ can be expressed as

$$S = s_1s_2s_3\cdots s_n,$$

where $s_i (i \in [1, 2, 3, \cdots, n])$ represents the vector $[r_{i1}, r_{i2}, \cdots, r_{id}] \in \mathbb{R}^{d \times |V_i|}$ ($V$ represents all behavior features) with the size of $d$ dimension corresponding to the $i$-th behavioral characteristic, and then, the original behavior record $S$ can be expressed as a matrix, where the size of the matrix is $n \times d$:

$$S = \begin{bmatrix}
    r_{11} & r_{12} & \cdots & r_{1d} \\
    \cdots & \cdots & \cdots & \cdots \\
    r_{n1} & r_{n2} & \cdots & r_{nd}
\end{bmatrix}. \tag{2}$$

3.2.2. Convolution Layer of Dropout Prediction Model. Graves and Schmidhuber [25] found in 1962 that the cat cortex contained a unique network structure, which could effectively reduce the computational complexity of the traditional BP neural network, and CNN was proposed on the basis of this structure. In order to make the network structure have scale, displacement, and deformation invariance, CNN combines the three structures of local perception, weight sharing, and downsampling to improve the accuracy and speed of operation. CNN has a clear end-to-end architecture that can automatically extract low-level and high-level behavior characteristics. Recently, CNN has become the most advanced technology: compared with the previous methods of manually extracting features, the performance with feature extraction based on CNN has been obvious improved [26]. Therefore, after the representation of behavioral characteristics, the original behavior records of students can be represented by vector matrix. In order to generate new behavioral features, these matrices will be used as the input of the convolution layer to realize the process of convolution calculation, which can realize the automatic extraction of effective behavioral features, avoiding the complex steps of feature extraction and feature optimization.

In the process of convolution calculation, the filter used is $w \in \mathbb{R}^{h \times d}$, where $h$ is the size of the convolution window and the number of continuous features covered in the convolution operation and $d$ is the dimension of the behavior feature vector. Its function is to slide a window with a size of $h$ on the behavior vector matrix, and every slide of the convolution window in the convolution calculation will generate a feature $c_i$, and the calculation formula is as follows:

$$c_i = f(w \ast s_{i+h-1} + b), \tag{3}$$

where $w$ and $b$ are the hyperparameters in the CNN model and $f$ is the nonlinear activation function.

The commonly used activation functions are ReLU, Tanh, sigmoid, and so on. $s_{i+h-1}$ is the matrix between the corresponding feature vectors of $i$ and $i + h - 1$. $s_{i+h-1}$ can be expressed as formula (4), $v_i$ is each eigenvector, and $\oplus$ is a joint operator. The convolution kernel with $h \times d$ convolution window size can generate $l-h+1$ features, named feature graph $c$, and calculated by

$$s_{i+h-1} = v_1 \oplus v_{i+1} \oplus v_{i+2} \oplus \cdots \oplus v_{i+h-1}, \tag{4}$$

$$c = [c_1, c_2, \cdots, c_{l-h+1}]. \tag{5}$$

In formula (4), feature graph $c \in \mathbb{R}^{l-h+1}$. Training CNN networks is equivalent to training each convolution filter, which has high activation for specific features to achieve classification or detection of CNN network. In the CNN model, local features of student behavior records can be extracted from convolution operations, while the extraction process of multiangle behavior combination features can be realized through multiple convolution operations of different sizes.

In order to retain more useful features, pooling method is usually adopted after the convolution layer, which is a feature selection method for the feature graph obtained from the convolution layer. Pooling layer includes commonly used max pooling and average pooling. Maximum pooling is the selection of the maximum in feature graph $c$, which can be considered the most efficient feature. Average pooling
is the average in characteristic pattern $c$. However, maximum pooling is commonly used in NLP tasks, and Zhang et al. [27] also believe that maximum pooling is more suitable for text classification. Therefore, this chapter assumes that the characteristics generated by the convolutional layer $\tilde{c}$ find can be obtained based on max pooling. Bearing diagram in max pooling can be expressed as follows:

$$\tilde{c} = \max \{ c_i \} , \forall i = 1, \cdots , (l - h + 1).$$

(6)

It can be seen from formula (6) that the length of feature vector is no longer related to the feature length after max pooling operation; that is, the effective behavior characteristics of each student are unified. As a matter of fact, the convolution operation extracts local characteristics of student behaviors, and the combination of behavioral characteristics is achieved through max pooling. Therefore, the feature graph $\tilde{c}$ is the most effective feature. Generally, there are many convolution windows of the same size, and the windows after convolution and pooling can be expressed in the feature vector corresponding to the pooling layer in

$$h = \tilde{c}^0 \bigoplus \tilde{c}^1 \bigoplus \cdots \bigoplus \tilde{c}^l.$$  

(7)

As per each convolution window, several $\tilde{c}$ matrices will be generated, and the maximum value extracted will be fully wired, then placed into the activation function (ReLU), and finally entered into the LSTM layer.

3.2.3. Sequence Processing Layer of Dropout Prediction Model. Long short-term memory model (LSTM) is proposed in 1997 to solve the problem of gradient disappearance [28]. LSTM has an input gate $i_t$, an output gate $o_t$, and a forgetting gate $f_t$. $f_t$ determines which cell state information needs to be discarded; $i_t$ determines what new information is stored in the cell state. For general sequence modeling, LSTM, as a special RNN structure, has been proved to be stable and powerful in establishing long-term dependence in previous studies [28–31].

Figure 2 shows how to calculate the output state $h_t$ and the updated memory state $C_t$ at time $t$ of a memory cell. $C_{t-1}$ refers to the status of the memory unit at $t - 1$, $C_t$ refers to the updated state of the memory unit at $t$, $\tilde{C}_t$ refers to the state information that needs to be added at $t$, and $h_{t-1}$ and $h_t$ are output values at $t - 1$ and $t$, respectively. $x_t$ represents the behavioral feature vector of students at time $t$. Moreover, if the forgetting gate $f_t$ is opened, it is possible to “forget” the past unit state $C_{t-1}$ in the process. Whether the latest unit output $C_t$ will propagate to the final state, $h_t$ is further controlled by the output gate $o_t$. An advantage of using memory cells and gates to control information flow is that gradients will be captured in cells and can be prevented from disappearing too quickly, which is a key problem faced by ordinary RNN models [28, 29]. The corresponding calculation formula is as follows:

Input gate:

$$i_t = \sigma(W_{ix}h_{t-1} + b_i)$$

(8)

Forgetting gate:

$$f_t = \sigma(W_{fx}h_{t-1} + b_f)$$

(9)

Memory unit:

$$\tilde{C}_t = \tanh (W_{cx}h_{t-1} + b_c)$$

(10)

Output gate:

$$o_t = \sigma(W_{ox}h_{t-1} + b_o)$$

(11)

State update:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

(12)

Hidden layer output:

$$h_t = o_t \cdot \tanh (C_t)$$

(13)

In the above equation, $x_t$ is behavior characteristic vector, and the input students $b_f, b_i, b_o$, and $b_c$ are the offset
values of different gates in LSTM; \( C_t \) is the cell state vector and \( \sigma \) represents the sigmoid function. The features of timing information generated by LSTM in each step length are fully connected, then put into activation function (ReLU), and finally put into sigmoid classifier for training. The LSTM network is used as a classifier for predictive models to determine when students drop out in different time steps.

4. Experimental Results and Analysis

4.1. Definition of MOOC Dropout. In the MOOC dropout prediction research, there is no strict definition of “dropout.” Previously, different researchers had different definitions according to their own views and interests, making it difficult to conduct parallel comparisons. Table 1 lists the definitions of dropout commonly used today. Definition_2 is used for the definition of dropout in this article.

4.2. Dropout Prediction Dataset. The CNN-LSTM dropout prediction model proposed in this chapter was trained and evaluated on the KDD Cup 2015 dataset. The dataset comes from Xuetang Online, one of China’s largest MOOC platforms. This dataset records the course selection records and behavior records of some learners of “Xuetang Online” from October 27, 2013, to August 1, 2014. The statistical information of this dataset is shown in Table 2. The dataset provides 120,542 registration activity logs from 79,186 students in 39 courses. Each activity consists of six fields: username, course number, timestamp, source of action, event and object, and each student has taken at least one course. There are two main sources of learner behavior recording information: browsers and servers.

Each student participates in the course by watching videos, trying to solve problems, participating in course modules, etc. Each student has a label in the course every week, it represent the learner pass or fail the proposed course in that week. The label of dropout is “1,” and the label of nondropout is “0.” In this chapter, the sample labeled “1” is regarded as a positive example. Table 3 shows the seven events for learning behavior logs.

The number of online learners refers to the number of learners per week. This chapter takes five courses with more than 4000 participants as examples for statistical analysis, and the results are shown in Figure 3. At the beginning of the course, the number of online students of these five courses was large, but as the course went on, the number changed from fluctuation to sharp decline. The number of participants in the early stage of the course is much higher than that in the later stage of the course, which also shows the widespread dropout phenomenon in online education.

In this chapter, the course 13 with the largest number of students (12,004 students) is selected for illustration, as shown in Figure 4, where \( D \) represents the students who dropped out and \( R \) represents the students who continued the course. As you can see, as the course goes on, more and more students drop out of class. From the beginning of the course to the fourth week after, the dropout rate reached the peak, as high as 86.77%. The overall number of students who dropped out in the first three weeks fluctuated little, which may be because this stage belongs to the stage of students getting to know the course. After a period of learning, students will make choices according to whether the courses match their needs, which leads to a certain increase in the number of dropouts in the second two weeks of the courses.

4.3. Evaluation Index of Dropout Prediction. Before introducing evaluation indicators for MOOC dropout prediction, the concept of confusion matrix is first introduced, as shown in Table 4.

According to Table 4, each evaluation indicator is defined as follows.

1. Precision (P): refers to the proportion of dropouts correctly predicted among all samples whose predicted result is dropout, namely,

\[
P = \frac{TP}{TP + FP}
\]

2. Recall (R): refers to the proportion of dropouts correctly predicted among all samples with actual values of dropouts:

\[
R = \frac{TP}{TP + FN}
\]

3. F1 value: harmonic average value of the precision and recall rate. F1 value is close to the smaller value of the precision and recall rate, namely,

\[
F1 = \frac{2TP}{2TP + FP + FN} = \frac{2PR}{P + R}
\]

The ROC takes the FPR as the x-axis and the true rate (TPR) as the y-axis. The area under the ROC curve, or AUC, can be used to evaluate the performance of the classifier by adjusting the threshold value of the classification. The value of AUC ranges from 0 to 100%. The larger AUC is, the stronger the generalization ability of the model is. One of the
The most important characteristics of AUC scoring is that it is not affected by the category ratio of data samples and is particularly suitable for the evaluation index of classification of categorically unbalanced datasets [31]. The dropout prediction problem is actually a dichotomous problem. To evaluate the generalization ability of each MOOC dropout prediction model, this chapter uses precision (P), recall (R) [32], F1 value [2], and AUC [21] as evaluation indicators for dropout prediction.

4.4. Feature Selection of Learning Behavior. In this chapter, the dropout prediction problem is regarded as a sequential marker prediction problem, in which both input features and output markers are expressed in a sequential form. As described in 3.2, the time step of the prediction task is set in weeks. Based on this assumption, learner input features within each time step are generated from all learning activities in a week. According to the dropout definition introduced, this chapter adopts the stacked week characteristics of students for training and prediction model.

Known student K behavior characteristics sequence in T weeks \((x_{k,1}, x_{k,2}, \ldots, x_{k,T})\) and the corresponding dropout labels sequence \((y_{k,1}, y_{k,2}, \ldots, y_{k,T})\), where T means the number of weeks that student takes part in the course, as shown in Figure 5. For the current week \(j\), if student K has activities in the next week, the dropout label of week \(j\) is considered to be \(y_{k,j} = 0\); otherwise, \(y_{k,j} = 1\). In this section, data is organized in the three-dimensional space of object, event, and time, and all combined features of object, event, and time are generated by slicing and partitioning operations. For example, to count the number of navigation events of students every week, first, cut the data of students in the object dimension; then, select the navigate event in the event space to generate a time series representing the navigate event; finally, generate weekly statistics for navigation events. Considering the seven types of events and their respective combination events, this chapter extracted 43 typical behavior characteristics for each week and expressed them as \(N\)-dimensional vectors \(x_{k,T} \in \mathbb{R}^N\). The specific behavior characteristics are introduced in Table 5.

4.5. Training Process Design. This section implements the CNN-LSTM dropout prediction model based on Keras, a Python library for implementing DL models. In the experiment, the logarithmic loss function was used to train the parameters of the model, the optimization function adopted an adaptive learning rate method, and the batch size was set to 16, and the model ran iteratively for 10 periods.

The CNN-LSTM proposed in our chapter is compared with three classical machine learning models (SVM, LR, and DT), LSTM, and CNN-RNN.

(1) Support vector machine (SVM) [8]: considering the nonlinear relationship among data, SVM classifier is used in the experiment in this section. A SVM classifier for weekly training is used to predict whether weekly learners will drop out. SVM is trained by svmtrain function in MATLAB, and Gaussian kernel function is used as the kernel.
function. After obtaining the trained SVM model, svmclassify function is used to realize the dropout prediction task.

(2) Logistic regression (LR) [7]: LR is a technique widely used to predict dichotomies [33] and is also often used to predict MOOC dropout problems [10]. Logistic regression uses logistic function to standardize the output value so that the output value y is in the interval (0,1). The logistic function of input $x_i$ under parameter $\theta$ is as follows:

$$h(x) = \frac{1}{1 + e^{-\theta x}}$$

(17)

The specific implementation of LR is trained by the glmfit function in MATLAB. After the trained logistic regression model is obtained, the specific dropout prediction task is realized by the glmval function.

(3) Decision tree (DT) [34]: it is a common method of supervised learning. The method of this tree structure has multiple internal nodes, multiple branches, and leaf nodes, where each internal node represents a test attribute, while branches represent test output and leaf nodes represent corresponding categories [35].

In the data preprocessing of the above three models, data were normalized, and each comparative experiment was iterated ten times. The purpose of normalization is to make the characteristics of different dimensions have a certain numerical comparison and accelerate the speed of gradient descent to obtain the optimal solution.

5. Experimental Results and Analysis of Dropout Prediction

The input of CNN-LSTM model adopted in this chapter is the weekly behavior characteristic matrix of each student. There are 5 matrices in total, and the size of each matrix is $1 \times 43$. According to experimental verification, Table 6 introduced the specific parameter settings of CNN-LSTM network. To prevent overfitting, set the dropout value to 0.3 to avoid the overfitting. Output of our proposed model is a value between 0 and 1, representing the probability of students dropping out of relevant courses. The threshold value is set at 0.5 in the experiment. When the output result of the model is greater than 0.5, the label of students is set as 1, indicating that they have dropped out; otherwise, it is set as 0, indicating that they have not dropped out.

Considering the influence of data volume on the training of prediction model, this section selects 5 courses from the courses with more than 4000 students for the experiment and trains one model for each of the five courses. Considering the sparsity of data, 60% of students who participated in the course were regarded as training set and the remaining 40% as test data. For the three baseline classifiers, SVM, LR, and DT, weekly feature training and testing were conducted for the course, respectively. For LSTM, CNN-RNN, and CNN-LSTM models, this section adopts stacking cycle characteristics $X = (x_1, x_2, \ldots, x_t, \ldots, x_N)$ experiment, in which the $x_t = (x_{t1}, x_{t2}, \ldots, x_{tT})$; these characteristics can be captured from the beginning of the course.

Table 7 describes the prediction result with several methods and evaluation indexes. We can found in Table 6 that the CNN-LSTM dropout prediction model outperformed all
the benchmark method such as some popular ML method like SVM and DT. Moreover, the result in the indicators about F1, AUC, and precision, the proposed method gained the best result. The LSTM method performs better than the proposed method in recall but with the worse result in both AUC and precision. Considering that CNN has the feature of automatic feature extraction, the complex process of manual feature extraction is avoided. LSTM model is more suitable for handling with the field of time series. Therefore, the convolutional neural network has the end-to-end characteristics, which is a more general and simple automatic feature extraction method.

Combined with the comparison of accuracy and AUC values of each prediction model in Figure 6, we can find that the CNN-LSTM dropout prediction method could predict the dropout situation of students at different stages of different courses. The experimental results show that in the first two weeks of the course, learners will try to learn at the contact stage and choose to give up learning once they find that some factors such as the course content are not suitable for them. However, in the last two weeks of the course, learners are most likely to drop out, and many learners will stop participating in the course because of the final assessment. Many learners pay more attention to the experience of learning process rather than a final course evaluation scores. If students timely take some auxiliary measures at the end of the semester, such as review knowledge, to do more exercises, or increase the number of video watching, these not only help to reduce the dropout rate of learners, at the end of the course, but also help them complete the final assessment tasks to get the final result.

Table 5: Learning behavior characteristics of each student in week T.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Total number of activities in week T</td>
</tr>
<tr>
<td>X2–X8</td>
<td>Total number of 7 different types of activities in week T</td>
</tr>
<tr>
<td>X9</td>
<td>Total number of events in week T</td>
</tr>
<tr>
<td>X10</td>
<td>Total average number of events over the T week</td>
</tr>
<tr>
<td>X11</td>
<td>Total number of active days in week T</td>
</tr>
<tr>
<td>X12</td>
<td>Total time consumed in week T</td>
</tr>
<tr>
<td>X13–X16</td>
<td>Total number of access objects by browser, try to solve problems, close pages, videos watched in week T</td>
</tr>
<tr>
<td>X17–X21</td>
<td>Total number of use the server to browse the course, visit the object, try to solve the problem, discuss, browse the Wikis in week T</td>
</tr>
<tr>
<td>X22–X23</td>
<td>Total number of servers and browsers used, respectively, in week T</td>
</tr>
<tr>
<td>X24–X30</td>
<td>The average number of 7 different types of activities over T week</td>
</tr>
<tr>
<td>X31</td>
<td>The average number of activities in T weeks</td>
</tr>
<tr>
<td>X32–X35</td>
<td>The average number of access objects by browser, try to solve problems, close pages, videos watched in week T</td>
</tr>
<tr>
<td>X36–X40</td>
<td>The average number of use the server to browse the course, visit the object, try to solve the problem, discuss, browse the Wikis in week T</td>
</tr>
<tr>
<td>X41–X42</td>
<td>The average number of servers and browsers used, respectively, in week T</td>
</tr>
<tr>
<td>X43</td>
<td>The average time consumed in week T</td>
</tr>
</tbody>
</table>

Table 6: CNN-LSTM network parameter setting table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution kernel size</td>
<td>1 × 2</td>
</tr>
<tr>
<td>Pool size</td>
<td>1 × 3</td>
</tr>
<tr>
<td>Number of filters</td>
<td>128</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.3</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>10</td>
</tr>
</tbody>
</table>
6. Conclusion

This paper proposes a network model based on CNN-LSTM for MOOC dropout prediction task. The model selects 43-dimensional behavioral features as input from students’ learning activity logs and adopts the convolutional neural network model to automatically extract continuous features over a period of time from students’ learning activity logs. Meanwhile, considering the time sequence of students’ learning behavior characteristics, a MOOC dropout prediction model was established through LSTM to obtain students’ learning status at different time steps. The algorithm
proposed in this paper was employed on the public dataset supported by the KDD Cup 2015 competition. Compared with the dropout prediction methods based on LSTM and CNN-RNN, the model improved the AUC by 2.7% and 1.4%, respectively.

**Data Availability**

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

**Conflicts of Interest**

The authors have declared to have no competing interests.

**Acknowledgments**

This work is support by the NSFC “Research on the Construction and Application Strategy of New Training Models for Rural Teachers Supported by Artificial Intelligence” (Grant Number: 72164004), the University-Level Doctor Starting Fund Project “Construction and Application of Internet Smart Campus based on Big Data” (Grant Number: 051704111904), and the Philosophy and Social Science Planning Topic of Guizhou Province “Construction Research on Rural Teachers Team of Guizhou Province” (Grant Number: 21GZYB54).

**References**


architectures,” *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, 2005.


