High fail and dropout rates are the major problems in distance education. Due to a large number of online learners and limited teacher resources, it is essential to accurately identify these potential at-risk students in advance and provide timely aids, which will help to improve the educational outcome. In the online learning environment, students’ online learning behaviors can be recorded easily, with the click data being the most common one. Students’ learning behavior can reflect their learning situation and may differ among different students and periods. This paper proposed a model that uses the short-period activity characteristic and long-term changing pattern to predict the potential at-risk students. The model contains two stages: information extraction and information utilization. The first stage extracts data from the log files and organizes it in a form suitable for the model. In the second stage, according to the different characteristics of students’ short-term and long-term learning behavior, a convolution residual recurrent neural network (CRRNN) model is proposed. The convolutional neural network is used to obtain the representation of the student’s learning behavior in a certain period. Then, the residual recurrent neural network is used to get the behavior changing pattern over the periods. The experimental results indicate that the proposed model has higher performance than the three widely used baseline methods on the OULA dataset and has good practical application value for teaching and management.

1. Introduction

With the emergence of computer and network technology, online learning is widespread in higher education and has become a new learning way. Especially in 2020, under the influence of COVID-19, most of the teaching and learning activities are carried out online. Online learning can overcome time and space limitations, reduce learning costs, and open learning opportunities for individuals who cannot join a face-to-face university class in many developing regions [1, 2]. Nevertheless, high dropout rates and failure rates are prominent issues in such a learning environment. Some literature reports that the average percentage of students who enrolled and succeeded in online courses is less than 15%, or even as low as 5% [3]. The research conducted by Bailey et al. [4] indicates that among 51,306 students enrolled in the online courses, only 795 students completed their studies, with a completion rate of just 1.5%. The extensive number of students who fail or drop out of the course reduces the effects of online learning and impacts the reputation of institutes. Therefore, there is a pressing need to develop methods to predict the at-risk students [5] for providing them timely help and intervention.

Unlike the face-to-face physical teaching environment, where teachers directly contact the students and know much more about their states, there is no direct contact between instructors and students in the online environment. Fortunately, with the progress of technology, plenty of data generated by students in the online learning environment is recorded and accumulated [6], which guarantees researchers
to analyze the potential reasons for failure. Various techniques have been introduced into this domain and form a new research direction, educational data mining (EDM). It is common to monitor users’ network access behavior by saving users’ clicking on different links and staying time on the pages in log files. In the online learning environment, the pages provide various teaching content. By analyzing students’ access to different resources, we can understand students’ learning situations. However, simply representing students’ learning behavior based on the cumulative visits to various resources is not accurate because it loses the information about the changing process of students’ learning behavior. Generally speaking, students’ behavior may remain the same in a short period, but as time goes on, this behavior will change gradually. This paper proposes an improved deep learning neural network based on the above two features to predict the at-risk student.

The main contributions of this paper are as follows.

1. The fail and withdrawn students are predicted with clickstream data that is easy to obtain in online environments. Therefore, it is more convenient to deploy in practical application systems.

2. According to the studied students’ learning behavior from the two aspects of short-term behavior characteristics and long-term changing trends, convolutional neural networks and improved recurrent neural networks are introduced into the model. Using convolution neural networks to learn the activity characteristics in a short period and recurrent neural networks to get behavior patterns in the long term can promote the model’s accuracy and precision.

3. Experimental results show that the proposed model has a better effect than the baseline methods.

The remainder of this paper is organized as follows. Literature related to EDM is reviewed in Section 2. Section 3 presents the details of the proposed model. Experiments and results are given in Section 4. Section 5 provides the conclusions and possible future work of this study.

2. Related Works

EDM is a comprehensive application of mathematical statistics, machine learning, and data mining techniques and methods to process and analyze big educational data. Predicting students’ academic performance and identifying the at-risk students in advance constitute an important research topic in EDM [7–9]. Recently, machine learning methods have been the most commonly used technique in this area. For example, Marbouti et al. [10] build an academic performance prediction model based on students’ previous course performance. In their research, several machine learning approaches are employed and compared, including logistic regression (LR), support vector machine (SVM), decision tree (DTC), multilayer perceptron (MLP), and naïve Bayes classifier (NBC). They found that the NBC and the ensemble model consisting of NBC, SVM, and KNN can obtain better accuracy. Their study also shows that students who perform poorly in the first few weeks are more likely to fail. Similarly, Hlosta et al. [3] studied the relation between the initial examination and the course final result. They found that nearly 90% of the students who failed the first test would fail the course, while among the students who did not submit the initial evaluation, 94.96% would not pass the course. Besides, random forest (RF) is also commonly used in online learning prediction due to its robustness to outlier data and the ability to inspect the model for insights into the most potent discriminating variables [11, 12].

In addition to students’ prior performance on the course, their online learning behaviors are also indicators of the final academic result. Hu et al. [1] study the online activities, including total time material viewed, assignment delay, and forum participation rate. They found that the time-dependent variables significantly impact the online learning performance. In several programming-related courses, Guerrero-Higueras et al. [13] use the submission records in the version control system to monitor students’ work progress to evaluate their academic performance and predict whether they will pass the course. Likewise, Wang et al. [14] feed students’ program submission sequence into a recurrent neural network (RNN) model to obtain representations of the student’s knowledge and then predict their future performance. Okubo et al. [15] proposed the recurrent neural network (RNN) based model to predict student’s final grades from the log data, including attendance, course views, and slide views in BookLooper. Waheed et al. [16] employ artificial neural network (ANN) to predict students’ academic performance based on clickstream data.

In addition to academic performance prediction, building models to predict students who are likely to drop out of online courses equally has a significant practical implication, which can help instructors design interventions to encourage them to complete the course before falling too far behind. At present, many efforts are devoted to the early identification of students who may drop out of the study [7, 17, 18]. However, the determinants causing students to drift toward dropping out of the course span across multiple domains and often interact with each other [19]. Literature indicates that factors associated with withdrawal might differ [20], and students drop out more frequently in the course early period [3]. Kuzilek et al. [7] employed the Markov chain model to analyze the behavior of the online student and its influence on the dropout. Their experiment results show that the probability of students dropping out of the study is twice as higher for students with no activity in VLE than for students with at least some activity in VLE, which demonstrates that the online activity is a significant indicator for the dropout prediction. Haiyang et al. [17] proposed the time series forest classification (TSF) model to predict the dropouts based on the interaction sequence between students and resources provided by online learning platforms. The accuracy of their model ranges from 80% to 90% in the selected courses.

Aiming at solving the problem of at-risk students in the online environment, researchers have tried to use different methods and data for analysis and prediction. For instance,
Adnan et al. [12] studied the use of demographics, clickstream, assessment scores, and a combination of these data to build prediction models and found that RF gives better performance than many other models such as SVM and KNN. Among various types of data, clickstream data is one of the most abundant and easily collected learning behavior data types in the online learning environment. It records the detailed interaction between students and the platform, reflecting students’ participation and behavior preferences. These behavioral data may indicate how hard students studied, their mental state, their knowledge level, etc. For example, active students may prefer to ask questions or solve others’ problems in the discussion forums.

In contrast, students with a poor foundation may like to view the teaching content, watch the related tutorial videos repeatedly in the course’s initial phase, and gradually participate in the discussion when they have some basic knowledge. Therefore, students’ interaction with the system can indicate their participation patterns and efforts devoted to study. Prior literature suggested that online behavior can be used for failure and withdrawn prediction. However, the relationship between students’ engagement in the virtual learning environment and their performance is complicated. Boulton et al. [21] found that students with high participation in the online learning environment tend to have higher achievement, while low engagement does not mean poor performance. Thus, the potential behavior patterns need to be further studied. Clickstream is the information that can reflect students’ activity in the online learning environments and has been widely used to predict at-risk students [17, 22].

Even though students’ behaviors are widely utilized to predict academic performance or dropouts, existing literature has three limitations. (1) It predicts either academic performance or withdrawals. These two problems can be unified actually since they both share some features. Although many reasons would lead students to drop out of the course, one of the main reasons may be that they encounter academic difficulties and fear of failure and then lose the confidence to keep on. (2) Most literature uses the final cumulative data, such as the total number of posted messages and the total time spent viewing material to build the prediction model. It is insufficient for discriminating students’ learning patterns, thus degrading the model performance. For instance, two students spent equal time viewing teaching videos during a specific time interval, and one’s viewing behaviors spread on different workdays while the viewing behaviors of the other mainly concentrated in the weekend. They are probably two different kinds of students. The latter may have a job and just have time to learn in the weekend, which may indicate more desire to advance. (3) It needs to select model input features manually, and improper input variables may impact the model performance, which requires domain professional knowledge.

Nevertheless, extracting appropriate features from the raw clickstream data to improve model performance is challenging [23]. Deep learning approaches, such as convolutional neural network (CNN) [25] and recurrent neural network (RNN) [26, 27], have been widely used since their application in 2006 [24]. CNN is one of the most representative deep learning frameworks, which has good performance in feature extraction from the raw input data and has been successfully used in many fields, for instance, image processing and handwriting recognition [28–30]. RNN is another type of neural network mainly used to process the sequence data, such as natural language processing and speech recognition. The output of RNN at the current time depends on the current input information and the input information of all previous times. Therefore, it can find the changing pattern and trend of things over time. Inspired by these, we proposed a novel prediction model in this study, which utilizes CNN to extract the representation of a student’s learning behavior in a period, then used RNN to obtain the evolving pattern across the periods.

3. Proposed Method

The problem of predicting whether a student will fail or drop out of the course can be treated as an ordinary binary classification problem. This paper expects to use the known learning activities of the student during the study to make the prediction. Many resources will be provided on the online platform for students to use in the learning process. Students’ use of these resources will produce click data, which is collected and recorded in the log files. Figure 1 shows some examples of click activities on the learning resources. Intuitively, the time spent online is a significant indicator of the engagement and effort students devote to the course. However, there is no unified model for the correlation between the online time and the final course achievement [31]. As mentioned previously, the total time spent on each resource does not well represent the whole learning process of the students, and there are different engagement patterns throughout the course duration in the online settings. For instance, some students engage in the social aspects of the online community by posting in forums, asking and answering questions, while others only watch lectures and take quizzes without interacting with community members. Some study on the weekdays, while others may study mainly on the weekends [32]. Therefore, their activity’s temporal (days) and spatial (resources) distribution may differ.

Generally, students with higher engagement usually have better academic performance and are less likely to drop out of school [21]. Students who do well in learning are much more disciplined and regular in their behaviors [33]. In addition, Haiyang et al. [17] pointed out that the contribution of different behavior types differs in the prediction. For instance, interaction data with video and text material are more relevant to dropout prediction than the interaction data with forums for most courses. Based on this, we propose a model to predict whether students will fail or drop out of the study by using the temporal and spatial distribution characteristics of the clickstream, according to the students’ usage of various resources at different periods and the changing process of these behaviors over time.
The prediction procedure in this paper contains two stages: In the first stage, the data was drawn from the record files and transformed to the form suitable for the prediction model. For each student’s learning behaviors, the clicks on various resources are counted by day and stacked in order, to form a two-dimensional matrix. The row is the sequence of days, and the column is the resource categories. The second stage uses the convolution residual recurrent neural network (CRRNN) for prediction. The convolution neural network is used to learn the activities characteristics in each period from the input matrix data. In addition, the behavioral changing patterns over periods are retained with the recurrent neural network. Then, the final behavior representation is input to a fully connected neural network to predict whether the student will fail or drop out of the course. Figure 2 shows the model framework. Next, we will introduce each part of the second stage in detail.

3.1. Part I. Part I includes a batch normalization layer and two convolutional layers. It takes the daily click data matrix as input, employs the convolutional neural network to attain the latent learning behavior characteristic in each period, and is represented with activity vectors \( \{x_i | x_i \in \mathbb{R}^{20}\} \). The number of clicks on different resources varies greatly. Hence, we introduced the data normalization layer to normalize the data, decreasing the influence of prominent value interval attributes on small value interval attributes and reducing the computational complexity in the calculation process. We use batch normalization (BN) [31] to normalize the data in our model. The BN procedure is defined as Algorithm 1.

Students’ behavior will change over time in the learning process, but it may have a specific character within a short period. Two convolutional layers are introduced for better obtaining the representation of behavior characters in different periods. The convolution layer can be expressed as (1) and (2).

\[
S_l^c(u) = \phi \left[ \sum_{c=1}^{CH} \sum_{f=1,v=p} W_{l,c}^f(f) \times S_{c-1}^l(v) \right] + b_k^l, \quad (1)
\]

where \( S_l^c(u) \) is the item value of the \( f \)th layer at position \( u \) of the \( k \)th kernel. \( CH \) is the total number of channels of the \( (l-1) \)th layer. \( W_{l,c}^f(f) \) is the weight of the \( k \)th kernel in layer \( l \) at position \( f \) of channel \( c \). \( S_{c-1}^l(v) \) is the value of the \( c \) channel of the \( (l-1) \)th layer at position \( v \), \( p \) is the initial item location of channel \( c \) in the \( (l-1) \)th layer corresponding to the \( f \)th element of \( W_{l,c}^f(f) \), \( t \) is the number of kernel’s total elements, and \( b_k^l \) is the bias term of the \( k \)th kernel of the \( l \)th layer. \( \phi(\cdot) \) is the nonlinear activation function, and in this paper, it is the rectified linear unit (ReLU) function [34], which can be represented as

\[
\phi(a) = \max(0, a). \quad (2)
\]

Different convolutional kernels have various receptive fields and extract the features from different angles. Our model expects to gain student activity characters on weekdays and non-weekdays, so we set each period to seven days, five weekdays and a two-day weekend. Moreover, we assign the first convolutional layer kernel size to \((5, 1)\) and the stride to \((1, 1)\). The second layer kernel size is \((3, 1)\), and the stride is \((7, 1)\). In this way, the receptive field of the two-layer network is 7, which may get the weekly activity representation vector. The dimension of the obtained representation vector is 20, equal to the types of resources. To preserve information, our model did not introduce the pooling layer.

3.2. Part II. Part II includes a batch normalization layer and an improved multilayer recurrent neural network (RNN) by shortcuts between layers. We use the batch normalization layer to normalize the vectors sequence before inputting them to the RNN. RNN is a kind of neural network that consists of the hidden state \( h \), and the output at time \( t \) is
related to both the current input $x_t$ and its previous hidden state $h_{t-1}$, which can be expressed as follows:

$$ h_t = f(h_{t-1}, x_t), $$

where $h_t$ is the output of the current input $x_t$ as well as the hidden state for the next input $x_{t+1}$. RNN has a good effect in processing the sequence data; however, due to the problems of gradient vanishing and gradient exploding, the training becomes difficult, and the application is limited. To solve this issue, many improved versions have been proposed, such as long short-term memory (LSTM) [26] and gated recurrent unit (GRU) [27]. Both can capture the much longer dependency features of the sequence data and have been widely used in many fields. Compared with LSTM, GRU has fewer parameters and easier convergence while achieving the same effect as LSTM. Therefore, GRU is adopted in this paper. The hidden state $h_t$ of GRU at time $t$ can be computed by (4)–(9), where $W^r$, $W^z$, and $W$ are matrices, which are learned through backpropagation. $\sigma(\cdot)$ and $\tanh(\cdot)$ are nonlinear activation functions that can be defined as (8) and (9), respectively. $[x, y]$ is a concatenation operation that concatenates the $x$ and $y$ vector into a new vector. Research [35] demonstrated that increasing GRU layers could capture better structure of the sequence and improve model performance. The learning behaviors of students in different periods can be treated as an activity sequence. To better capture the activity sequence patterns, this paper adopts three layers. The length of the hidden state is five. Experiments show that this can avoid overfitting caused by too many parameters and insufficient samples and yield promising results. The input of the first layer

**Algorithm 1:** Batch normalization.

```
Input: Values of a mini-batch $\mathcal{B} = \{x_1, \ldots, x_m\}$; Parameters to be learned: $\gamma, \beta$
Output: $\text{BN}_{\gamma, \beta}(x_i)$
(1) $\mathcal{B} \leftarrow \{x_1, \ldots, x_m\}$
(2) $\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$
(3) $\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2$
(4) $\tilde{x_i} \leftarrow (x_i - \mu_B) / (\sigma_B^2 + \epsilon)$
(5) Return: $\text{BN}_{\gamma, \beta}(x_i) = \gamma \tilde{x_i} + \beta$
```

**Figure 2:** Model framework.
is the output vector sequences of CNN in part I, and the length of each vector is 20, which is equal to the number of the resource items. Then, we take the output of the first layer as the input of the second layer. For the third layer, the results of the previous two layers are added as its input, which can enhance the model representation ability [36].

$$r_i = \sigma([h_{t-1}, x_i]W^r),$$

$$z_i = \sigma([h_{t-1}, x_i]W^z),$$

$$\tilde{h}_t = \tanh([r_t * h_{t-1}, x_i]W),$$

$$h_t = z_t * \tilde{h}_t + (1 - z_t) * h_{t-1},$$

$$\sigma(z) = \frac{1}{1 + e^{-z}},$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$  

3.3. Part III. Part III concatenates the three layers’ last hidden state of the input sequence as the embedding of the activity sequence, which can be expressed as follows:

$$\text{out}_\text{rmn} = \text{cat}[h^1_M, h^2_M, h^3_M],$$

where the subscript sl indicates the length of the input sequence and the superscript is the layer number.

3.4. Part IV. Part IV is a two-layer fully connected network. It takes the activity sequence embedding as the input and outputs the final classification prediction. The $i$th neuron in the first layer can be expressed as follows:

$$a^1_i = \varphi(b^1_i + \sum_{j=1}^{N} \omega_{ji}R_j),$$

where $a^1_i$ is the $i$th neuron in the first layer, $b^1_i$ the $i$th bias in the first layer, $\omega_{ji}$ is the weight from the $j$th item in Part III concatenated vector to the $i$th neuron in the first layer, $R_j$ is the $j$th item of the concatenated vector, $N$ is the number of items in the vector, and $\varphi(\cdot)$ is a ReLU activation function defined as (2). The second layer can be expressed as follows:

$$y^2_i = b^2_i + \sum_{j=1}^{M} \omega_{ji}a^1_i,$$

where $b^2_i$ is the $i$th bias in the second layer, $\omega_{ji}$ is the weight from the $j$th neuron in the first layer to the $i$th neuron in the second layer, $a^1_j$ is the $j$th neuron in the first layer, and $y^2_i$ is the $i$th neuron in the second layer, which is also the final output of the model.

4. Results and Evaluation

4.1. Datasets. To verify the effectiveness of the proposed model, extensive experiments are carried out on the Open University Learning Analytics (OULA) dataset [6], which is one of the datasets provided for learning analytics. OULAD collected the data of 32,593 students in the virtual learning environment (VLE), including demographics, clickstream history, activity type, course information, and assessment submission. The raw data are organized into seven tables: studentInfo, studentRegistration, studentAssessment, assessments, courses, studentVle, and vle. The dataset contains 22 courses belonging to 7 modules, named AAA through GGG: 4 Science, Technology, Engineering, and Mathematics (STEM) modules and 3 Social Science modules. Each module was presented at least twice. The final results of the courses are divided into four categories: Distinction, Pass, Fail, and Withdrawn. Students’ interaction with VLE is recorded and represented by the clickstream. The click activities are classified into 20 categories, each of which is associated with one kind of module material, namely, resource, ocontent, URL, homepage, subpage, glossary, foruming, oucollaborate, dataplus, quiz, ouelluminate, shared subpage, questionnaire, page, external-quiz, ouwiki, dualpane, repeatability, folder, and htmlactivity. Students’ interactions with VLE are recorded in studentVle and vle tables. The studentVle table consists of 10,655,280 rows with the column fields of code_module, code_presentation, id_student, id_site, date, and sum_click, which are module identification code, presentation identification code, student identification number, VLE material identification number, the day of student’s interaction with the material, and the number of times the student interacted with the material, respectively. The vle table contains information about the materials available in the VLE, and it consists of 6,364 rows with the column fields of id_site, code_module, code_presentation, activity_type, week_from, and week_to. Among these fields, id_site, code_module, and code_presentation have the same meaning as those in studentVle table. The activity_type field indicates which module material type the access behavior is associated with. There are 20 kinds of materials, just as mentioned above, so there are 20 kinds of behavior categories. Week_from and week_to indicate the week from which the material is planned to be used and the week until which the material is planned to be used, respectively. For most records in vle, week_from and week_to values are empty. For instance, suppose there is a record (AAA, 2013J, 28400, 546943, 15, 30) in studentVle table and a record (546943, AAA, 2013J, resource, Nan, Nan) in vle table; this means that the student with the id number of 28400 clicked the resource material of module AAA starting in October 2013 30 times in the 15th day after the beginning of the module.

The proposed model aims to predict whether the student will fail or drop out of the course. Therefore, we merge “Distinction” with “Pass” students to form a new category, labeled “Good,” and join “Fail” with “Withdrawn” students in a new type, tagged as “At_risk.” “Good” is the negative category and “At_risk” is the positive category. The details of the courses after processing are shown in Table 1:

4.2. Experimental Settings. The proposed model was implemented in Python 3.7 using the PyTorch 1.5.1 library. To predict whether the student is at risk of failing the course or dropping out of the course, we use the
The model parameters are then updated by backpropagation. The loss value of each batch can be computed by (13) and (14). First, the softmax operation, defined as (13), is performed to scale the instance’s outputs to value range $[0, 1]$ and the sum to 1, where $y_i^\text{c}$ represents the probability of the instance belonging to the $i$th category, $x_i^\text{c}$ is the $i$th output of the input instance, $K$ is the total number of the instance outputs (equal to the number of categories of the instances), and the category number here is two. Then, the binary cross-entropy loss, defined as (14), is calculated, where $y_i$ is the ground truth category of instance $i$, and $\hat{y}_i$ is the predicted probability of instance $i$ belonging to the ground truth category. Furthermore, the Adam optimization algorithm is adopted to minimize the loss function. The learning rate is set to 0.0001. The batch size is set to 50.

$$y_i = \frac{e^{x_i^\text{c}}}{\sum_{j} e^{x_j^\text{c}}}, \quad \text{(13)}$$

$$L = \frac{1}{N} \sum_{i} \left( y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right). \quad \text{(14)}$$

To verify the performance of the proposed model, it was compared with several commonly used approaches in the educational settings, including decision trees (DT) [2], support vector machine (SVM) [8, 10], and artificial neural network (ANN) [16], which is also called multilayer perceptron (MLP). These methods are also frequently adopted as baseline models by researchers. In order to deploy these baseline approaches, for each student, the click data of each activity are accumulated till the compared day and then fed into the baseline models. For the decision tree classifier, the Gini index is used to measure the quality of a split and use the best strategy to choose the split at each node, and the max depth of the tree was set to four. The radial basis function was chosen as the kernel function for the SVM classifier. For multilayer perceptron classifier, the hidden layer size is 100, the activation function is ReLU, and the optimizer is Adam. The maximum number of iterations is set to 200 to make sure the model converges and the learning rate is 0.001. All these methods are implemented in the sklearn 0.24.2. In addition, other parameters are set by default.

### 4.3. Metrics

The most commonly used metric for the classification prediction problem is the prediction accuracy, the proportion of the correct predicted instances to the total number of instances. Two other metrics are precision and recall. These two metrics, for most cases, are contradictory; that is, a high precision rate usually means a low recall rate, and vice versa. Because our model is aimed at the situation that there are a large number of students in the online learning environment and the teacher resources are very limited, it prefers precision rate rather than recall rate. The precision rate is the proportion of true positive cases to the cases that were predicted as positive cases. Moreover, AUC is also used as the model performance metric, which can overcome the bias introduced by imbalanced data and been commonly used in similar literature [37]. AUC is the area under the ROC (receiver operating characteristic) curve. Therefore, we mainly compare the significant differences between the proposed method and the baseline methods in terms of accuracy, precision, and AUC metrics in the experiments.

### 4.4. Results

In order to evaluate the performance of the proposed model and make the comparison more meaningful, we do the following for each course: (1) randomly dividing it into a training set (70%) and a validation set (30%), then training all methods on the training set, and using the validation set for validation; (2) repeating process (1) 10 times. In this way, each approach runs ten rounds in each course with the different training set and validation set, thus obtaining ten accuracy, precision, and AUC values. Then, we calculate the mean and standard deviation of these values, respectively. A paired two-tailed t-test is also conducted to compare the baseline results with the proposed model results.

In the OULA dataset, the course duration ranges from 234 to 269 days. As the goal is to predict the at-risk students to provide timely help and intervention, we select the early time point to make the prediction. In these experiments, the previous 70 days, that is, ten weeks’ click data of each course, were used for the prediction.

Figures 3–5 show the training and validation process of the courses BBB, CCC, and DDD, respectively. It can be seen from the figures that the accuracy, precision, and AUC of the model are increasing as the number of iterations increases. In particular, the precision rate of the model increases significantly. For example, the course CCC increases from about 70% at the beginning to more than 90% after 100 iterations. Overall, after 100 iterations, all indicators have reached convergence. Therefore, in the experiments, we set the number of iterations of model training to 100 uniformly.
Table 2 shows each method’s accuracy, precision, and AUC statistic results. For each metric, the average and deviation values were calculated and expressed in the form of mean ± std, and asterisks were used to indicate the significant difference level between the baseline method results and the proposed method CRRNN results on the corresponding metric as follows: ** the p value is larger than 0.01 and less than 0.05, * the p value is larger than 0.01 and less than 0.05. If there is no asterisk, the p value is larger than 0.05, which means that there is no significant difference between the corresponding baseline method results and the proposed method results.

The accuracy column of Table 2 shows that the proposed method CRRNN achieves the highest average accuracy on all
courses except course AAA. For the course AAA, there is no significant difference between CRRNN, MLPC, and DTC in prediction accuracy, and all of them have higher accuracy than SVC. It can be observed from the precision column of Table 2 that the prediction precision of the proposed method is much better than that of the baseline methods in most of the courses except course AAA. In addition, the prediction precision of all four STEM courses is higher than 85%. Especially for the course CCC, CRRNN obtains the best result, more than ten percent over other methods. For course
GGG, the prediction precision of CRRNN and DTC is almost the same, and it is much higher than that of SVC and MLPC. However, the precision of CRRNN is much lower than that of DTC and MLPC in the course AAA. From the AUC column of Table 2, it can be observed that the AUC value of the proposed method is significantly better than those of the baseline methods in all of the courses, which demonstrates the supremacy and robustness of the proposed model in predicting the at-risk student.

At the same time, we also note that all methods have poor effects on course AAA and course GGG. Through further research and analysis, it is found that there are two main reasons for this situation. One is that the number of instances is relatively small, and the other is that students’ online behavior is somewhat insufficient. It can be seen from Table 1 that the number of students in the course AAA is only 748. In addition, the click data of students is sparse, which is detrimental to the training of an effective model. In particular, the number of clicks per person on resources such as forum, home page, outcontent, subpage, and URL in course GGG is far less than that in other courses. In fact, this is not surprising because these two courses are Social Science courses. Furthermore, many liberal arts students may prefer to read offline or consult materials in the library rather than online learning and discussion, so students leave relatively few learning behaviors in the online learning system.

The small number of instances and sparse behavior data in some courses are not conducive to training a good neural network model. We wonder whether we can use similar courses’ data to improve the prediction model effect. Therefore, we studied the effect of migrating the prediction models of other courses to the current course. Courses AAA, BBB, and GGG are Social Science courses. We first pretrain a model with all the data of course BBB and then fine-tune the model with the data of course AAA and course GGG, respectively. As before, the data of the course AAA and course GGG are randomly divided into a training set (70%) and a verification set (30%); the training set is used to fine-tune the model, and the verification set is used to verify the effect of the fine-tuned model. We repeat this process ten times and compare it with the previous results. It is somewhat surprising that accuracy and AUC are not significantly different from the previous results. After fine-tuning the model, the accuracy and AUC of the model in course AAA were 75.96% (p value = 0.29) and 0.74 (p value = 0.28), respectively. After fine-tuning the model, the accuracy and AUC in course GGG were 72.37% (p value = 0.07) and 0.75 (p value = 0.10), respectively. However, the precision has been greatly improved. The precision of the model after being fine-tuned is 79.30% (p value = 0.0006) for course AAA and 79.00% (p value = 0.003) for course GGG.

5. Discussion and Conclusion

It is crucial for the instructors to monitor learners’ motivation and how it varies along the course weeks, and identifying potential at-risk students in advance holds many benefits. First of all, it can provide decision support for managers to formulate relevant policies to improve the learning environment. Secondly, instructors can deliver the intervention and help the target students to enhance their learning outcomes and reduce the dropout rate. Thirdly, it is helpful for educators to improve the teaching content, process, and arrangement to enhance the teaching effect further.

Clickstream is a kind of fine-grained data from which we can extract different information, such as the time of operation, the type of access resources, and the number of access times. To discover the hidden knowledge behind these data, we must correctly express this information and design effective feature extraction methods. The traditional way of accumulating the number of clicks is a low and straightforward data representation method. It may lose some changing information of students’ behavior and cannot sufficiently represent their behavior patterns. With consideration of the above problems, this paper proposed an approach, named CRRNN, to obtain the students’ activity characteristics representation of different periods and the changing patterns, by which we can distinguish good students from at-risk students.

The experimental results show that the proposed model can effectively predict the at-risk student in advance and outperform the baseline methods in precision, accuracy, and AUC metrics. For most courses, the proposed model can achieve an accuracy higher than 75% with precision of nearly 90%, which means the most potential at-risk students can be identified in less than one-third of the course duration. Such prediction has significant practical implications since it leaves sufficient time for institutes or instructors to formulate measures to improve retention rates.

It is worth noting that this paper only uses click data to predict at-risk students. In practice, many other factors will affect students’ learning, such as age, gender, place of birth, educational level, and other demographic information. For example, in Part V of Figure 2, we extract some personal information, including gender, region, highest education, age band, and disability, to make the prediction. These data were transformed into one-hot code and input into fully connected networks. We obtain an accuracy of 57.4%, precision of 58.7%, recall of 63.5%, and AUC of 0.6. The result demonstrates that this information has some effect in predicting the final results. However, the results did not show a significant difference when we added embedding personal information to Part III straightforwardly. Therefore, incorporating more information into the model to improve performance still needs further research.

Data Availability

The data used in this study are available at https://analyse.kmi.open.ac.uk/open_dataset.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61772352 and
References

[1] Y.-H. Hu, C.-L. Lo, and S.-P. Shih, “Developing early warning systems to predict students’ online learning performance,” Computers in Human Behavior, vol. 36, pp. 469–478, 2014.

[2] S. Rizvi, B. Rinties, and S. A. Khoja, “The role of demographics in online learning: A decision tree based approach,” Computers & Education, vol. 137, pp. 32–47, 2019.

[3] M. Hlosta, Z. Zdrahal, and J. Zendulka, “Ouroboros: early identification of at-risk students without models based on legacy data,” in Proceedings of the 7th Intl. Learning Analytics & Knowledge Conference, pp. 6–15, Vancouver, Canada, March, 2017.

[4] J. Bailey, R. Zhang, and B. Rubinstein, “Identifying At-Risk Students in Massive Open Online Courses,” in Proceedings of the Twenty-ninth Aai Conference on Artificial Intelligence, pp. 1749–1755, Texas, USA, January, 2015.

[5] A. Y. Q. Huang, O. H. T. Lu, J. C. H. Huang, C. J. Yin, and S. J. H. Yang, “Predicting students’ academic performance by using educational big data and learning analytics: evaluation of classification methods and learning logs,” Interactive Learning Environments, vol. 28, no. 2, pp. 206–230, 2020.

[6] J. Kuzilek, M. Hlosta, and Z. Zdrahal, “Open university learning analytics dataset,” Scientific Data, vol. 4, no.1, Article ID 170171, 2017.

[7] J. Kuzilek, J. Vaclavek, V. Fuglik, and Z. Zdrahal, “Student drop-out modelling using virtual learning environment behaviour data,” Lifelong Technology-Enhanced Learning, vol. 11082, pp. 166–171, 2018.

[8] K. T. Chui, D. C. L. Fung, and M. D. Lytras, “Predicting at-risk university students in a virtual learning environment via a machine learning algorithm,” Computers in Human Behavior, vol. 107, p. 7, 2020.

[9] Y. Yang, P. Fu, X. Yang, H. Hong, and D. Zhou, “MOOC learner’s final grade prediction based on an improved random forests method,” Computers, Materials & Continua, vol. 65, no. 3, pp. 2413–2423, 2020.

[10] F. Marbouti, H. A. Diefes-dux, and K. Madhavan, “Models for early prediction of at-risk students in a course using standards-based grading,” Computers & Education, vol. 103, pp. 1–15, 2016.

[11] S. Dass, K. Gary, and J. Cunningham, “Predicting student dropout in self-paced MOOC course using random forest model,” Information, vol. 12, no. 11, p. 476, 2021.

[12] M. Adnan, A. Habib, J. Ashraf et al., “Predicting at-risk students at different percentages of course length for early intervention using machine learning models,” Ieee Access, vol. 9, pp. 7519–7539, 2021.

[13] A. Guerrero-higueras, C. Llamas, and L. Gonzalez, “Academic success assessment through version control systems[J],” Applied Sciences-basel, vol. 10, no. 4, 2020.

[14] L. Wang, A. Sy, and L. Liu, “Deep Knowledge Tracing on Programming Exercises,” in Proceedings of the 4th (2017) Acm Conference on Learning @ Scale, pp. 201–204, Cambridge, USA, April, 2017.

[15] F. Okubo, T. Yamashita, and A. Shimada, “International Learning Analytics & Knowledge,” in Proceedings of the Seventh International Learning Analytics & Knowledge Conference, pp. 598–599, Coloumbia, Canada, March, 2017.

[16] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, “Predicting academic performance of students from VLE big data using deep learning models,” Computers in Human Behavior, vol. 104, no. Mar., Article ID 106189, 2020.

[17] L. Haiyang, Z. Wang, and P. Benachour, ”A Time Series Classification Method for Behaviour-Based Dropout Prediction,” in Proceedings of the 2018 Ieee 18th International Conference on Advanced Learning Technologies (icalt), pp. 191–195, IEEE, Mumbai, India, July, 2018.

[18] W. Wang, L. Guo, and L. He, “Effects of social-interactive engagement on the dropout ratio in online learning: insights from mooc,” Behaviour & Information Technology, vol. 38, no. 4, pp. 621–636, 2018.

[19] A. Sarra, L. Fontanella, and S. Di zio, “Identifying students at risk of academic failure within the educational data mining framework,” Social Indicators Research, vol. 146, no. 1, pp. 41–60, 2019.

[20] J. Russell, A. Smith, and R. Larsen, “Elements of Success: Supporting At-Risk Student Resilience through Learning Analytics,” Computers & Education, vol. 152, Article ID 103890, 2020.

[21] C. A. Boulton, C. Kent, and H. T. P. Williams, “Virtual learning environment engagement and learning outcomes at a ‘bricks-and-mortal’ university,” Computers & Education, vol. 126, pp. 129–142, 2018.

[22] M. Kloft, F. Stehler, and Z. Zheng, “Predicting Mooc Dropout over Weeks Using Machine Learning Methods,” in Proceedings of the Emmlp 2014 Workshop on Analysis of Large Scale Social Interaction in Moocs, Quatar, UAE, October, 2014.

[23] N. Alboukaey, A. Joukhadar, and N. Ghneim, “Dynamic behavior based churn prediction in mobile telecom,” Expert Systems with Applications, vol. 162, Article ID 113779, 2020.

[24] G. E. Hinton and R. R. Salakhudinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504–507, 2006.

[25] N. S. Raj, S. Prasad, P. Harish, M. Boban, and N. Cheryiayed, “Early prediction of at-risk students in a virtual learning environment using deep learning techniques,” in Adaptive Instructional Systems. Adaptation Strategies and Methods, pp. 110–120, Springer, Cham, Singapore, 2021.

[26] S. Hochreiter, J. Schmidhuber, and S. Hochreiter, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] K. Cho, B. V. Merrienboer, and C. Gulcehre, “Learning Phrase Representations Using Rnn Encoder-Decoder for Statistical Machine Translation,” 2014.

[28] L. Xiang, G. Guo, G. Guo, J. Yu, V. Sheng, and P. Yang, “A convolutional neural network-based linguistic steganalysis for synonym substitution steganography,” Mathematical Biosciences and Engineering, vol. 17, no. 2, pp. 1041–1058, 2020.

[29] S. Wan and S. Goudos, “Faster R-Cnn for Multi-Class Fruit Detection Using a Robotic Vision System,” Computer Networks, vol. 168, 2020.

[30] O. Yildirim, U. B. Baloglu, and U. R. Acharya, “A deep convolutional neural network model for automated identification of abnormal eeg signals,” Neural Computing & Applications, vol. 32, no. 20, Article ID 15857, 2020.

[31] L. P. Mafadyen, S. Dawson, and L. P. Mafadyen, “Mining LMS data to develop an “early warning system” for educators: a proof of concept,” Computers & Education, vol. 54, no. 2, pp. 588–599, 2010.

[32] A. Ramesh, G. Dan, and B. Huang, “Modeling Learner Engagement in Moocs Using Probabilistic Soft Logic,” in
Proceedings of the Nips Workshop on Datadriven Education, Lake Tahoe, Nevada, USA, December 2013.

[33] X. Zhang, G. Sun, Y. Pan, H. Sun, Y. He, and J. Tan, "Students performance modeling based on behavior pattern," Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 5, pp. 1659–1670, 2018.

[34] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair," in Proceedings of the International Conference on International Conference on Machine Learning, pp. 807–814, Haifa, Israel, June, 2010.

[35] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," Advances in Neural Information Processing Systems, vol. 26, 2013.

[36] Proceedings of the eee Conference on Computer Vision and Pattern Recognition, IEEE, New York, USA, 2016.

[37] M. Mouríño-garcía, R. Pérez-rodríguez, and L. Anido-rifón, "Cross-repository aggregation of educational resources[J]," Computers & Education, vol. 117, pp. 31–49, 2018.