Predicting MOOCs Dropout with a Deep Model

Fan Wu, Juntao Zhang, Yuling Shi, Xiandi Yang(✉), Wei Song, and Zhiyong Peng

School of Computer Science, Wuhan University, Wuhan, Hubei, China
{fan2013,juntaozhang,sylyjs,xiandiy,songwei,peng}@whu.edu.cn

Abstract. With the deep integration of information technology and education, Massive Open Online Courses (MOOCs) become popular and receive high attention. Although MOOCs are popular among people, it faces a great challenge—the high dropout rate, which affects its development. Predicting the dropout rate in advance can take relevant measures to avoid as many dropouts as possible. Traditional machine learning classification prediction and single sequence label prediction methods are difficult to accurately predict complex user behaviors. To solve the problem, in this paper, we perform a deep analysis of user learning behavior to find that user activity shows a periodic distribution based on the time of course release. In addition, user gender and course category also affect users' behaviors. To this end, we propose a deep model based on recurrent network which combines the influence factors of cyclical historical behavior on the basis of a single sequence of events. Meanwhile, we combine behavior periodicity with attention mechanism to select effective historical behavior impact factors. Then we embed the attributes of user and course to predict the dropout rate. Finally, experiments on different data sets show that our approach performs better than the state-of-the art methods.

Keywords: Dropout in MOOCs · Period · Attention

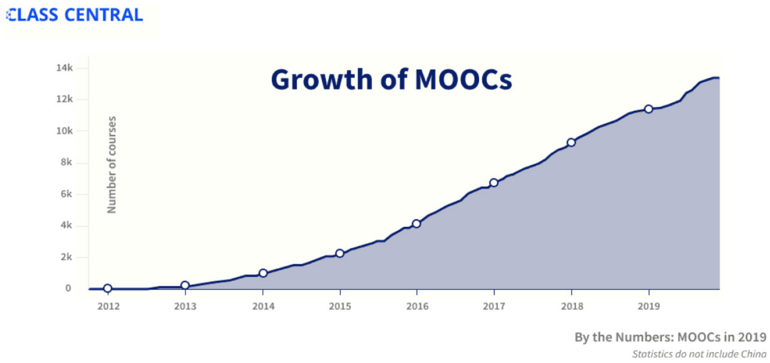
1 Introduction

With the deep integration of information technology and education, large scale online education is developing rapidly under the support of artificial intelligence and big data technology. The concept of Massive Open Online Courses (MOOCs) [1, 2] first appeared in 2008, and the learning revolution represented by it is strongly impacting the ecology of traditional education. In 2012, three educational platforms, Coursera, Udacity and edX emerged, causing a MOOC wave around the world and severely impacting traditional education model. As a result, the MOOCs wave broke out in China in 2013, and top domestic universities cooperated with edX and Coursera to create a domestic online education platform-XuetangX. MOOCs, led by XuetangX, is also rapidly developing [3]. In recent years, MOOC learning has become more and more popular. Due to its strong advantages, it has broken the time and space limitations of traditional education mechanisms and an electronic device connected to the Internet can complete the course. According to Class Central's annual report[1], by the end of 2019, more than 900 colleges

[1] https://www.classcentral.com/report/moocs-stats-and-trends-2019/.

Z. Huang et al. (Eds.): WISE 2020, LNCS 12343, pp. 488–502, 2020.
https://doi.org/10.1007/978-3-030-62008-0_34

and universities had opened 135,000 MOOC courses, excluding China. The trend of courses offered on the MOOC platform from 2012 to 2019 is shown in Fig. 1, and this number is still growing rapidly. In recent months, during special virus outbreaks, online education has provided great convenience to the majority of students. The epidemic has brought MOOC to a new climax, MOOC quickly occupied the education market with unstoppable momentum again and led the education revolution.



By the Numbers: MOOCs in 2019

Statistics do not include China

Fig. 1. The trend of the number of courses on MOOC platform from 2012 to 2019

However, with the rapid development of online education, some shortcomings have gradually emerged. The main problem is the occurrence of dropouts. Very few people can actually complete a course to obtain a certificate [4, 5], compared with the compulsory learning mechanism in traditional education, it is the openness of online education and the lack of supervision mechanism which leads to the loss of users. The reasons for users dropping out may be inappropriate learning resources, mismatched learning abilities, incorrect learning methods, or lack of communication between users, resulting in insufficient learning motivation and driving force, etc. [6]. In fact, the domestic average online school dropout rate has now reached 95.5% [3]. Facing the severe challenge, a large number of researchers have studied the learning behavior patterns and preferences of learners from multiple different perspectives and the relationship with the final learning effect [7–9]. User loss is a major challenge for MOOCs, we need to be able to predict the possibility of user dropouts in advance, then analyze the causes and take corresponding measures.

Through the deep analysis of the actual datasets, we find that most courses are published with a fixed time interval and users have a high degree of activity before or after the new course release time, and user learning behavior may be periodic. So this paper proposes a periodic attention mechanism to predict dropout rates. The dropout rate prediction is actually a sequence labeling problem [10] or a time series prediction problem. Most of the existing sequence events are predicted by using Recurrent Neural Network (RNN) or Long Short Term Memory networks (LSTM) as the model. LSTM can also be used for text context sentiment analysis [11]. The method proposed in this paper is based on the prediction of the sequence of events combined with the attention mechanism of the association period, taking the impact of historical behavior into consideration, and combining the two aspects to predict the probability can ensure accuracy.

The main contributions of this paper can be summarized as follows:

- Perform in-depth analysis on user behavior data to find demographic and behavior characteristics that have a greater impact on user behavior. At the same time, we propose a period detection algorithm to find the best user behavior period from the distribution period and structure period, and performing locating the specific target for the attention mechanism selector.
- We propose a deep learning architecture based on recurrent neural network. Take historical behavior as a predictor through the attention mechanism associated with the cycle. Combining sequential and historical behavior to improve model performance.
- Extended experiments are performed on two datasets, at the end of the model prediction, the user and course information are added to make predictions with the support of the dataset. The experimental results prove that our proposed model performs better than several current methods.

The remainder of this paper is organized as follows. In Sect. 2, we systematically review the related works in dropout prediction in MOOCs. After that we take a deep analysis about users' learning activities. Further, we introduce our predicting model in details. Section 5 we apply our model on real datasets and give the descriptions about the experiments. Finally, we conclude our paper.

2 Related Work

In this part, we make a brief summary of the research on the dropout rate prediction in the MOOCs field in the past ten years.

Many researchers study the relationship between learner learning behavior and learning effectiveness from different perspective, they use different mathematical models to predict learners' short-term learning behavior and long-term learning effectiveness. In Anderson et al. [12], learners were divided into five categories based on their learning behavior preferences, and learning effects were analyzed based on different learning models. In Kloft et al. [13], a simple linear SVM is used to predict the dropout rate. Taylor et al. [14] applies logistic regression to learn behavior characteristics and predicts student dropouts based on the students' last learning activities in the course. Ramesh et al. [15] used the discussions in the MOOCs forum and the completion of learners' homework to construct a predictive model to study learner dropout behaviors. Balakrishnan et al. [16] proposes a dropout prediction model based on Hidden Markov Model combined with support vector machines. Unlike other studies, Chanchary et al. [17] uses K-means for quantitative analysis and automatically discover inactive students by clustering students in a MOOCs environment. W Xing et al. [18] takes a combination of Bayesian Network and Decision Tree to make predictions. In addition to traditional machine learning, deep learning is also used to predict dropout rates. Fei et al. [19] believes that the prediction of dropout rates is a time series prediction problem, and proposes a temporal model which can complete predictions separately under the different definition of dropouts, they predict by using traditional RNN model with LSTM cell. Wang et al. [20] completes the prediction through a deep neural network, which is a combination of a Convolutional Neural Network and a Recurrent Neural Network. This model can automatically extract features from the original data. Scott et al. [21] adopt Natural Language Processing and other methods to analyze learners' questions and answers on the forum to predict learner completion. By combining learners' statistical information, forum behavior data and learning behaviors, a hidden dynamic factor model is proposed to predict the learning effects of learners by Qiu et al. [22].

At present, for the problem of user dropout rates on MOOCs platform, some traditional machine learning methods are used. Although the operation is simple and widely used, the internal associations of user behavior are not considered. Others use deep learning methods based on recurrent neural networks. Although they have considered the problem as a time series problem but the prediction effect will be limited if the time span is too long. Our proposed method not only introduces the influence of current sequence events, but also combines the influence of historical behavior associated with the potential period of user behavior which can improve the accuracy of prediction to some extent.

3 Datasets and Analysis

3.1 Datasets

The datasets we analyze and use in the laboratory are derived from XuetangX[2] and KDDCUP2015[3].

XuetangX is a Chinese MOOC platform developed by Tsinghua University. It was officially launched on October 10, 2013 and provides online courses to the world. As of now, there are 1800+ courses with a wide range of subject categories. This dataset contains 1,213 courses and 378,273 users. Some courses have a fixed scheduling cycle, and some courses do not have. The second dataset is from the KDDCUP competition in 2015. The KDDCUP is an annual data mining and knowledge discovery competition organized by the ACM knowledge discovery and data mining special interest group. This dataset provides a record of user behaviors within half a year of 39 online courses.

The specific categories of user behavior in the two datasets are: watching videos, doing homework, forum discussions, browsing course pages (navigate), accessing objects (access), and so on. Table 1 is the relevant statistics for these two datasets.

Table 1. Statistics of the datasets

Dataset	Courses	Users	Records
KDDCUP	1213	378273	115078786
XuetangX	39	112448	21552534

[2] http://moocdata.cn/data/user-activity.

[3] http://kddcup2015.com/.

3.2 Analysis

Although each data set contains multiple courses and log records, we actually use some courses and log records for data analysis and experiments.

Figure 2 statistics the user behavior activity in the course. It is calculated from the three types of users: all users in the course, users who did not drop out, and users who dropped out. We can see that when new content is released in a course, it is obvious that user activity is greatly improved. The user's activity changes periodically based on the course release, and the probability of dropping out of a user group with more regular course learning is far less than that of irregular user group.

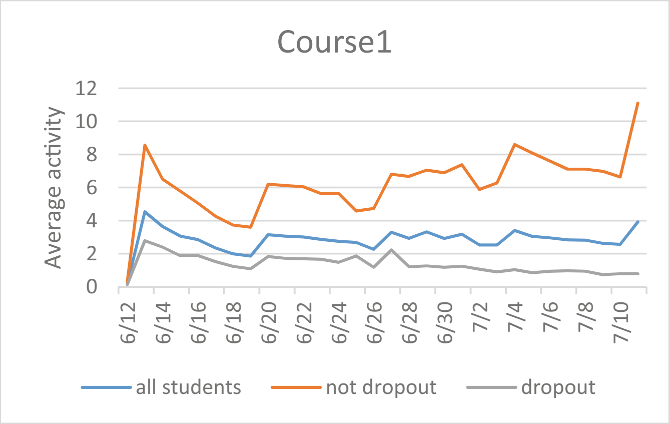


Fig. 2. User behavior distribution

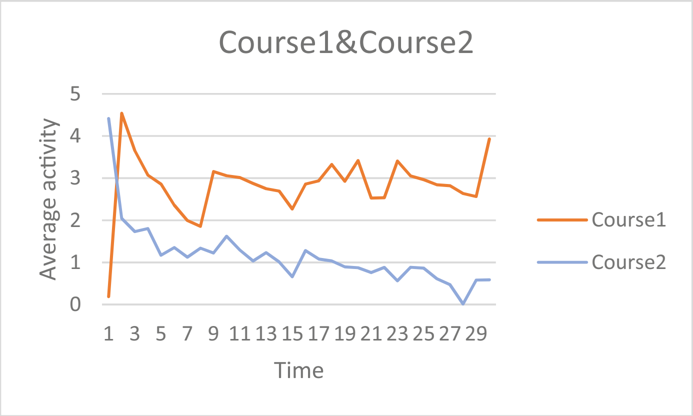


Fig. 3. Comparison of user activity between Course 1 and Course 2

From Fig. 3, the record for Course 1 is from June 12 to July 11, and Course 2 is from January 17 to February 15. We know that compared with Course 1, the release of Course 2 is before and after the winter vacation, and the user activity in Course 2 is significantly lower than that of other courses. It can be seen that the number of user visits during the holidays is sharply lower than usual. During holidays, users rarely participate in learning, so if the course includes holidays, the course publisher need to adjust the course release time reasonably.

It can be seen in Fig. 4 that the dropout rate can be very different in different courses. The phenomenon of withdrawal from courses that requires a certain academic foundation is more obvious. It may be due to the mismatch of abilities and course difficulty or lack of interest. At the same time, due to the different genders in the same type of courses, there is a certain discrepancy in dropout ratios between the male and female. It can be seen in the figure that female users prefer humanities and humanities, while male users prefer social science.

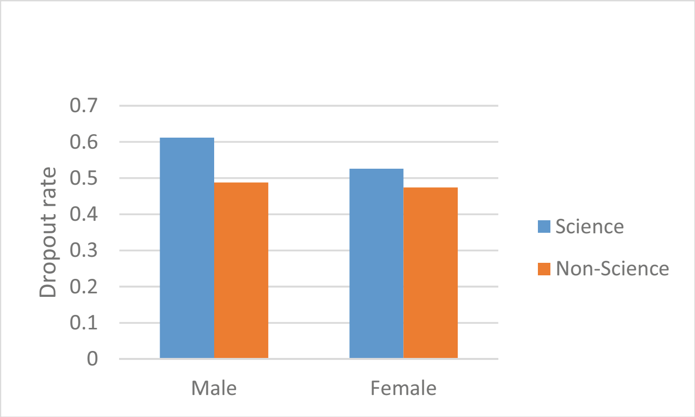


Fig. 4. Course category and user sex

4 Methodology

4.1 Formulation

Definition 1 (Behavioral Sequence). A sequence $X_u = (x_1, x_2, \ldots, x_t, \ldots, x_n)$ is defined as a series of activities that a user u has taken from the first day to the last day.

Definition 2 (Behavior). For each user u we define a m-dimensional vector of an activities sequence $x_t = (x_{t1}, x_{t2}, \ldots, x_{ti}, \ldots, x_{tm})$ which represents the user behavior series of the tth day, with $x_{ti} \in [0, 1]$. If is 0, which means the corresponding activity is not taken by the user in the tth day. On the contrary, is taken by the user.

Definition 3 (Other attributes). Continuously process discrete features such as gender, course information such as course categories, and other information in the dataset except user behavior $Z = (z_1, z_2, \ldots, z_l)$.

Our goal is to predict whether the user will drop out in the next period based on the existing behavior. If there is effective behavior, it will be recorded as not dropped out, which is represented by 0, otherwise, 1 represents dropped out.

4.2 Deep Model

Figure 5 shows the model proposed in this paper. The framework mainly includes the following parts: input module, encoding module, period detection and attention mechanism selection module, and prediction module.

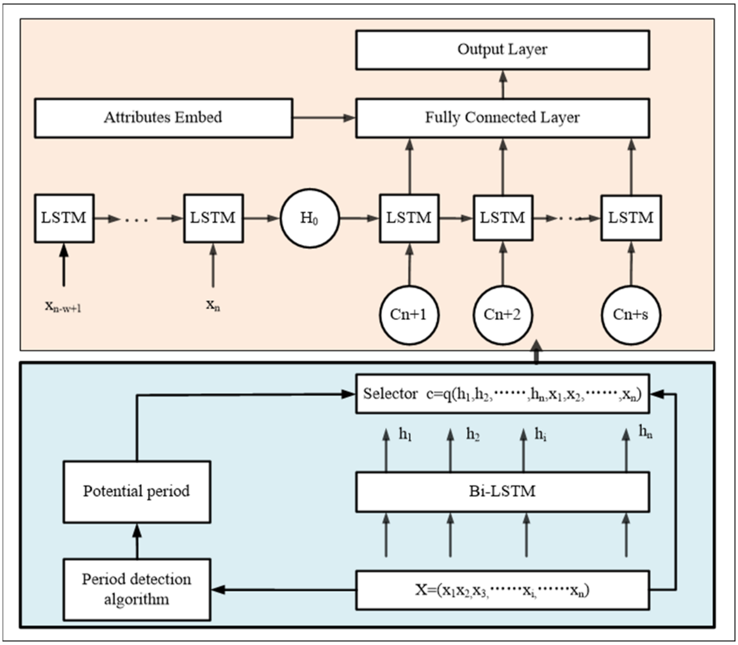


Fig. 5. Model structure

Input Module: The input module preprocesses the given user behavior data, and then selects *m* behavior categories that have a large impact on the dropout rate based on the hypothesis test method. Finally, the user behavior is converted into the one-hot vector as the feature vector, combining the feature vectors of each day we get the matrix $X_u = (x_1, x_2, \ldots, x_t, \ldots, x_n)$.

Encoding Module: As shown in Fig. 6, we encode each vector in the matrix in turn. The behavior is coded by using the Bi-LSTM method. There are two purposes of encoding: (1)

the Bi-LSTM method can retain the behavior characteristics before and after the cycle, and reduce the errors caused by the learner's behavior fluctuations. When introducing the influence of historical behavior, the relevant factors are selected through the cycle. Since the detected learner behavior cycle is within a certain confidence interval which means that the behavior before and after the cycle may have a certain deviation. (2) The context information captured by encoding provides a weight reference for the attention mechanism selector. When the attention mechanism selector selects historical behaviors, the weight corresponding to each behavior is obtained by calculating the similarity between the current hidden state and the result obtained by encoding. In this way, it is helpful for the attention mechanism to select more relevant behavior vectors in later prediction.

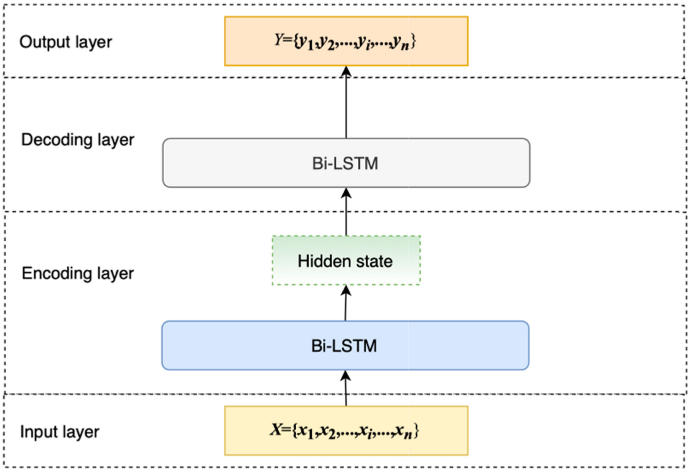


Fig. 6. Structure of encoding module

The input raw data $X = (x_1, x_2, \ldots, x_t, \ldots, x_n)$ is used as the input of the Bi-LSTM layer, and the hidden state $H' = (h_1, h_2, \ldots, h_t, \ldots, h_n)$, then passed into the decoding layer, the decoding layer is also a two-way LSTM, the hidden state is restored to the same or similar result of the original input, and the loss function is the mean square error:

$$\min \sum_{i=1}^{i=n} \left(||\boldsymbol{x_i} - \boldsymbol{y_i}|| \right)^2 \tag{1}$$

Bi-LSTM is composed of forward LSTM and backward LSTM, and the basic model LSTM is an improvement on the traditional RNN, it is a special RNN network in order to solve the problem of long dependence. Each unit of LSTM contains a unit state and three controlled gates to update the unit state. The specific calculation formulas are shown below:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{2}$$

$$f_t = \sigma\big(W_f h_{t-1} + U_f x_t + b_f\big) \tag{3}$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{4}$$

$$\widetilde{C}_t = \tanh(W_a h_{t-1} + U_a x_t + b_a) \tag{5}$$

Afterwards, the cell output state can be calculated by:

$$C_t = C_{t-1} \odot f_t + i_t \odot \widetilde{C}_t \tag{6}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{7}$$

The input is $\mathbf{x_t}$ at the time t, the cell input state is $\widetilde{C}_t$, the cell output state is C_t and its former state is C_{t-1}, the hidden layer output is h_t and its former output is h_{t-1}, a LSTM cell has three gates, which are input gate, forget gate and output gate and the corresponding states are i_t, f_t and o_t. W, U, b are weight matrices corresponding to hidden layer, input layer and bias vectors, they all can gotten by training. In addition, σ is a activation function and tanh represents the hyperbolic tangent function.

Time Series Period Detection and Attention Mechanism Selector. According to the analysis of user behavior in the article, when new content is released in a course, the activity is significantly increased in the course learning, and the user activity shows periodic changes based on the course release, so the work in this section is to find user behavior cycles in a series of sequential events and select candidate elements for attention.

We use cross entropy for period detection. Cross entropy is used to measure the difference between two probability distributions. We use $d_1, d_2, \ldots, d_n$ to indicate whether a user has a valid record of visiting the course every day. If so, d is recorded as 1 and vice versa as 0. Therefore, for each user, a binary sequence string $S = [d_1, d_2, d_3, \ldots, d_n]$ of length n is obtained, and the purpose is to analyze the sequence S to find its potential period a. Period detection is to find a suitable division from a series of 0,1, so that the elements in S are divided into k segments according to the equal length, so $S' = \{P_1, P_2, \ldots, P_k\}, P_i = \left[d_{a\cdot(i-1)+1}, d_{a\cdot(i-1)+2}, \ldots, d_{(a\cdot i)}\right]$. We need to find a suitable value of a such that the number of occurrences of 1 in each interval after division is the same, and the relative position of 1 in each division interval is the same. Assume that the uniform distribution is $R = \left\{\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}\right\}$, and the distribution obtained according to a certain period is P. Calculate the KL (Kullback-Leibler Divergence) between two distributions by the following cross entropy. Among them, $P(i)$ refers to the ratio of the number of occurrences of '1' to the total number of times in P_i.

$$D(P\|R) = \sum_{i \in S_I} P(i) log \frac{P(i)}{R(i)} \tag{8}$$

We calculate the similarity between actual period division and uniform distribution based on cross entropy. Through the greedy algorithm we traverse from 2 to $\left\lceil \frac{|S|}{2} \right\rceil$ in turn, based on the KL divergence distance we find the K elements with the smallest distance to form the candidate period set KD $= \{a_1, a_2, .., a_k\}$, after satisfying the distribution periodicity, the structural periodicity still needs to be satisfied, that is, in each sub-division obtained according to the periodic division of distribution, the relative position of 1 should be consistent, and we use the intra-class distance to measure, and each sub-sequence after division is regarded as a particle $P_1, P_2, .., P_k$, calculate the sum of the distances between the particles, the smaller the distance between the classes, the smaller the confidence level meets the structural periodicity. The formula for calculating the distance within a class is as follows:

$$l^2 = \frac{1}{\left\lceil \frac{|S|}{a} \right\rceil^2} \sum_{i=1}^{\left\lceil \frac{|S|}{a} \right\rceil} \sum_{j=1}^{\left\lceil \frac{|S|}{a} \right\rceil} d^2(P_i, P_j) \tag{9}$$

$$d^2(P_i, P_j) = \sum_{k=1}^{a} (d_{a \cdot (i-1)+k} - d_{a \cdot (j-1)+k})^2 \tag{10}$$

Finally, the candidate period with the smallest distance within the class is selected as the final period. The specific period detection method is shown in Algorithm 1.

Algorithm 1. The algorithm of period detection

```
Input:
    Behavior sequence: S
Output:
    period a
$KD \leftarrow \emptyset$
for $a$ = 2 to $\left\lceil \frac{|S|}{2} \right\rceil$ do
    Segment the S into $\left\lceil \frac{|S|}{a} \right\rceil$ subsequences, $\left\lceil \frac{|S|}{a} \right\rceil = k\ and\ S' = \{P_1, P_2, .., P_k\}$
    $P_i = [d_{a \cdot (i-1)+1}, d_{a \cdot (i-1)+2}, \dots, d_{(a \cdot i)}]$
    $D(P||R) = \sum_{i \in S'} P(i) log \frac{P(i)}{R(i)}$
    if |KD|<k:
        $KD \leftarrow KD \cup a$
    else if $D(P||Q)$<min KD:
        $KD \leftarrow KD \cup a$
end for
$A \leftarrow \infty$
for $a$ in $KD$:
    $S' = \left\{P_1, P_2, .., P_{\left\lceil \frac{|S|}{a} \right\rceil}\right\}$ and $P_i = [d_{a \cdot (i-1)+1}, d_{a \cdot (i-1)+2}, \dots, d_{(a \cdot i)}]$
    $l^2 = \frac{1}{\left\lceil \frac{|S|}{a} \right\rceil^2} \sum_{i=1}^{\left\lceil \frac{|S|}{a} \right\rceil} \sum_{j=1}^{\left\lceil \frac{|S|}{a} \right\rceil} d^2(P_i, P_j)$, $d^2(P_i, P_j)$ is the square of the Euclidean distance between $P_i$ and $P_j$
    if $l^2 < (\min A)^2$:
        $a$ replace ($\min A$)
end for
return $a$
```

We obtain the potential period a of the user through the Algorithm 1. In encoding phase we get $H = \{h_1, h_2, \ldots, h_i, \ldots, h_n\}$, h_n represents the intermediate state corresponding to x_n, the prediction time is from t_{n+1} to t_{n+s} and assume the currently predicted moment is t_x, $k = t_x \, mod \, a$ and we get the set $TR_{in} = \{k + i * a\}$ of historical time periods aligned at time t_x, $i \in [0, \lfloor \frac{\mathrm{s-k+1}}{a} \rfloor]$. The hidden layer output corresponding to each time in TR_{in} constitutes a set $H_{select} = \{h_k, h_{k+1*\mathrm{a}}, \ldots, h_{k+\lfloor \frac{\mathrm{s-k+1}}{a} \rfloor * a}\}$. The purpose of this selector is achieved.

In order to introduce the influence of historical behavior, we put the original behavior data with a certain weight as part of the input at the predicted moment. At the same time, avoiding the behavior deviation of the learner before and after the behavior cycle, the input also contains the encoded value corresponding to the original behavior data.

Hidden Layer State Initialization: Since the period detection takes the effects of historical behavior into account, and it is necessary to introduce the effects of sequence event. The influence of the time series requires a suitable time window size w. The selection of the initial time period of the cyclic neural network chain of the prediction module is w days before the start time of the prediction, and the initial hidden layer state is obtained from t_{n-w+1} to t_n to get the initialized hidden layer state. we set the window size w to detected period a. If the selected behavior matrix in the current time period is sparse, it is replaced by the mean value of the behavior matrix of other users in the corresponding time period. Therefore, the input of the prediction module and the state of the hidden layer initialized introduce the influence of historical period behavior and the influence of sequence events respectively.

Prediction. According to different prediction time, the selector selectively collects information from the encoding module and performs prediction. In order to introduce the influence of historical behavior, we put the original behavior data with a certain weight as part of the input at the predicted moment. At the same time, avoiding the behavior deviation of the learner before and after the behavior cycle, the input also contains the encoded value corresponding to the original behavior data. The specific calculation formula is as follows:

$$c_t = \sum w_i(\beta h_i + \gamma x_i) \tag{11}$$

$$w_i = softmax(f(h_i, h_{curr})) \tag{12}$$

where w_i is the weight for h_i, h_i is the output of the coding layer, $h_i \in H_{select}$ and h_{curr} denotes current status from the recurrent layer, f is a function which can calculate the similarity between h_i and h_{curr}. c_t takes the information collected by the input layer and encoding layer as the input of the prediction module.

For datasets with relevant user information and course information data, the prediction is completed by embedding user information and course information in binary representation through a fully connected layer, and increases the original vector by some dimensions.

5 Experiment

For the experiments in this article, we used KDDCUP's 2015 competition data, which included user behavior characteristics such as watching videos, submitting assignments, forum discussions, accessing course Wiki, browsing other course objects other than video assignments, closing web pages, etc. Among them, we predict from the known 30-day behavior logs whether users will have valid behavior records for the next 10 days. The XuetangX dataset contains specific course information, including course categories, course start and end dates, user personal information, gender, age, education level, etc. and user behavior logs including the behavior initiator, occurrence time, related objects, etc. Choose a 42-day behavioral record with a forecast period of 7 days. Ten-fold cross-validation is used during the training of the algorithm.

5.1 Performance Metrics

In order to evaluate the performance of our proposed model, it is measured by four indicators, namely Precision, Recall, and F1-score, and Area Under Receiver Operating Characteristic Curve (AUC) score. We show two representative indicators, F1score and AUC value.

Precision P:

$$P = \frac{TP}{TP + FP} \tag{13}$$

Recall R:

$$R = \frac{TP}{TP + FN} \tag{14}$$

F1-score:

$$F1 = \frac{2 * P * R}{P + R} \tag{15}$$

TP: The positive class that is correctly predicted

FP: The negative class that is predicted as positive

FN: The positive class that is predicted as negative

AUC: It is the area corresponding to the ROC curve. The larger the area, the stronger the generalization ability of the model.

5.2 Performance of Methods

We compare the proposed new model with several existing classification methods:

1) SVM: The support vector machine is a binary classification algorithm for supervised learning
2) LR: Logistic regression model is a classification algorithm that can handle binary classification and multivariate classification

3) RF: Random Forest model is ensemble learning algorithms based on decision tree
4) AdaBoost: AdaBoost is an iterative algorithm, an important ensemble learning technology.
5) LSTM: Long Short-Term Memory is a special RNN network, designed to solve the long dependency problem.

Table 2. The performance of the whole methods on KDDCUP

Methods	F1-score(%)	AUC(%)
SVM	91.07	87.81
LR	91.42	88.12
RF	92.10	88.63
AdaBoost	92.17	88.68
LSTM	92.19	88.72
Our Method	**92.78**	**89.84**

Table 3. The performance of the whole methods on XuetangX

Methods	F1-score(%)	AUC(%)
SVM	87.73	81.34
LR	87.52	81.19
RF	88.11	82.65
AdaBoost	88.77	84.06
LSTM	88.89	84.12
Our Method	**89.68**	**84.94**

Table 2 and Table 3 show the experimental results of our model and baseline methods on the KDDCUP and the XuetangX. From this, we can clearly see that all models perform better on KDDCUP than XuetangX. The former has better data quality in data processing and less noise. The same method can differ by three to five percentage points on two different datasets. At the same time, the performance of our proposed model is better than several baseline methods in F1-score and AUC values, which proves the effectiveness of our model. In baseline methods, the integrated learning algorithm, as an enhancement algorithm, is better than a single base learner. AdaBoost has achieved good results on both datasets. Compared with traditional machine learning classification algorithms, the deep learning algorithm LSTM has some advantages but it is not very obvious, may be our data is not very complicated and the time span is long, or the simple LSTM cannot fully learn the regularity of user behavior changes. The learning effect of LSTM is not very ideal. However, using it as the basic unit of our proposed model and redesigning the entire framework, the overall effect is obviously better than other methods. It can be seen that in different scenarios, although the model cannot be universally used, it may be improved according to the actual situation. In this paper, we focus on the characteristics of user learning, not only considering that user behavior is a sequence event problem, but also that user behavior will have a learning period based on course release or their own learning plan, Therefore, we have added the corresponding influencing factors of historical behavior, and various considerations make the method more effective.

6 Conclusion

In this paper, we studied the problem of predicting the dropout rate in MOOCs. Firstly, we do a deep analysis of user learning behavior to find which are the important factors the affect the dropout rate. And then we propose a novel deep model based on recurrent network. In the novel model, we combine the effects of sequential behavior over the current period with the effects of past historical behavior to predict the dropout and we also embed the attributes of user and course. Finally, we demonstrate the effectiveness of our methods by taking the experiments on two datasets, our proposed method performs better than the state-of-the art methods. In future work, we will further study the choice of sequence length in the influence of sequence behavior in the current period.

Acknowledgements. This work is supported by the key projects of the national natural science foundation of China (No. U1811263), the major technical innovation project of Hubei Province (No. 2019AAA072), the National Natural Science Foundation of China (No. 61572378), the Teaching Research Project of Wuhan University (No. 2018JG052), the Natural Science Foundation of Hubei Province (No. 2017CFB420). We also thank anonymous reviewers for their helpful reports.

References

1. Ipay, B., Ipay, C.B.: Opportunities and challenges for open educational resources and massive open online courses: the case of Nigeria. Commonwealth of Learning. Educo-Health Project. IIorin (2013)
2. Mackness, J., Mak, S.F.J., Williams, R.: The ideals and reality of participating in a MOOC. In: Networked Learning Conference (2010)
3. Feng, W.Z., Tang, J., Liu, T.X.: Understanding dropouts in MOOCs. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 517–524 (2019)
4. Kate, J.: Initial trends in enrolment and completion of massive open online courses. Int. Rev. Res. Open Distance Learn. **15**(1), 133–160 (2014)
5. He, J., Bailey, J., Rubinstein, B.I.P., Zhang, R.: Identifying at-risk students in massive open online courses. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 1749–1755 (2015)
6. Dalipi, F., Imran, A.S., Kastrati Z.: MOOC dropout prediction using machine learning techniques: review and research challenges. In: Global Engineering Education Conference (EDUCON), 2018 IEEE, pp. 1007–1014 (2018)

7. Shi, Y.L., Peng, Z.Y., Wang, H.N.: Modeling student learning styles in MOOCs. In: Proceedings of the 26th International Conference on Information and Knowledge Management (CIKM), pp. 979–988 (2017)
8. Natek, S., Zwilling, M.: Student data mining solution–knowledge management system related to higher education institutions. Expert Syst. Appl. **41**(14), 6400–6407 (2014)
9. Coleman, C.A., Seaton, D.T., Chuang, I.: Probabilistic use cases: discovering behavioral pattern for predicting certification. In: Proceedings of the Second ACM Conference on Learning @ Scale, pp. 141–148 (2015)
10. Graves, A.: Supervised Sequence Labelling with Recurrent Neural Networks. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2
11. Ito, T., Tsubouchi, K., Sakaji, H., Yamashita, T., Izumi, K.: Contextual sentiment neural network for document sentiment analysis. Data Sci. Eng. **5**(2), 180–192 (2020). https://doi.org/10.1007/s41019-020-00122-4
12. Anderson, A., Huttenlocher, D., Kleinberg, J.: Engaging with massive online courses. In: Proceedings of the 23rd International World Wide Web Conference, pp. 687–698 (2014)
13. Kloft, M., Stiehler, F., Zheng, Z., Pinkwart, N.: Predicting MOOC dropout over weeks using machine learning methods. In: Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs, pp. 60–65. Association for Computational Linguistics, Doha (2014)
14. Colin, T., Kalyan V., Una-May, O'Reilly.: Likely to stop? Predicting stopout in massive open online courses. Computer Science (2014)
15. Ramesh, A., Goldwasser, D., Huang B.: Uncovering hidden engagement patterns for predicting learner performance in MOOCs. In: Proceedings of the Second ACM Conference on Learning @ Scale, pp. 157–158 (2014)
16. Balakrishnan, D., Coetzee, D.: Predicting students retention in massive open online courses using hidden Markov models. Technical report, UC Berkeley (2013)
17. Chanchary, F.H., Haque, I., Khalid, M.S.: Web usage mining to evaluate the transfer of learning in a web-based learning environment. In: Proceedings of the First International Workshop on Knowledge Discovery and Data Mining, pp: 249–253. IEEE Computer Society (2008)
18. Stein, J., Xing, W., et al.: Temporal predication of dropouts in MOOCs: reaching the low hanging fruit through stacking generalization. Comput. Hum. Behav. (2016)
19. Fei, M., Yeung, D.Y.: Temporal models for predicting students dropout in massive open online courses. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW), pp. 366–372 (2011)
20. Wang, W., Yu, H., Miao, C.: Deep model for dropout prediction in MOOCs. In: Proceedings of the 2nd International Conference on Crowd Science and Engineering, pp. 26–32 (2017)
21. Crossley, S.A., McNamara, D.S.: Developing component scores from natural language processing tools to assess human ratings of essay quality. Rev. Manag. Sci. **9**(4), 1–26 (2014)
22. Qiu, J., Tang, J., Liu, T.X.: Modeling and predicting learning behavior in MOOCs. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 93–102 (2016)