Student Academic Performance Prediction Using Deep Multi-source Behavior Sequential Network

Xiang Li[1], Xinning Zhu[1(✉)], Xiaoying Zhu[2], Yang Ji[1], and Xiaosheng Tang[1]

[1] Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, China
{lixiang14,zhuxn,jiyang,txs}@bupt.edu.cn
[2] Information Technology Center, Beijing University of Posts and Telecommunications, Beijing, China
zhuxy@bupt.edu.cn

Abstract. Online education is becoming increasingly popular and often combined with traditional place-based study to improve learning efficiency for university students. Since students have left a large amount of online learning data, it provides an effective way to predict students' academic performance and enable pre-intervention for at-risk students. Current data sources used to predict students' performance are limited to data just from the corresponding learning platform, from which only learning behaviors on that course can be observed. However, students' academic performance will be related to other behavioral factors, especially the patterns of using Internet. In this paper, we utilize two types of datasets from 505 university students, i.e., online learning records for a project-based course, and network logs of university campus network. A deep learning framework: Sequential Prediction based on Deep Network (SPDN) is proposed to predict students' performance in the course. SPDN models students' online behavioral sequences by utilizing multi-source fusion CNN technique, and incorporates static information based on bidirectional LSTM. Experiments demonstrate that the proposed SPDN model outperforms the baselines and has a significant improvement on early-warning. Furthermore, it can be learned that Internet access patterns even have a greater impact on students' academic performance than online learning activities.

Keywords: Educational data mining · Multi-source online behaviors · Student performance prediction · Student clustering

1 Introduction

Since online learning can generate large amounts of records in students' learning process, it provides an effective way to get deep understanding of students' learning behaviors and predict their academic performance. Due to the benefits of online learning, more and more universities combine traditional place-based courses with online education to achieve better teaching results. For this kind of course, it is feasible to give early predictions of

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 567–579, 2020.
https://doi.org/10.1007/978-3-030-47426-3_44

the students' final performance through the student's online learning records, so that a timely pre-intervention could be carried out for at-risk students.

In this paper, we conduct research on university students' academic performance prediction for a course which combines the online learning and traditional place-based learning. While current researches generally focus on the learning behaviors records collected from the corresponding learning management system, but ignoring other factors that may be potentially relevant to students' academic performance. As indicated in [15], internet access activities were discovered to be a major factor affecting students' academic performance. Both users' online behaviors which can be clustered into several distinct pattern [14] and students' static information [5] have impacts on the academic performance prediction. In this study, two types of data are collected from 505 anonymous students to predict at-risk students in a university project-based course. One dataset records the students' online learning activities of the course which provides a learning platform for self-study. The other collects Internet access activity data from the campus network logs, from which the students' behavioral patterns of accessing the Internet can be explored. Combining these two datasets will obtain deep insights into students' learning behaviors and the correlation with their academic performance.

The remaining part of this paper is organized as follows. Section 2 reviews the related work on the EDM techniques for predicting student's performance and feature learning from time series. Section 3 mentioned two types experimental datasets used in detail. Section 4 presents the proposed SPDN model. Experimental results are described in Sect. 5 and finally Sect. 6 concludes this work and discusses future avenues of research.

2 Related Work

2.1 Related Methods of Education Data Mining

There has been a large amount of relevant work about the student performance prediction. The current methods of EDM are generally divided into two categories. The first traditional method relies on machine learning methods for binary classification prediction. In [1, 3, 4, 13], each machine learning model considers different types of predictive features extracted from raw online learning activity records to predict whether students can graduate on time. Also generalized linear model is used to predict students' dropout by extracting features from the original learning website log files such as page click rate, forums and so on [2, 10]. The second emerging approach involves the exploration of neural networks (NN). Because deep learning achieves better performance than traditional machine learning in many respects, work has been done to predict students' dropout in MOOC through deep neural network (DNN) models [10] and recurrent neural network (RNN) models [6]. Different from all current methods which still rely on feature engineering to reduce the input dimension and limit the development of larger NN models, Kim et al. [11] propose GritNet which extracts the original learning behavior sequence from network log as raw input of the RNN model. It outperforms the standard logistic-regression based method without complex feature engineering.

2.2 CNN for Behavioral Feature Learning of Time Series

The method of learning a time series feature is to represent a sequence of behaviors within a time window as a low-dimensional vector. KimCNN [12] is a typical CNN structure, which applies the convolution operation with several different size kernels on every possible location of the activity vector matrix, and use max-pooling to get the most prominent feature. In this way, it can automatically extract the features of the behavior sequence, and the model can be easily transferred to other datasets. KimCNN has been used in news recommendation to fuse semantic-level and knowledge-level representations of news [17]. Wang et al. proposed knowledge-aware CNN (KCNN) to treat words and entities as multiple channels instead of simple concatenating, and explicitly keeps their alignment relationship during convolution. In this way, it is suitable to connect words and associated entities and convolute them together in a single vector space. In this paper, we implement this structure to fuse student Internet access activities and learning activities and learning student behavioral representation in MFCNN component.

3 Dataset Description and Insight

The analysis in this work is based on two datasets from 505 anonymous students. One of the datasets is website log of a university project-based course for freshmen which involves students' online learning activities. The other one is the campus network logging record which reflects students' Internet access activities.

In this section, we will introduce and describe the details of online learning activities with the university project-based course and Internet access activities. Then, we investigate four distinct online behavior patterns by clustering.

3.1 Online Learning Activity

Students' online learning activities are extracted from the online learning website log of a university project-based course which spans over 13 weeks from 2018.9.28 to 2018.12.27. This course aims to help freshmen students get started in communication engineering and its greatest characteristics is implementing the online education combining with traditional class. The course is taught by teachers every Friday and students can learn on course's wiki and forum messages from the online learning website, also create their own wiki post or participate in the forum. Meanwhile, all online learning activities of students will be recorded in the website log as online learning activity sequence. Table 1 lists statistics of actions in this dataset which involves two categories of activities such as viewing and writing. Each category involves six activities and we have a more detailed distinction between different types of web pages for each activity.

In addition, there is a weekly quiz on each Wednesday which are scored by the teachers. And the students are grouped to do the final innovation project which are scored by the teachers. The final performance of students consists of two parts: average score of weekly quiz and the final innovation project score. In this paper, we judge the at-risk students based on the course results. Specifically, students are considered at-risk

Table 1. Statistics of learning behavior dataset of the Introductory course

Category	Activity	#Type
View	# Learning the theoretical basics from course's wiki (i.e. wiki)	4724
	# Learning the requirements of project (i.e. project)	806
	# Files and images in the post (i.e. attachment)	15
	# Viewing the questions raised in the forum (i.e. question)	211
	# Viewing the answer in the forum (i.e. answer)	94
	# Other pages (i.e. other)	32
Write/Create	# Adding terms to the wiki (i.e. w_wiki)	4650
	# Creating a post to introduce the own project (i.e. w_project)	195
	# Asking a question in the forum (i.e. w_question)	103
	# Answering a question in forum (i.e. w_answer)	91
	# Editing the post of project (i.e. w_revision)	4021
	# Uploading the files or images (i.e. w_attachment)	3942

students whether their average score of quizzes or innovation project scores is at the last 25% of the whole grade. Because he or she is lacking in theory or practice. In our dataset, there are 202 at-risk students in total.

3.2 Internet Access Activity

The campus network can record the students' internet access activities in the log file which contains the categories of URLs and corresponding timestamp. There are 11 categories of internet access activities, namely: 'News', 'Game', 'Music', 'Download', 'File transfer', 'Search engine', 'Video', 'Shopping', 'Living tools', 'Instant messaging' and 'Non-instant messaging'. The log file holds a total of 22 million records for 505 students during the semester of the project-based course.

In order to build a complete student online activity sequence, we converge the students' online learning activities with current Internet access activities based on the students' anonymous IDs. We rename the all online learning activities to a new category of internet access activity, 'Learning', and merge them with the original internet access activities in chronological order as the new internet access activities. In order to ensure that the online learning activity sequence and the Internet access activity sequence are aligned in the time dimension, we use zero padding to complete the online learning activity sequence.

3.3 Distinct Behavior Patterns and Static Information

To investigate the different online habits, we conducted a cluster analysis and feed the normalized frequency counts of each action of all students into Ward's hierarchical cluster algorithm [7]. The number of clusters is set to 4 based on Calinski-Harabasz

(CH) index [16] on the data. Table 2 shows the students' number and at-risk rate of four clusters. It illustrates that cluster1 and cluster 2 have low at-risk rate and high proportion, while nearly a quarter of the student in cluster 3 and cluster 4 are at-risk.

Table 2. Statistics of four clusters

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Total #student	276	111	74	44
At-risk rate	0.18	0.14	0.28	0.25

Specifically, Fig. 1 illustrates the proportion of occurrence frequency of each activity in different clustering patterns. The proportion is calculated by the frequency of the particular activity divided by the count of all activities for each case. And in each cluster, we calculated the average of the above proportions across all the cases which are assigned to the particular cluster. In the x-axis, we list different actions of original internet access activities and online learning activities. It can be seen that there are obvious differences between clusters. The overall access internet frequency of students in cluster 1 is very low, so they may prefer offline learning. Students in Cluster 2 often use search engines, which may be related to learning. Conversely, Cluster 3's students prefer to watch videos and use life tool applications which may not be educational. On the learning website, they ask and answer questions in the forum relatively frequently, but there are few viewing actions. Cluster 4 has the fewest numbers, but is extremely focused on online games and rarely involves other types of online activities. On the learning website, their learning behavior is relatively inactive, which may also be the reason why the student's at-risk rate is high in the cluster.

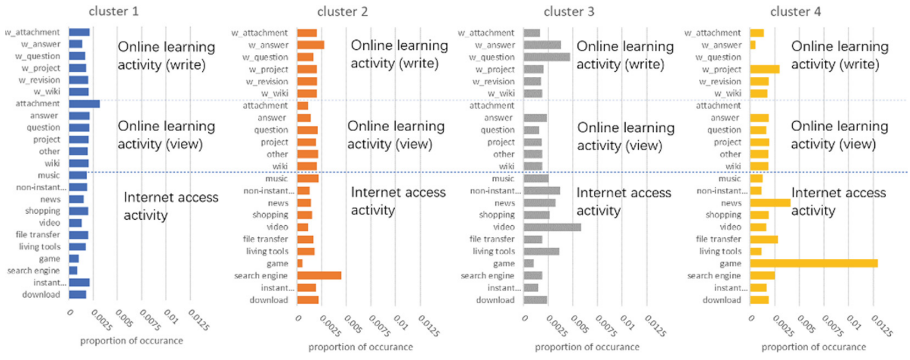

Fig. 1. The four cluster interaction patterns

In addition, students experiment in a group and always learn together in a group. So students in the same group will have a high probability of having the same academic status and the grouping information has important impact on prediction. For the reason

above, we take the student's group id and cluster patterns as static information and joint them into the framework to model the prediction of student performance.

4 Framework of Sequential Prediction Based on Deep Network

The overall framework of sequential prediction based on deep network (SPDN) is shown in Fig. 2 and can be divided into four parts roughly. In this section, we first introduce the process of constructing and embedding the complete input sequence in input representation component. Then we will discuss the details of Multi-source fusion CNN (MFCNN) which represents the student's multiple activity sequences in weeks. After that, we will present the process of joining static information with students' behavioural representation and feed them into bi-LSTM model for prediction. Let us begin with a formulation of the problem we are going to address.

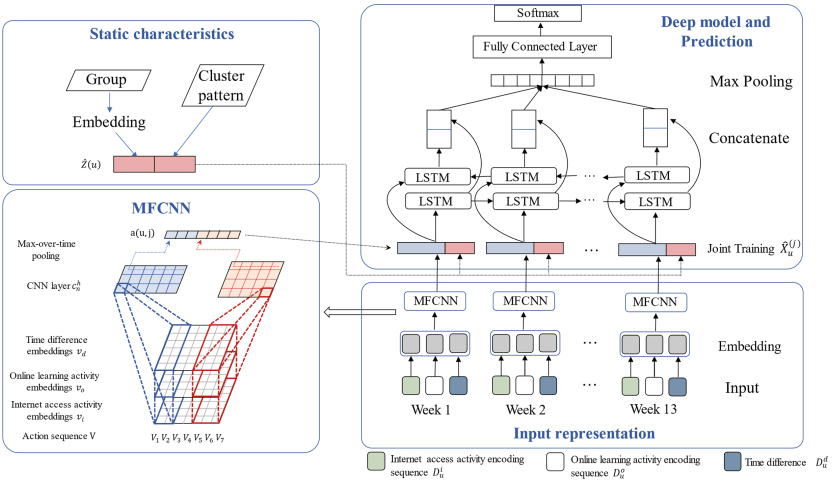

Fig. 2. The architecture of SPDN

4.1 Formulation

As introduced in Sect. 3.2, we converge the campus network logging records and the course's website log to build a complete student u's Internet access activity sequence, and complement the online learning activity sequence by zero padding. In order to formulate the problem more precisely, we first introduce the following definitions.

Definition 1. Internet Access Activity. Let $\boldsymbol{I}$ denote the set of Internet access activities. The complete student u's Internet access activity sequence can be formulated into $\hat{I}(u) = i_{1:M} = [i_1, i_2, \ldots, i_M]$, where M is the length of weekly Internet access activity sequence. Each element i_t is defined as a paired tuple of $\left(a_t^i, d_t\right)$, where a^i represents the Internet access activities such as "Game", "Music" or "Learning" and d_t is the corresponding timestamp at time t.

Definition 2. Online Learning Activity. Let $\boldsymbol{O}$ denote the set of online learning activities. Student u's zero-padding online learning activity sequence which can be formulated into $\widehat{O}(u) = o_{1:N} = [o_1, o_2, \ldots, o_N](N = M)$, where N is the length of weekly online learning activity sequence. Each element o_t is defined as a paired tuple of $\left(a_t^o, d_t\right)$ which a_t^o is the online learning activity with zero-padding at time t. If a_t^i is "Learning", $a_t^o \in O$, otherwise, $a_t^o = 0$.

Definition 3. Static Characteristics. Static information comprises student u's group id Z_g and cluster pattern Z_p. These characteristics do not vary over time and can be concatenated and represented by a vector $Z(u)$.

Definition 4. Time Difference. Since directly employing each timestamp d_t will increase the input space too fast, we define the discretised time difference between adjacent events as:

$$\Delta d_t = d_{t+1} - d_t \tag{1}$$

In this way, each activity sequence will be accompanied by a time difference sequence as $\widehat{T}(u) = [\Delta d_1, \Delta d_2, \ldots, \Delta d_N](N = M)$.

Problem Formulation. With these definitions, our task of predicting student performance can be expressed as a sequential event prediction problem: given student u's Internet access activity $\widehat{I}(u)$, online learning activity $\widehat{O}(u)$ in first $j(j \leq 13)$ weeks of the semester, as well as static characteristics $Z(u)$, our goal is to predict whether u will be at-risk in the course. More precisely, let y(u) $\in \{0, 1\}$ denotes the ground truth of whether u is at-risk, y(u) is positive if and only if u is at-risk in the course. Then our task is to learn a function:

$$\mathrm{f} : \left(\widehat{I}(u), \widehat{O}(u), \widehat{T}(u), Z(u),\right) \rightarrow \mathrm{y(u)} \tag{2}$$

4.2 Input Representation

In order to feed students' activity sequence into the SPDN, we transform each online learning activity a_t^o, Internet access activity a_t^i and time difference Δd_t into one-hot encoded feature vector $1\left(a_t^o\right) \in \{0, 1\}^{L_o}, 1\left(a_t^i\right) \in \{0, 1\}^{L_i}, 1(\Delta d_t) \in \{0, 1\}^{L_d}$, where L_o, L_i and L_d respectively are the number of online learning activity unique types, Internet access activity unique types and hours of the week. The student u's encoding vectors are represented by $D_u^o = \left[1\left(a_1^o\right), 1\left(a_2^o\right), \ldots, 1\left(a_M^o\right)\right] \in R^{M \times L_o}$, $D_u^i = \left[1\left(a_1^i\right), 1\left(a_2^i\right), \ldots, 1\left(a_M^i\right)\right] \in R^{M \times L_i}$ and $D_u^d = [1(\Delta d_1), 1(\Delta d_2), \ldots, 1(\Delta d_M)] \in R^{M \times L_d}$.

Then each one-hot vector is converted to a dense vector through an embedding layer. That means to learn three embedding matrixes $E_o \in R^{e \times L_o}$, $E_i \in R^{\mathrm{e} \times L_i}$, and

$E_{\mathrm{d}} \in R^{e \times L_{\mathrm{d}}}$, where e is the embedding dimension. The low-dimensional embedding vectors of online learning activity, Internet access activity and time difference are defined as:

$$\begin{cases} v_o = E_o \cdot 1\left(a_t^o\right) \\ v_i = E_i \cdot 1\left(a_t^i\right) \\ v_d = E_d \cdot 1(\Delta d_t) \end{cases} \tag{3}$$

The dimensions of the various embedded vectors are the same and similar events appear to be closer in the embedding event space.

4.3 Multi-source Fusion CNN (MFCNN)

Following the process used in Sect. 4.2, the next step is multi-source fusion. We employ the MFCNN component which is multi-channel and multiple-activities-aligned to compress the representation of the student's three types of embedding activity sequences per week. They can be regarded as representations of multiple different channels of the same action. We align and stack the three vector matrices $V = [[v_{o1}\ v_{i1}\ v_{d1}][v_{o2}\ v_{i2}\ v_{d2}]\ldots[v_{oM}\ v_{iM}\ v_{dM}]] \in R^{e \times M \times 3}$. Then similar to KimCNN [12] introduced in Sect. 2.2, we use multiple convolution kernels $\mathrm{h} \in R^{e \times k \times 3}$ to extract a particular local pattern in the action sequence, while $k(k \leq M)$ is window size. The local activation of the submatrix $V_{n:n+k-1}$ with respect to the convolution kernel h can be recorded as:

$$c_n^h = f(h * V_{n:n+k-1} + b)(0 \leq n \leq M - k + 1), \tag{4}$$

where f is the nonlinear function and * is the convolution operator and b is the bias.

Then we use the max pooling operation on the feature map of the output as:

$$\widetilde{c}^h = \max\left\{c_1^h, c_2^h, \ldots, c_{M-k+1}^h\right\} \tag{5}$$

All the features are concatenated together to form the final representation $a(u, j)$ of the student u's online behavior in $j^{th}(0 \leq j \leq 13)$ week $\mathrm{a(u, j)} = \left[\widetilde{c}^{h_1}\widetilde{c}^{h_2}\ldots\widetilde{c}^{h_m}\right]$, where m is the number of kernels. The weekly online behavioral representation will be passed into the bi-LSTM with static information.

4.4 Static Characteristics Component

This component builds a simple effective strategy to incorporate group id Z_g and cluster pattern Z_p into SPDN. Since these characteristics are categorical values, we model them into one-hot vectors as $1\left(Z_g\right) \in \{0, 1\}^{L_g}$ and $1\left(Z_p\right) \in \{0, 1\}^{L_p}$, where L_g is the number of students learning group and L_p is the cluster pattern types. And embed the group encoding vector $1\left(Z_g\right)$ and convert it into a low-dimensional embedding vector $v_{Z_g} \in R^{e_g}$, where e_g is the dimension of the embedding vector. The student u's static characteristics can be represented by $\widehat{Z}(u) = \left[v_{Z_g} \oplus 1\left(Z_p\right)\right] \in R^{e_g + L_p}$. Then we join the same static feature vectors $\widehat{Z}(u)$ with students' weekly behavioural representation

a(u, j) as shown in Fig. 2. Let $\widehat{X} = \widehat{X}_u^{(1)} \oplus \widehat{X}_u^{(2)} \oplus \ldots \oplus \widehat{X}_u^{(j)}$ represents the augmented feature vector, where each $\widehat{X}_u^{(j)} \in R^{e_g+L_p+k}$ is a fused feature group which consists of student u's weekly online behavioural representation $a(u, j)$ and his or her static characteristics: $\widehat{X}_u^{(j)} = \left[a(u, j) \oplus \widehat{Z}(u)\right]$.

4.5 Bi-LSTM and Prediction

The fused feature groups of each week are passed into bi-LSTM [8] and the output vectors are formed by concatenating each forward and backward direction outputs. The purpose of bi-LSTM is to make full use of context information and prevent gradient explosion. Then a max pooling layer is added to learn the most relevant part of the event embedding sequence and the output is fed into a fully connected layer and a Softmax layer sequentially to estimate the student's at-risk probability $\hat{y}(u) \in [0, 1]$.

The parameters to be updated in the whole framework SPDN mainly come from four parts, 1) embedding layers parameters. 2) CNN parameters. 3) bi-LSTM parameters and 4) fully connected layer parameters. All the parameters can be learned by minimizing the follow binary cross entropy objective function:

$$L(\theta) = -\sum\nolimits_{u \in U} \left[y(u) \log\left(\hat{y}(u)\right) + (1 - y(u)) \log\left(1 - \hat{y}(u)\right)\right], \tag{6}$$

where θ denotes the set of model parameters, $\hat{y}(u)$ is the probability of student at risk, $y(u)$ is the corresponding ground truth, U is the set of the whole students.

5 Experiments

We conduct various experiments to evaluate the effectiveness of SPDN on the online action datasets of 505 anonymous students and adopt Adam to optimize the model.

5.1 Setting

In our experiments, we divide all behaviour sequences into 13 weeks and encode respectively as input series. The inter-event time interval is an hour. The embedding dimensions of Internet access activity, online learning activity and time difference are 100 while the dimension of embedding group id is 50 and the cluster patterns are one-hot encoding. In the MFCNN, we use 64 different kernels which the window size of kernel is 1. The bi-LSTM with forward and backward LSTM layers containing 64 cell dimensions per direction is used. In addition, batch normalization layer [9] is applied to the bi-LSTM output and fully connected layer output. It can avoid gradient disappearance problems and speed up the training with a mini-batch size of 64. We divide 64% of the data set into a training set, 16% is a validation set, and 20% is a test set.

All the parameters above are the best group in all experiments with the grid search. Since the true binary target label is imbalanced, the evaluation metrics include Accuracy, Area Under the ROC Curve (AUC) and F1 Score (F1).

5.2 Baseline Models

In order to assess how much added value is brought by the SPDN, we set several baseline models to compare.

To compare with the universal deep learning method, we take the **BLSTM_MA** (Bidirectional Long Short-Term Memory with Multiple Activity) as a baseline model. We encode and embed the activities in the same way as SPDN, however, in order to show the effect of MFCNN, we align and stack the Internet access activity embeddings, online learning activity embeddings and time difference embeddings vector matrices in multi-channel and use the max pooling operation to get the features on each dimension instead of extracting the features by CNN. In addition, other parameters are consistent with the experimental parameters of SPDN.

For other baseline models **LR** (logistic regression model), **NB** (Naive Bayesian), **DT** (Decision Tree) and **RF** (random forest), we use the bag of words (BoW) model to represent each student's past event sequence. After transforming all students' activities into a BoW model, we count the number of each unique activity appearing in weekly sequence as the part of input. The group id and cluster pattern are other parts of input. The purpose of these experiments is to demonstrate the effectiveness of deep learning.

5.3 Prediction Performance

Table 3 presents the results on the test set for all comparison methods. Overall, SPDN gets the best performance on the dataset. Furthermore, BLSTM_MA and SPDN have the clearly better performance than other traditional machine learning algorithms, that means deep learning models can automatically get more effective information from the activity sequence. Moreover, as the F1 score is a weighted average of both precision and recall, thus it provides more comprehensive evaluation of the model. In our problem, the higher F1 of positive sample is excepted. As can be observed clearly, SPDN gets higher F1 score of positive samples than BLSTM_MA, so it shows that extracting features through MFCNN can provide advantages for predicting positive samples.

Table 3. Overall results

Approaches	Accuracy (%)	AUC (%)	F1	
			Positive	Negative
SPDN	**73.51**	**79.67**	**0.65**	**0.78**
BLSTM_MA	70.30	76.31	0.57	0.76
LR	52.48	52.20	0.41	0.59
NB	53.27	58.09	0.49	0.57
RF	61.78	54.64.	0.32	0.73
DT	65.15	60.09	0.51	0.72

In order to identify the importance of different kinds of engagement activities in this task, we conduct feature ablation experiments for three parts of input, i.e. online learning

activity, Internet access activity and static characteristics. Specially, we first input three parts of input to the SPDN, then remove every type of activity one by one to observe the variety of performance.

The results are shown in Table 4. We can observe that all three inputs are useful in this task, especially static information. Because when it is removed, the experimental result of AUC steeply drops to 0.7419. Furthermore, Internet access activity play a more important role, while the student's online learning activity is sparser than Internet access activity, so it is less important.

Table 4. Contribution analysis for different engagement activities

Removed feature	Accuracy	AUC	F1	
			Positive	Negative
Total	0.7129	0.7911	0.66	0.74
Online learning activity	0.7364	0.7831	0.654	0.78
Internet access activity	0.6908	0.7771	0.644	0.728
Static characteristic	0.7128	0.7419	0.62	0.77

5.4 Early Prediction

As shown in the Fig. 3, with the accumulation of activity sequences, the performance of the SPDN and baseline models gradually improve from the perspective of AUC. But it can be clearly seen that the deep learning model always has a higher AUC than the general machine learning model (Fig. 3 only shows one of machine learning baseline models, RF, and others have the similar trend). Meanwhile, one of deep models BLSTM_MA

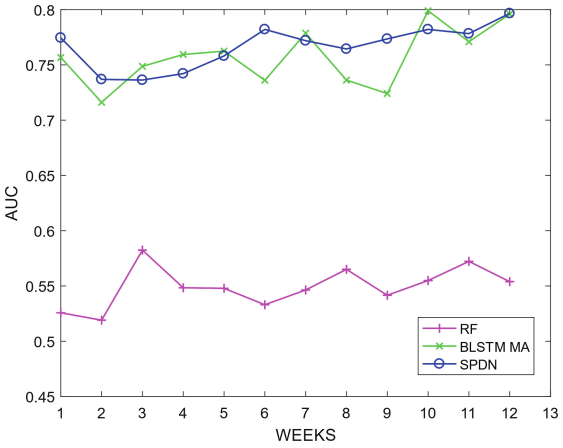

Fig. 3. Comparisons of the SPDN and baseline models in terms of mean AUC for early prediction

requires 11 weeks of student data to achieve the same performance as SPDN is able to achieve significant prediction-quality improvements within the first seven weeks of the semester. It illustrates that MFCNN can form the suitable weekly representation vector of user behaviour and extract features from a long behaviour sequence. In this way, SPDN can be used to early prediction and it can promote early intervention by teachers.

6 Conclusion

In this paper, we propose the model named SPDN, which fully uses online learning activities and Internet access activities and joins the static information to predict the performance of the students based on bi-LSTM. Through the experiments on the dataset of a university project-based course and the anonymous student's network logging records, the results show that SPDN gets the best performance and can achieve results close to the final value within the early weeks to find the at-risk students in time. Meanwhile, Internet access activities have a greater impact on students' academic performance prediction. In the future, we can combine more courses information into the model to make it more scalable.

Acknowledgements. This work is supported by the project "Virtual Simulation Experiment of Engineering Cognition and Innovation Diathesis Cultivation for Freshmen".

References

1. Tamhane, A., Ikbal, S., Sengupta, B., Duggirala, M., Appleton, J.: Predicting student risks through longitudinal analysis. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 1544–1552. ACM, New York (2014)
2. Bailey, J., Zhang, R., Rubinstein, B., et al.: Identifying at-risk students in massive open online courses. In: AAAI (2015)
3. Aguiar, E., Lakkaraju, H., Bhanpuri, N., Miller, D., Yuhas, B., Addison, K.: Who, when, and why: a machine learning approach to prioritizing students at risk of not graduating high school on time. In: Proceedings of the 5th Learning Analytics and Knowledge Conference. ACM (2015)
4. Er, E.: Identifying at-risk students using machine learning techniques: a case study with is 100. Int. J. Mach. Learn. Comput. **2**(4), 279 (2012)
5. Feng, W., Tang, J., Liu, T.X.: Understanding dropouts in MOOCs. In: AAAI, 2019 (2019)
6. Mi, F., Yeung, D.-Y.: Temporal models for predicting student dropout in massive open online courses. In: Proceedings of 15th IEEE International Conference on Data Mining Workshop (ICDMW 2015), Atlantic City, New Jersey, pp. 256–263 (2015)
7. Murtagh, F., Legendre, P.: Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? J. Classif. **31**(3), 274–295 (2014)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on International Conference on Machine Learning. JMLR.org (2015)

10. Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., Tingley, D.: Delving deeper into mooc student dropout prediction. arXiv preprint arXiv:1702.06404 (2017)
11. Kim, B.H., Vizitei, E., Ganapathi, V.: GritNet: student performance prediction with deep learning (2018)
12. KIMY: Convolutional neural networks for sentence classification. In: EMNLP. [S.l.]: [s.n.] (2014)
13. Lakkaraju, H.: A machine learning framework to identify students at risk of adverse academic outcomes. In: KDD (2015)
14. Lee, S.Y., Chae, H.S., Natriello, G.: Identifying user engagement patterns in an online video discussion platform. In: (Education Data Mining) EDM (2018)
15. Shahiri, A.M., Husain, W., Rashid, N.A.: A review on predicting student's performance using data mining techniques. Proc. Comput. Sci. **72**, 414–422 (2015)
16. Calinski, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat. - Theory Methods **3**(1), 1–27 (1974)
17. Wang, H., Zhang, F., Xie, X., et al.: DKN: deep knowledge-aware network for news recommendation (2018)