








Predicting Dropouts Before Enrollments in MOOCs: An Explainable and Self-Supervised Model

Jin Li , Member, IEEE, Shu Li , Yuan Zhao , Longjiang Guo , Member, IEEE, Fei Hao ,
Meirui Ren , Member, IEEE, and Keqin Li , Fellow, IEEE

Abstract—Massive Open Online Courses (MOOCs) belong to a new cloud-based service in education that suffers from low completion rates. Effective pre-learning intervention services, such as recommending courses with a high probability of completion or filtering courses with a very low probability of completion, will encourage students to spend more time and energy on proper courses, thus can reduce the dropout ratio. In practice, intervention services are introduced when students are predicted to drop out. However, existing methods concentrate on analyzing students' learning actions and predicting final dropout after a period of enrollment, which are insufficient in preventing students from enrolling in unsuitable courses and withdrawing mid-way. This paper presents a neural network-based Explainable Self-supervised Model (ESM) to predict MOOC dropout before enrollment. Specifically, the student's learning actions on an unenrolled course are estimated using previous logs by the neural network. And then, the action's contribution to the completion of a course is calculated in a similar way. Therefore, the probability of completion for an unenrolled course is predicted by aggregating the learning actions and their contribution to the completion. To train the neural network, a self-supervised training strategy is proposed, where enrolled courses in the training data are randomly selected as validation in each epoch. The ESM outperforms existing methods in terms of prediction accuracy and efficiency. The average increment of Area Under the ROC Curve (AUC) and F-score (F1) in the two MOOCs datasets, XuetangX and KDDCUP, are 8.3% and 0.6%, respectively. Furthermore, the two pre-learning intervention services named courses recommendation and courses filtration are proposed. When courses are recommended, the completion rate increased from 22% to 60% in XuetangX, and from 27% to 45% in KDDCUP. By filtering courses predicted with low completion

probability, 40% wasted time in uncompleted courses will be saved in XuetangX.

Index Terms—Dropout prediction, MOOCs, neural network, pre-learning intervention service, self-supervised training.

I. INTRODUCTION

WITH the rapid development of computing technologies, there has been a surge of mobile applications that impact various aspects of modern life, including personalized education, healthcare, and entertainment services [1]. In education, cloud computing offers desirable properties for delivering e-learning services, particularly in scenarios where these services are computer-intensive. Massive Open Online Courses (MOOCs) platforms, for example, gather resources such as virtual worlds, simulations, video streaming, and more [2], [3]. This new form of education based on e-learning services generates a vast amount of log information, including information on students and courses, students' actions, and more, which is incredibly valuable for prediction and pre-learning intervention services aimed at improving the quality of education.

The past decades have witnessed the rapid development of MOOCs, where people can get access to learning resources anywhere and anytime. However, e-learning services suffer from a low completion rate [4], [5] because many students often enroll in multiple courses without considering their ability for completing them. This makes it difficult to guarantee the quality of MOOCs. To achieve a better effect of this service, platforms should first predict the objective event of whether students drop out of enrolled courses, which is useful to analyze the deep reason for dropout. Many pioneer researchers contributed to the analysis of the dropout in MOOCs [6], [7], [8], where both prediction method and quantitative analysis of dropout are proposed. After predicting the dropout, the second step is to develop learning intervention services for enrollments (pairs of students and courses) with a high probability of dropout. For example, MOOCs platforms could recommend courses to students before enrollment, or provide learning suggestions during student's study. Literature [9] is an early work for dropout analysis and intervention designing. More importantly, earlier dropout prediction leads to earlier learning intervention services by MOOCs, significantly improving the quality of online platforms.

Since finding the potential dropout is the first step to revealing the deep reason for dropout and designing the learning intervention services, MOOC platforms, instructors, and students stand

Manuscript received 6 March 2023; revised 21 July 2023; accepted 29 August 2023. Date of publication 4 September 2023; date of current version 13 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 61977044 and 62206162, in part by the Second Batch of New Engineering Research and Practice Projects of the Ministry of Education of China under Grant E-RGZN20201045, in part by the Natural Science Basis Research Plan in Shaanxi Province of China under Grant 2020JM-302, in part by the Fundamental Research Funds for the Central Universities under Grant GK202205037, in part by the Ministry of Education's Cooperative Education Project under Grant 202102591018, and in part by CCF-Tencent Open Fund under Grant RAGR20220127. Recommended for acceptance by R. Mizouni. (Shu Li is co-first author.) (Corresponding authors: Longjiang Guo; Fei Hao.)

Jin Li, Shu Li, Longjiang Guo, Fei Hao, and Meirui Ren are with the School of Computer Science, Shaanxi Normal University, Xi'an 710062, China, and also with the Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China (e-mail: jin.li@snnu.edu.cn; ls980108@163.com; longjiangguo@snnu.edu.cn; fhao@snnu.edu.cn; meiruiren@snnu.edu.cn).

Yuan Zhao is with the College of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu Province 730070, China (e-mail: zhaoyuan2233@163.com).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TSC.2023.3311627

to gain from the ability to predict and analyze the dropout before enrollment. First, the earlier dropouts of students are predicted, the earlier personalized learning intervention services could be developed to improve the graduation rate [10]. It is shown that online education institutions with the highest graduation rate also attract more students than the average for all higher education [11]. Next, the dropout rate can be used for evaluating the quality of an instructor's course. Predictive dropout rates can inform the optimization of curriculum construction [12], [13], enabling instructors to improve their course quality as early as possible. Finally, predictive dropout rates also provide meaningful information for course recommendations by the platform. By recommending prerequisite courses to students who are predicted to drop out, platforms can improve the student-graduation rate in MOOCs [14], [15].

However, existing research on dropout prediction primarily focuses on students' learning behavior after they have already enrolled in a course. This requires students to first enroll in a course and study for a period [11], [16], [17], which leads to inefficient enrollments and results in a collection of useless data if the student ultimately drops out. Predicting dropout in advance is challenging, and it is also difficult to measure the quality of a course promptly with limited enrollments. Additionally, understanding the learning process of students is just as important as predicting the outcome in education [18], and uncovering explainable evidence behind high dropout rates is crucial, yet often overlooked in current dropout prediction methods.

This paper investigates the feasibility of accurately predicting student dropout before enrollment in a course. To achieve this, the authors propose an Explainable self-supervised Model (ESM) that not only predicts dropout but also provides intermediate results for dropout analysis. Particularly, ESM characterizes a student's learning habits by estimating the probability of actions taken in previous courses. For a completed course, ESM also defines the contribution of each action to the completion. Both the learning habits and action's contribution can be regarded as the probability of taking actions, which are explainable intermediate information (displayed in Fig. 4). And the probability of a student completing a course is defined as the consistency between the learning habits and the action's contribution. In experiments, ESM is evaluated on two benchmark datasets and achieves better performance than existing methods without using any additional learning logs.

Based on the early dropout prediction method, a pre-learning intervention service is designed, where the courses with a high probability of completion to the student are recommended. The result of the simulation experiment on course recommendation shows that introducing early intervention services can significantly improve the completion rate. Most students will select suitable courses and finally complete them. Existing learning intervention strategies, such as sending personalized encouragement emails [10], offering support materials before exams [19], and gradually increasing the difficulty in tests [20], mainly focus on designing intervention during the learning courses, rather than pre-learning intervention service. Therefore, the pre-learning intervention is a novel intervention form in education.

To verify the effectiveness of ESM on dropout prediction and pre-learning intervention, an ablation experiment and case study are designed using two benchmark datasets, XuetangX and KDDCUP. These two datasets record the log data of actions when students actually learn in online platforms, and details of these datasets are shown in Table III. In the task of dropout prediction, results show that ESM improves Area Under the ROC Curve (AUC) and F-score (F1) in two benchmark datasets with 8.3% and 0.6% on average, respectively. Moreover, in the simulation experiment of intervention, the first intervention, course recommendation, improves the 28% completion rate on average in the two platforms. And in the second intervention, course filtration, 40% of wasted time in courses that are predicted to not be completed will be saved in XuetangX. The source code is available on GitHub.¹

The contributions of this paper are summarized:

- To the best of our knowledge, ESM is the first work to address the issue of predicting student dropout before enrollment in MOOCs. By establishing the probability model, the relationship between courses, students, and actions is more explainable. Moreover, the training phase of ESM is self-supervised, which does not require additional annotation of data.
- Two pre-learning intervention services based on the early dropout prediction ESM is proposed. The first intervention is recommending courses to improve the average completion rate of MOOCs platforms. The second intervention is filtering courses to avoid student wasting time in uncompleted courses.

The remainder of this paper is organized as follows. Section II reviews related works on early dropout prediction in online courses. The mathematical formulation of the early dropout prediction task is presented in Section III. The proposed method is introduced in detail, including the explainable analysis, in Section IV. The experimental results demonstrating the performance of the proposed method compared to the State-of-the-Art (SOTA) baseline in the same setting are shown in Section V. A pre-learning intervention service based on the early dropout prediction method is proposed in Section VI, with simulation results that verify its effectiveness for both the platform and students. The paper is concluded in Section VII.

II. RELATED WORK

It is pointed out that MOOCs suffer from a very low completion rate [5], which implies that most students tend to enroll in excessive courses but only complete a few of them. Statistical results show that the probability that a student completes a course is lower than 10% in some MOOCs platforms [4], [21], [22]. One of the main reasons is the low barrier to dropping out from these courses because of the negligible cost of enrollment [10]. In this paper, improving the completion rate is regarded as a two-stage process, dropout prediction, and learning intervention service.

¹[Online]. Available: <https://github.com/SNNU-CmpEdu/ECPM.git>.

A. Dropout Prediction

Dropout prediction is an emerging topic in the field of AI education, which is divided into two categories: (1) Study Session Dropout Prediction (SSDP) and (2) Student Dropout Prediction (SDP) in MOOCs.

MOOCs have a low completion rate, with statistics showing that only a small percentage of students complete the courses they enroll in. Dropout prediction in MOOCs is a growing field of study in AI education, with two main categories: (1) Study Session Dropout Prediction (SSDP) and (2) Student Dropout Prediction (SDP). These categories aim to predict dropouts in MOOCs by analyzing student behavior and other factors.

The SSDP problem was recently defined by Lee et al. [23], which is a task to predict the probability that a learner drops out from his ongoing study session. It is hypothesized that the commonality between the dropout prediction and knowledge tracing tasks would be beneficial to the SSDP task [24]. However, a learner's study session dropout in a course does not mean that the learner eventually drops out of that course. In practice, the course dropout is more important since it can reflect the final result of the learning state. In this section, methods and applications of SDP are mainly discussed.

For the SDP task, the goal is to predict if a student will finish the course at last. Most existing SDP research efforts are focused on SDP after students' enrollment. These existing methods use action or behavior data such as test score data, watching videos, click action, and answering questions generated by students during the learning process after enrolling in courses to establish the dropout prediction model. Prenkaj et al. first classified the existing SDP literature into *Plain Modelisation* and *Sequence Labelling* [11], [25].

- *Plain Modelization*

Plain Modelization refers to the time-invariant nature of the raw log of student activities performed in a MOOC platform, such as the gender, age, and previous GPA of students. Nagrecha et al. took the first step in the direction of incorporating interpretability in MOOC dropout prediction [26]. They selected two interpretable classification methods, decision trees, and logistic regression, to predict the dropout. Another early deep-learning-based dropout prediction model combining the Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) [27]. Feng et al. established a Context-aware Feature Interaction Network (CFIN) model to predict users' dropout in MOOCs by employing a data set of XuetangX [16], which achieves State-of-the-Art performance compared to previous methods.

Some other methods can also be classified into the plain modelization category, but the formulations differ from CFIN. Xing et al. developed a deep learning algorithm using a weekly temporal prediction mechanism to build dropout models [4]. A self-tuning early detection system was built in [28], where personal information, previous education, and current enrollment information are considered. Bonifro et al. exploited machine learning techniques that allow them to predict the dropout of a first-year undergraduate student [29]. An optimization method was structured to modify the initial weight for each training

sample in [17]. Lin et al. designed a double-tower framework [30] to predict whether a student will drop out this semester.

- *Sequence Labelling*

Sequence Labelling only considers time-varying features modeled as time series, for example, the time series of click-stream. Chen et al. proposed a Decision Trees Extreme Learning Machine (DT-ELM), a novel hybrid algorithm combining a decision tree and extreme learning machine [31], to predict who will terminate learning in the next week. The time series was modeled as one-dimensional grid data sampled at fixed time intervals in [32]. Drousiotis et al. established a data optimization method [33] to convert the XuetangX time-series dataset into a discrete-variable dataset, and further predicted the student dropout by using Long Short Term Memory (LSTM), decision trees, and random forests. Wu et al. implemented a deep neural network model consisting of CNN, LSTM, and SVM (CLMS-Net) [34], predicting whether the student will drop out the following day. Mogavi et al. addressed Dropout-Plus (DP) to predict student dropouts and explained the possible reasons why dropouts happened in a real-world platform [35].

There are also a few literature discussions on SDP before students' enrollment. Cheng et al. utilized students' academic data to predict whether these students will drop out in the next semester [36]. However, this work mainly focuses on traditional classes instead of the MOOC environment.

Most previous methods used the data generated after students' enrollment in courses to predict their dropouts. Different from past research, this paper is to predict whether students drop out of the courses that have not been enrolled.

B. Learning Intervention Service

There are different reasons that will lead to the student dropping out from courses in MOOC platform, where the course-related (institutional) factor is one of the main categories of reasons [10]. The institutional factor is highly dependent on the pedagogical approach. The different approach focuses on a different way of learning, e.g., individually learning, learning by experience, and building on prior knowledge [37]. These factors will influence the students' learning experience such as motivation [38], determination [20], curiosity [39], and self-efficacy [40], which further lead to dropout or completion. Therefore, necessary learning intervention services that were designed according to these factors can improve the completion rate in MOOCs.

In [10], several kinds of learning intervention services were introduced to encourage students to complete their courses. The first learning intervention service is sending encouragement emails to students who are learning in the course, which aims to enhance the self-motivation beliefs in an online learning environment [41]. Another learning intervention is offering materials and guidelines before exams, which is considered to improve the student's confidence in their skill. This intervention belongs to the effort on the self-efficacy that is a personal judgment of abilities to meet the challenges [42]. Gradually increasing the difficulty of the course, is also a popular learning intervention that lets the student be more motivated. This is because the

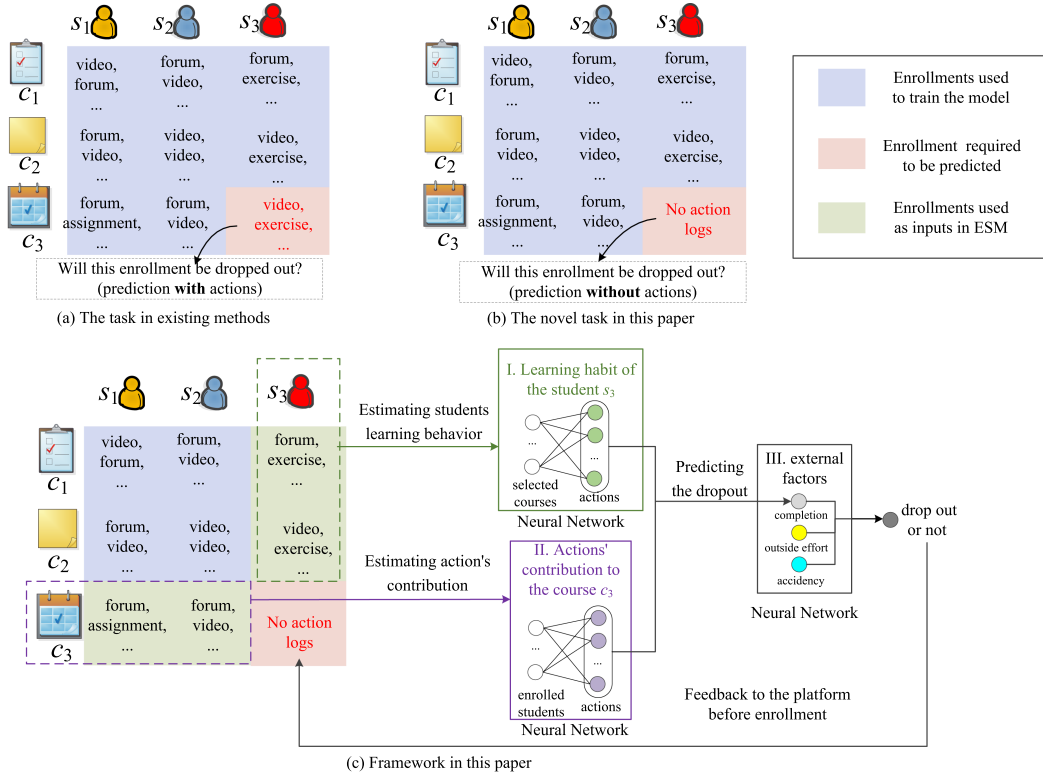


Fig. 1. Overview of this paper. (a) The task in existing methods. (b) The novel task predicting dropout in advance. (c) The framework of ESM.

perceived difficulty of course contents is a known diver of dropout in MOOCs [43], [44]. Besides, gamification in MOOCs is verified to arouse the motivation of learning for students [45], [46], which can be regarded as the learning intervention of optimizing learning contents.

All of the above research concentrates on introducing learning intervention services during the student's study in courses, which requires the collection of logs in the virtual classrooms online. Differently, benefiting from the proposed early dropout prediction method, pre-learning intervention services that are helpful to platforms and students can be considered before enrollment.

III. PROBLEM DEFINITION

This paper aims to explore a novel task in the intelligent education field, whether the dropout can be predicted without using any action logs. The main difference between the existing method and the proposed work is illustrated in Fig. 1. In detail, existing researches aim to train the prediction model using previous learning actions that are shown in black text in Fig. 1(a). In the test phase, test logs of a course are input for prediction, which is shown in red text in Fig. 1(a). In this paper, there are no learning activities in the course that is not enrolled. The mathematical description of the proposed task is represented as follows, which is illustrated in Fig. 1(b).

Specifically, assume a student will select a course from the total K courses randomly. The selected course can be denoted as a random variable \mathcal{C} , where the possible values of \mathcal{C} belong to sample space $C = \{c_1, c_2, \dots, c_K\}$. Here, C is the set of all available courses in a MOOC platform. The student learning in

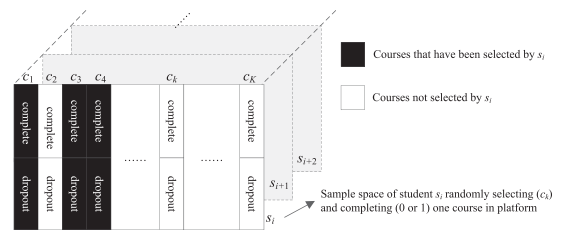


Fig. 2. Sample space E of a student in one sampling process, where each small box denotes an event.

the MOOCs platform is also regarded as a random variable S from $S = \{s_1, s_2, \dots, s_N\}$. Moreover, the possible action that students make is defined as the random variable \mathcal{A} from sample space $A = \{a_1, a_2, \dots, a_M\}$, where each a_m is an allowed action such as play video, check the problem, click forum. Different from existing research, the proposed method aims to predict whether the student s_n will drop out of the course c_k before enrolling in it. Thus, it requires none of the log of action that s_n take in c_k .

To establish a probability model solving the prediction problem, a random variable \mathcal{F}_k is introduced to represent the event that the course c_k will be completed by a student. The possible value of \mathcal{F}_k is 1 or 0, denoting the course c_k is completed or dropped out, respectively. For concise presentation, this paper defines an event $e_k = (\mathcal{C} = c_k) \cap (\mathcal{F}_k = 1)$, meaning a student enrolls in course c_k and completes it. \mathcal{E} is a random variable from the sample space $E = \{e_1, e_2, \dots, e_K\}$. The sample space of \mathcal{E} is displayed in Fig. 2. Notice that the probability of e_k may be very small because there is a potential condition that

TABLE I
IMPORTANT SYMBOLS USED IN THIS PAPER

symbol	description	defined in
C	The set of available courses in the dataset, $C = \{c_1, c_2, \dots, c_K\}$, $K = C $.	3
c_k, c_j	c_k is the k -th course, while c_j can be any course in C .	3
\mathcal{C}	Random variable indicating a student select a course, $\mathcal{C} \in C$.	3
S	The set of all students recorded by the dataset, $S = \{s_1, s_2, \dots, s_N\}$, $N = S $.	3
s_n, s_i	s_n is the n -th student, while s_i can be any student in S .	3
\mathcal{S}	Random variable indicating any student, $\mathcal{S} \in S$.	3
A	The set of available actions students may take during the study, $A = \{a_1, a_2, \dots, a_M\}$, $M = A $.	3
a_m, a_l	a_m is the m -th action, while a_l can be any possible action in A .	3
\mathcal{A}	Random variable indicating which action may appear, $\mathcal{A} \in A$.	3
\mathcal{F}_k	The random variable that if a student complete course c_k , $\mathcal{F}_k \in \{0, 1\}$.	3
e_k	The event $(\mathcal{C} = c_k) \cap (\mathcal{F}_k = 1)$.	3
P	The probability of the event that a student completes a course happens.	3
θ	The set of learnable parameters in ESM, including $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{g}, \mathbf{h}$.	4.2.1
P_θ	The probability of completion revised by trainable parameters θ .	4.2.1
$\mathbf{P}^{(\cdot)}, \mathbf{P}_\theta^{(\cdot)}$	High dimensional tensors storing all probabilities P or P_θ .	4.2.1
$\mathbf{y}, \tilde{\mathbf{y}}$	High dimensional tensors storing the (revised) prediction.	4.2.1
ESM_θ	The abstract representation of the proposed model.	4.3

s_n selects c_k in one sampling process. In practical application, it only care about that whether $P(\mathcal{C} = c_k) \cap (\mathcal{F}_k = 1)$ is larger than $P(\mathcal{C} = c_k) \cap (\mathcal{F}_k = 0)$. For example, for a student, $P(\mathcal{C} = c_k) \cap (\mathcal{F}_k = 1) = 0.09$ and $P(\mathcal{C} = c_k) \cap (\mathcal{F}_k = 0) = 0.01$, this means the student has a very high probability of completion when he/she enrolls in this course.

In this paper, $P(e_k)$ is named as the Completion Probability of the course c_k , which denotes the probability that a student will not drop the course c_k . The goal in this work is to infer $P(\mathcal{F}_k = 1, \mathcal{C} = c_k | \mathcal{S} = s_n)$ for student $s_n \in S$ not selecting course $c_k \in C$, i.e.,

$$P(\mathcal{F}_k = 1, \mathcal{C} = c_k | \mathcal{S} = s_n) = P(\mathcal{E} = e_k | \mathcal{S} = s_n), \quad (1)$$

In the remainder of the paper $P(\mathcal{E} = e_k | \mathcal{S} = s_n)$ is abbreviated as $P(e_k | s_n)$ when there is no confusion.

All symbols used in this paper are summarized in Table I. Generally, italicized non-bold capital letters represent the set, such as C , S , and A . Italic non-bold lowercase letters represent elements in the set, such as c_k , s_n , and a_m . Flourish not-bold capital letters represent the random variables, such as \mathcal{C} , \mathcal{S} , and \mathcal{A} . Non-italic bold lowercase letters represent vectors, such as \mathbf{y} , \mathbf{g} , and \mathbf{h} .

IV. METHODOLOGY

In this section, an explainable conditional probability model, ESM, is established to address the dropout behavior prediction problem. The main framework of ESM is derived in Section IV-A, which includes three modules (shown in Fig. 1(c)), learning habits, actions' contribution, and two external factors. The detailed model structure is displayed in Section IV-B. After that, a self-supervised training strategy is proposed to train the ESM in Section IV-C. Finally, in Section IV-D, an explainability analysis is performed to show the reason that ESM could

predict dropouts before enrollment in courses without using actions.

A. Theoretical Basis

The ESM mainly contains three modules, that are in charge of estimating the learning habit of a student, calculating the importance of actions in a course, and revising the prediction with external factors, respectively.

To estimate the completion probability of student s_n in course c_k , i.e., $P(e_k | s_n)$, $e_k \in E$ in (1), the possible action a_m is first introduced as a hidden variable when using the Law of Total Probability [47],

$$\begin{aligned} P(e_k | s_n) &= \sum_{a_m \in A} P(e_k, a_m | s_n) \\ &= \sum_{a_m \in A} P(a_m | s_n) P(e_k | a_m, s_n). \end{aligned} \quad (2)$$

Here, $P(a_m | s_n)$ is the probability that the student s_n will make action a_m when he/she is learning and $P(e_k | a_m, s_n)$ denotes that the completion probability under the condition that the student s_n makes action a_m . Further, two definitions are extended by these two terms.

Definition 1. Learning habit

Vector $\mathbf{lh} = [P(a_1 | s_n), \dots, P(a_m | s_n), \dots, P(a_M | s_n)]$ is defined as the learning habit, which means *what actions the student prefers to choose*. ■

Definition 2. Actions' contribution for completion

Vector $\mathbf{ac} = [P(e_k | a_1, s_n), \dots, P(e_k | a_m, s_n), \dots, P(e_k | a_M, s_n)]$ is defined as the actions' contribution for completion, indicating the conditional probability that a course will be completed under each action. ■

The computation of the learning habit and the action's contribution will be derived in Sections IV-A1 and IV-A2.

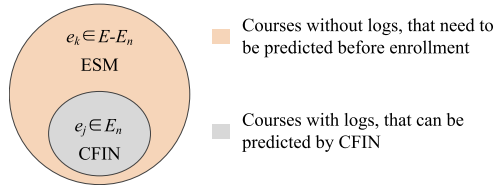


Fig. 3. Difference between e_k and e_j .

1) *Computation of Learning Habit*: In this paper, $P(a_m|s_n)$ is further calculated by

$$\begin{aligned} P(a_m|s_n) &= \sum_{e_j \in E} P(a_m, e_j|s_n) \\ &= \sum_{e_j \in E_n} P(a_m|e_j, s_n)P(e_j|s_n). \end{aligned} \quad (3)$$

The E_n includes all courses that have been selected by the student s_n . The reason why other courses ($e_j \notin E_n$) can be ignored in (3) is represented in the explanation of (4). The relation in Fig. 3 illustrates the difference between $P(e_k|s_n)$ in (2) and $P(e_j|s_n)$ in (3), which also the main difference of ESM from other methods such as CFIN [16]. Two terms in (3) are computed, respectively.

- *Computation of $P(a_m|e_j, s_n)$*

The conditional probability $P(a_m|e_j, s_n)$ can be marginalized as

$$P(a_m|e_j, s_n) = \frac{P(a_m, e_j, s_n)}{\sum_{a_l \in A} P(a_l, e_j, s_n)}. \quad (4)$$

In (4), for the course c_j not selected by student s_n , the joint probability $P(a_m, e_j, s_n)$ is simply set to 0. Meanwhile, the others can be calculated by the frequency that the student s_n makes the action a_m in the course c_j . By marginalizing the random variable \mathcal{A} , the conditional probability of the action a_m is done by a specific student s_n in the course c_j is obtained by (4). ■

- *Computation of $P(e_j|s_n)$*

The conditional probability $P(e_j|s_n)$ is estimated, which is the probability that the student s_n selects and completes the course c_j . Notice that only the student s_n enrolls in c_j should be considered, because $P(a_m|e_j, s_n)$ equals 0 in other cases according to the definition of $P(a_m, e_j, s_n)$. Existing methods aim to predict the dropout behavior for those enrolled courses for the course c_j enrolled by the student s_n , which is generally treated as a binary classification. Therefore, the output score of those existing methods can be used as an estimation of the $P(e_j|s_n)$ in (3), which means the probability of the student s_n completing the course c_j . To evaluate $P(e_j|s_n)$ for $e_j \in E_n$, the CFIN is trained as a binary classification [16]. Different from the original network, the score before binarization is extracted to be the completion probability, i.e., $P(e_j|s_n)$. And the term $P(a_m|s_n)$ in (2) can be inferred via (3)–(4). ■

There is practical meaning in the way of computing $P(a_m|s_n)$ using (3). Specifically, when the student s_n completes the course e_j , which indicates that his or her actions appearing during learning should be important. In contrast, the actions made by

the student s_n in a dropped course should have a low influence on a course's completion. In (3), $P(e_j|s_n)$ can be regarded as the weight to reflect the importance of the actions of student s_n . That is, for those completed courses, the predicted $P(e_j|s_n)$ is close to 1. Meanwhile, $P(e_j|s_n)$ will be close to 0 for the courses where the student s_n has dropped out.

Existing researches point out that learning habit is essential propriety of the student [48], and it is consistent for a student who tends to obtain a completion certificate [49]. Therefore, the learning habit can be estimated by averaging the learning behavior on other courses selected by this student. And it is reasonable to be regarded as a prior for the student s_n learning in a not enrolled course c_k . The illustration of this process is shown in Fig. 4(a).

2) *Computation of Action's Contribution to Completion*: The second term in (2) is $P(e_k|a_m, s_n)$, which can be regarded as the action's contribution of student s_n to the completion, of course, c_k . To estimate this, another random variable \mathcal{S}' is introduced to represent a different student who studies on this platform. Similar to (3), we have

$$\begin{aligned} P(\mathcal{E} = e_k | \mathcal{A} = a_m, \mathcal{S} = s_n) \\ = \sum_{s_i \in \mathcal{S}} P(\mathcal{E} = e_k, \mathcal{S}' = s_i | \mathcal{A} = a_m, \mathcal{S} = s_n). \end{aligned} \quad (5)$$

For the student s_i who does not enroll in the course c_k , the event $(\mathcal{S}' = s_i) \cap (\mathcal{C} = c_k)$ does not happen. Therefore, let $P(\mathcal{E} = e_k, \mathcal{S}' = s_i | \mathcal{A} = a_m, \mathcal{S} = s_n) = 0$ for the case that there is no enrollment pair between s_i and c_k . And then (5) can be simplified as

$$P(e_k|a_m, s_n) = \sum_{s_i \in \mathcal{S}_k} P(e_k, s_i|a_m, s_n), \quad (6)$$

where \mathcal{S}_k is the set of students who have enrolled in the course c_k . Since ESM aims to predict the completion probability before the student s_n selects the course c_k , it has $s_n \notin \mathcal{S}_k$. This means that $s_n \neq s_i$ in (6). Moreover, any user could learn what he or she is interested in MOOCs platform, thus the relationship between students in MOOCs is not as close as that in traditional classes. In this way, two different students enrolling in the same course can be regarded as two independent events. According to the conditional independence, (6) is derived as

$$\begin{aligned} P(e_k|a_m, s_n) &= \sum_{s_i \in \mathcal{S}_k} P(e_k, s_i|a_m, s_n) \\ &= \sum_{s_i \in \mathcal{S}_k} P(e_k, s_i|a_m) \\ &= \sum_{s_i \in \mathcal{S}_k} P(e_k|s_i, a_m)P(s_i|a_m). \end{aligned} \quad (7)$$

Two terms in (7) are computed as follows, respectively.

- *Computation of $P(e_k|s_i, a_m)$*

In (7), $P(e_k|s_i, a_m)$ means the course c_k is completed on the condition that the student s_i has made the action a_m . For the students completing the course c_k , ESM counts the frequency of actions for each student, which is used as the joint probability $P(e_k, s_i, a_m)$, and the conditional probability can be

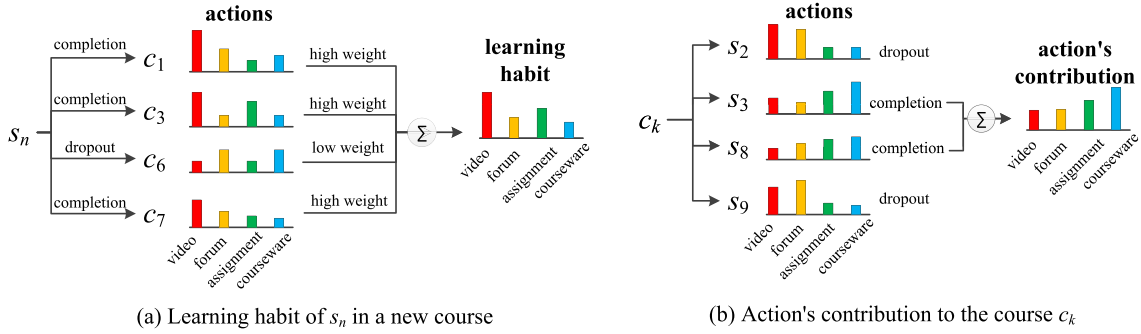


Fig. 4. Illustration of prior information. (a) Learning habit: Assume the student s_n has enrolled in four courses and completed three of them. The weighted average of the frequency of all actions is counted. (b) Action's contribution: Assume four students have enrolled in the course c_k , meanwhile two students completed it finally. The actions only for the completed cases are aggregated, which can avoid the effect of meaningless actions in uncompleted enrollments.

calculated as

$$P(e_k|s_i, a_m) = \frac{P(e_k, s_i, a_m)}{\sum_{e_j \in E} P(e_j, s_i, a_m)}. \quad (8)$$

In practice, (8) can be obtained by normalizing the frequency of actions from the courses completed by the student s_i , who has also completed the course c_k . ■

- *Computation of $P(s_i|a_m)$*

$P(s_i|a_m)$ in (7) means the conditional probability of a student s_i when the action a_m is observed, which can be estimated in a similar way,

$$P(s_i|a_m) = \frac{P(s_i, a_m)}{\sum_{s_l \in S} P(s_l, a_m)}. \quad (9)$$

s_i in (9) can be any student who makes action a_m when he/she studies. Therefore, the time that the student s_i makes the action a_m in all selected courses is counted, and the time of this action made by all students is accumulated. ■

In practice, actions' contribution can reflect *whether an action is useful to learn this course*. This process is illustrated in Fig. 4(b).

3) *Estimation on New Student*: The learning habit cannot be computed directly for the new student who does not enroll in any course before, because E_n is the empty set in (3). For this case, the learning habit can be simply calculated as the average of all existing students, i.e., $P(a_m|s_n) = \frac{1}{N} \sum_{i=1}^N P(a_m|s_i)$. Here, S_i is existing students whose learning habits can be estimated using historical logs. This means if we know nothing about a student, it is reasonable to treat him/her as the most uncharacteristic one. The action's contribution to the completion of a new course can be approximated in a similar way. Moreover, in practice, platforms can deliver a short questionnaire to new registers, which is utilized to estimate the learning habit, for example, new students are asked to select the frequent learning action for their study.

B. Model Structure

In Section IV-A, the relationship among students, courses, and (estimated) actions is represented as the conditional probability model, which is the theoretical basis of ESM. In practical

computation, the data used for the prediction can be regarded as samples of random variables.

Parameterization and Tensorization: In MOOC platforms, the number of enrolled courses is limited, which leads to an insufficient number of samples. Therefore, the predicted results using available samples may differ from the practical results. To solve the problem of limited data, trainable parameters are introduced, which make the revised predicted results close to the practical results. In order to represent parameters more conveniently, the symbols in Section IV-A are represented in the form of tensors.

1) *Common Parameters*: First, the derivation in Section IV-A is extended to the form of tensor, to enable batch processing of data. In order to calculate the action's contribution to completion, $\mathbf{P}^{(A,S,E)} = (P(e_k|s_i, a_m))_{M \times N \times K}$ is stored in a 3-dimensional tensor. Different from e_j , here e_k is the specific course, k th course required to be predicted. To better estimate the contribution, $P(e_k|s_i, a_m)$ is corrected by trainable parameters $\theta = [\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{g}, \mathbf{h}]$

$$\mathbf{P}_\theta^{(A,S,E)} = \mathbf{P}^{(A,S,E)} \odot \mathbf{W}_1 + \mathbf{b}_1. \quad (10)$$

In (10), \mathbf{W}_1 , and \mathbf{b}_1 are tensors with the same dimension as $\mathbf{P}^{(A,S,E)}$, where \mathbf{W}_1 and \mathbf{b}_1 belong to set θ . ' \odot ' is the pairwise multiplication operation. $\mathbf{P}_\theta^{(A,S,E)} = (P_\theta(e_k|s_i, a_m))_{M \times N \times K}$ is revised by parameters in \mathbf{W}_1 and \mathbf{b}_1 in θ .

Similarly, $\mathbf{P}^{(A,S)} = (P(s_i|a_m))_{M \times N \times 1}$ includes all students and actions in (9), which is corrected by

$$\mathbf{P}_\theta^{(A,S)} = (\mathbf{P}^{(A,S)} \odot \mathbf{W}_2 + \mathbf{b}_2) \otimes \mathbf{W}_3, \quad (11)$$

where ' \otimes ' represents the tensor multiplication operation. \mathbf{W}_2 and \mathbf{b}_2 have the same size with $\mathbf{P}^{(A,S)}$. $\mathbf{W}_3 \in \mathcal{R}^{M \times M \times 1}$, which can be multiplied with $\mathbf{P}^{(A,S)}$. Tensor $\mathbf{P}_\theta^{(A,S)} = (P_\theta(s_i|a_m))_{M \times N \times 1}$ the revision based on θ .

Combining the output in both (10) and (11), the actions' contribution to completion can be calculated as

$$\mathbf{P}_\theta^{(A,E)} = \mathbf{P}_\theta^{(A,S)} \otimes \mathbf{P}_\theta^{(A,S,E)}. \quad (12)$$

To calculate the learning habit of student s_n , the log data that student s_n learns in all courses (0 for an unenrolled

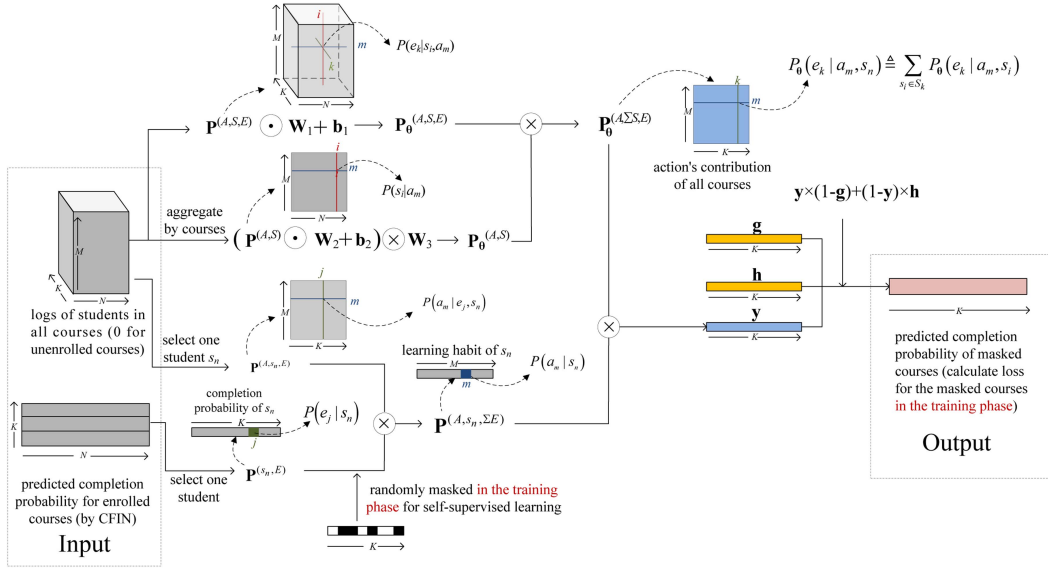


Fig. 5. Detailed structure of ESM.

 TABLE II
 DETAILS OF ARCHITECTURE

	XuetangX	KDDCUP		XuetangX	KDDCUP
\mathbf{W}_1	9441×247	10451×39	\mathbf{B}_1	9441×247	10451×39
\mathbf{W}_2	9441×19	10451×3	\mathbf{B}_2	9441×19	10451×3
\mathbf{W}_3	19×19	3×3	\mathbf{g}	1×247	1×39
\mathbf{h}	1×247	1×39			

course) is extracted and stored in tensor $\mathbf{P}^{(A,s_n,E)} = (P(a_m | e_j, s_n))_{M \times 1 \times K}$. Here, s_n is fixed, and e_j represents any course in K courses. $\mathbf{P}^{(s_n,E)} = (P(e_j | s_n))_{1 \times 1 \times K}$ can be estimated by CFIN [16], because it contains logs of enrollments. And then, (3) can be extended as the form of tensor multiplication

$$\mathbf{P}^{(A,s_n)} = \mathbf{P}^{(A,s_n,E)} \otimes \mathbf{P}^{(s_n,E)}, \quad (13)$$

$\mathbf{P}^{(A,s_n)} = (P(a_m | s_n))_{M \times 1 \times 1}$ is the learning habit.

Finally, the prediction of all courses for student s_n is estimated in batches

$$\mathbf{y} = \mathbf{P}_\theta^{(A,E)} \otimes \mathbf{P}^{(A,s_n)}. \quad (14)$$

In (14) $\mathbf{y} = (P_\theta(e_k | s_n))_{1 \times 1 \times K}$ includes K completion probabilities for K courses. In practice, the size of parameters is depended on the logs recorded in platforms, which is shown in Table II.

2) *External Factors*: In this part, two additional parameters are considered to revise the estimated completion probability.

Dropout by Accident: There are some unexpected reasons leading to students' dropout behavior. For example, students work hard in a course, however, the platform judges them to be uncompleted because of some special reasons. Then his or her actions in this class tend to refer to dropout behavior. However, these actions should have been associated with completion.

Outside Effort: Some students may complete courses by factors outside the MOOCs. For instance, to learn courses in

computer sciences, students have to spend a lot of time coding after class. Since these actions cannot be recorded by platforms, students' learning actions in these courses have less relationship to their completion.

For all courses, c_k , parameter g_k is utilized to indicate the factor that a student drops the course c_k by accident. Similarly, h_k is introduced to denote the factor that a student completes the course c_k via outside effort. Let $\mathbf{g} = [g_1, \dots, g_K]$ and $\mathbf{h} = [h_1, \dots, h_K]$, the predicted completion probability is revised as

$$\tilde{\mathbf{y}} = \mathbf{y} \odot (1 - \mathbf{g}) + (1 - \mathbf{y}) \odot \mathbf{h}. \quad (15)$$

In (15), $\mathbf{y} \odot (1 - \mathbf{g})$ means the student s_n could have completed all courses by probability $P(e_1 | s_n), \dots, P(e_K | s_n)$, but finally drops out actually with a certain probability $\mathbf{g} = [g_1, \dots, g_K]$. Similarly, $(1 - \mathbf{y}) \odot \mathbf{h}$ denotes that s_n is predicted to drop out of courses as the probability $1 - P(e_1 | s_n), \dots, 1 - P(e_K | s_n)$, however, complete by the outside effort that happens with the certain probability $[h_1, \dots, h_K]$.

To learn the correspondence between students, courses, and actions from recorded logs in the training dataset, the neural network is introduced in the ESM. The detailed structure is shown in Fig. 5.

C. Self-Supervised Training Strategy

To train parameters in the ESM, a self-supervised training strategy is adopted. In particular, a specific student s_n is selected, and then the completion probability of the courses that have log data is predicted by CFIN, i.e., $\mathbf{P}^{(s_n,E)}$. Here, the completion probability of unenrolled courses, $P(e_k | s_n), e_k \notin E_n$ (in Fig. 3), are set to 0. In each training step, $\mathbf{P}^{(s_n,E)}$ is the input of ESM, meanwhile, output $\tilde{\mathbf{y}}$, represents the prediction for all courses considering the correlation between students, courses, and actions. For brevity, the relation between input and output

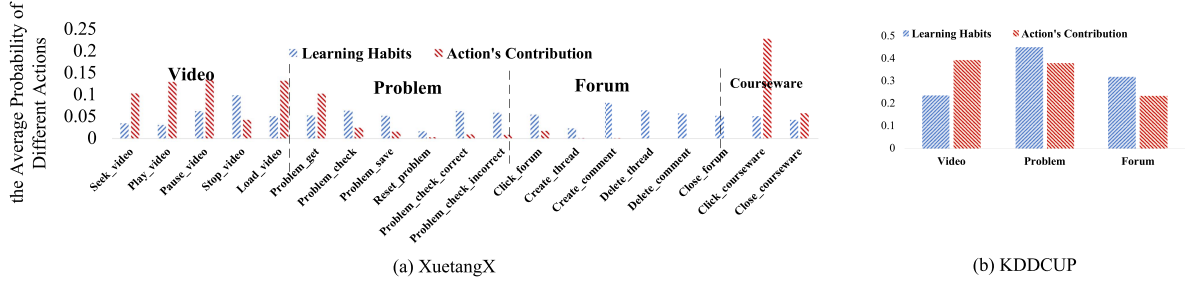


Fig. 6. Visualization of the learning habit and action's contribution. (a) XuetangX, cosine similarity is 0.57 (b) KDDCUP, cosine similarity is 0.94. Since KDDCUP only records 3 kinds of actions, the similarity is close to 1.

in the training phase is defined as

$$\tilde{\mathbf{y}} = \text{ESM}_{\theta} \left(\mathbf{P}^{(s_n, E)} \right), \quad (16)$$

where 'ESM_θ(·)' is regarded as a function including operation defined in (13)–(15).

To train parameters θ in ESM_θ for correctly predicting the unenrolled courses, a number of courses when $P(e_j|s_n) > 0$ ($e_j \in E_k$ that can be predicted by CFIN), is randomly selected and set to 0. This process can be implemented by multiplying $\mathbf{P}^{(s_n, E)}$ with a mask vector. It means that the information from these selected courses will not be used in prediction, in contrast, should be predicted by ESM in this training step. The closer between $\mathbf{P}^{(s_n, E)}$ and $\tilde{\mathbf{y}}$, the better performance of ESM in the prediction task. The loss of each iteration is designed as

$$\text{loss} = \left\| \left(\text{ESM}_{\theta} \left(\mathbf{P}^{(s_n, E)} \odot \mathbf{P}^{\text{mask}} \right) - \mathbf{P}^{(s_n, E)} \right) \odot (1 - \mathbf{P}^{\text{mask}}) \right\|_F^2. \quad (17)$$

Here $\mathbf{P}^{\text{mask}} \in \{0, 1\}^K$ indicates the randomly generated mask vector. '⊙' is the pairwise multiplication. The loss between the predicted result $\text{ESM}_{\theta}(\mathbf{P}^{(s_n, E)} - \mathbf{P}^{\text{mask}})$ and the real probability $\mathbf{P}^{(s_n, E)}$ will guide the optimization of model parameters, and only the masked elements are considered in the loss, i.e., $(1 - \mathbf{P}^{\text{mask}})$.

To train all parameters θ in ESM, students in the training set are iteratively chosen in a completed training epoch, and the loss in (17) is minimized by randomly generating \mathbf{P}^{mask} . After several epochs, the parameters will gradually converge. The training phase does not require extra annotation, which is the self-supervised training strategy. The detailed implementation is shown in Algorithm 1.

D. Explainability Analysis

The main idea of ESM can be represented as follows. Although there is no log of actions available for a student s_n making in one course c_k , it could infer completion probability by two kinds of prior information, learning habit (*Definition 1*) of the student s_n and the actions' contribution (*Definition 2*) of the course c_k . The first term means the probability that s_n will take action in any new course. Meanwhile, the second term is how these actions will influence course completion c_k . In this way, the logs of action that student s_n takes in course c_k are

Algorithm 1: Self-Supervised Training.

Input: log of student's study $\mathbf{P}^{(A, S, E)}$, training set $S_{tr} \times C_{tr}$, learning rate η , number of epoch T

Output: parameters θ in ESM_θ

- 1: Initialize $\theta = [\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2, \mathbf{W}_3, \mathbf{g}, \mathbf{h}]$;
- 2: Set elements in $\mathbf{P}^{(A, S, E)}$ to 0, for $\forall (s_i, c_j) \notin S_{tr} \times C_{tr}$;
- 3: $\mathbf{P}^{(A, S, E)} \leftarrow$ Normalize $\mathbf{P}^{(A, S, E)}$ from action dimension;
- 4: Calculate $\mathbf{P}_{\theta}^{(A, S, E)}$ via (10);
- 5: $\mathbf{P}^{(A, S)} \leftarrow$ Aggregate $\mathbf{P}^{(A, S, E)}$ from course dimension;
- 6: Calculate $\mathbf{P}_{\theta}^{(A, S)}$ via (11);
- 7: Calculate $\mathbf{P}_{\theta}^{(A, E)}$ via (12);
- 8: $\mathbf{P}^{(S, E)} \leftarrow$ Predict completion probability by CFIN;
- 9: **for** t in 1 to T **do**
- 10: **for** n in 1 to $|S|$ **do**
- 11: Select a tensor $\mathbf{P}^{(A, s_n, E)} \in \mathcal{R}^{M \times 1 \times K}$ from $\mathbf{P}^{(A, S, E)}$;
- 12: Select a tensor $\mathbf{P}^{(s_n, E)} \in \mathcal{R}^{1 \times 1 \times K}$ from $\mathbf{P}^{(S, E)}$;
- 13: Randomly generate \mathbf{P}^{mask} in $\{0, 1\}^{1 \times 1 \times K}$;
- 14: $\mathbf{P}^{(s_n, E)} \leftarrow \mathbf{P}^{(s_n, E)} \odot \mathbf{P}^{\text{mask}}$;
- 15: Calculate $\mathbf{P}^{(A, s_n)}$ via (13);
- 16: Calculate $\tilde{\mathbf{y}}$ via (14);
- 17: $\text{loss} \leftarrow \left\| (\tilde{\mathbf{y}} - \mathbf{P}^{(s_n, E)}) \odot (1 - \mathbf{P}^{\text{mask}}) \right\|_F^2$;
- 18: Calculate gradient $\nabla \theta$ according to loss ;
- 19: $\theta \leftarrow \theta + \eta \nabla \theta$;
- 20: **end for**
- 21: **end for**

not necessary for ESM. Therefore, it can predict the completion probability before students' enrollment.

To quantize the relationship between learning habits and actions' contribution, the similarity between these two vectors, \mathbf{lh} , and \mathbf{ac} , can be calculated, such as cosine similarity. A greater similarity indicates that the student is more likely to complete the course. In Fig. 6(a) and (b) in Section V-C, statistical results show the most comment action of students, as well as the most important action of courses. The lower similarity can partly explain why platforms have lower course completion ratios.

TABLE III
DETAILS OF TWO BENCHMARK DATASETS

dataset	students	courses	enrollments	actions*
XuetangX	254,518	698	467,113	Click forum, Play video, ...
KDDCUP	112,448	39	200,904	Video, Forum, ...

* all possible actions are listed in Fig. 6.

V. EXPERIMENTS

In experiments, the effectiveness and explainability of ESM are evaluated on two benchmark datasets. Since this paper first essays to infer dropouts in advance, ESM is compared with State-Of-The-Art baselines with the same settings.

A. Datasets, Settings, and Evaluation Metrics

For the action-based dropouts prediction, there are two large-scale public and real datasets, XuetangX² and KDDCUP³, shown in Table III. In XuetangX, 4 actions including 19 fine-grained actions are provided, for example, load video, check the problem, click courseware, create comments, et al. In KDDCUP, 3 actions are recorded: video, problem, and forum. After students enroll in courses, MOOC platforms create logs for these actions. In the self-supervised training phase, students who have enrolled in at least 5 courses are selected for evaluation, considering the trade-off between the number of available students and the diversity of input. In contrast, the test phase does not require a student to learn 5 courses before. Moreover, if a new student has not enrolled in any courses, the learning habit could be approximately estimated by the method in Section IV-A3. The practical number of samples in the training and test set is about 8:2 in the experiments, which is a popular setting in pattern recognition research [50]. In the training phase, the logs of all students in the training set are utilized to train the parameters. And in the test phase, the historical logs of one student are inputted to predict the completion probability on other unenrolled courses.

The main baseline for comparison is CFIN [16], where the problem settings and the evaluated datasets are similar to ours. In CFIN, students are required to learn courses for a period. CFIN is also implemented with different lengths of the learning period, varying from 1 d to several weeks. In this paper, ESM equals to use the 0-day action log.

The proposed method in this paper is implemented by Pytorch 1.10.0. The parameters are optimized using Adam, where the learning rate is 10^{-3} and the number of training epochs is 25. All experiments are run in the environment of Intel Core I7-7820× CPU@3.60GHz× and GPU@2070.

In the main experiment comparing the accuracy of the early prediction task, the evaluation metrics include F-score (F1) and Area Under the ROC Curve (AUC).

To calculate the F1, four variables are first defined. True Positive (TP), the number of completed course predicted to be completed (correct prediction of completion). True Negative

TABLE IV
COMPARISON OF DIFFERENT METHOD

Methods	KDDCUP		XuetangX	
	AUC(%)	F1(%)	AUC(%)	F1(%)
LRC	86.78	90.86	82.23	89.35
SVM	88.56	91.65	82.86	89.78
RF	88.82	91.73	83.11	89.96
DNN	88.94	91.81	85.64	90.40
GBDT	89.12	91.88	85.18	90.48
CFIN-7w	90.93	92.87	86.71	90.95
CFIN-6d	72.67	88.23	65.72	86.33
CFIN-5d	71.51	88.13	65.57	86.35
CFIN-4d	70.29	87.97	64.67	86.21
CFIN-3d	69.29	87.85	63.62	86.00
CFIN-2d	67.17	87.62	62.52	85.90
CFIN-1d	67.24	87.18	62.24	85.62
CFIN-0d	55.64	86.60	60.71	87.28
ESM	66.00	87.07	67.00	87.99

The bold values indicate the best performance.

(TN), the number of dropout course predicted to be dropout (correct prediction of dropout). False Positive (FP), the number of dropout course predicted to be complete (wrong prediction of completion). False Negative (FN), the number of completed predicted to be dropout (wrong prediction of dropout). And then, the precision and recall can be calculated, respectively,

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}. \quad (18)$$

The F1 is the balance of precision and recall, which is computed as

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \quad (19)$$

AUC reflects the performance for the classification problems at various threshold settings, which can be defined as

$$AUC = \frac{\sum I(p, n)}{(|TP| + |FN|) \times (|FP| + |TN|)}, \quad (20)$$

where $I(p, n) = 1$ for the pair of positive sample p and negative sample n satisfying the predict score of p higher than n .

B. Main Results

The dropout prediction task can be regarded as a binary classification problem. In this section, the F1 and AUC measurements are used for evaluating the accuracy of different methods. The comparison of performance on two datasets is shown in Table IV.

Clearly, the first term shows the results that require students to enroll in courses and study for several weeks, which includes the main comparison reported in [16]. Here, CFIN-7 w indicates the performance of the CFIN model using logs during the first 7 weeks after enrollment. The second term in Table IV lists the results of CFIN using different lengths of logs, which can be the reference for evaluating the performance of ESM. Similarly, CFIN- x d means the results of the CFIN model using x days logs, which is named short-period CFIN in the following discussion. The third term represents the accuracy of results that is estimated

²[Online]. Available: <http://moocdata.cn/data/user-activity>

³[Online]. Available: <https://data-mining.philippe-fourmier-viger.com/the-kddcup-2015-dataset-download-link/>

without any actions. The CFIN with 0-day logs and ESM are compared.

From Table IV following conclusions can be derived. First, the performance of the method improves as the duration of logs increases, since results in the first term are better overall than that in the second term. In addition, ESM achieves comparable performance with short-period CFIN in two datasets. Finally, ESM is better than CFIN-0 d in most cases, where no action logs are used. Since students' profile is also available in CFIN, the performance of CFIN-0 d is better than a random guess. In summary, the main results show the effectiveness of ESM, which provides a new idea for predicting dropout before enrollment.

Moreover, ESM has comparable results with CFIN-1 d. However, CFIN-1 d still requires students to enroll in a course and learn for a short period. For example, if the platform contains 100 courses, then CFIN-1 d needs students to register for all these courses before prediction. Instead, ESM requires students to finish several courses (or fulfill some questionnaire for estimating their learning habits), and then predict the completion probability for all courses directly.

C. Explainability Evaluation

In the proposed method, the result of each step is explainable by conditional probability. Therefore, it could explore many details from the intermediate results of the model, which is helpful to analyze how actions in the courses influence the completion. The learning habit of all students is estimated, i.e., accumulating the most frequent actions that students take,

$$\frac{1}{|S|} \sum_{s_n \in S} \mathbf{I}(P(a_m | s_n)), \quad (21)$$

where $\mathbf{I}(\mathbf{z})$ indicates the one-hot encode of a vector \mathbf{z} , i.e., only the maximum elements in \mathbf{z} is set to 1 and the others are set to 0.

Similarly, the importance of actions for the completed courses is estimated. Considering that the number of courses is much smaller than that of students, the accumulation of one-hot encode will be not stable. Therefore, the average of the action's contribution in all courses is calculated,

$$\frac{1}{|C|} \sum_{c_k \in C} P(e_k | a_m, s_n). \quad (22)$$

The statistical results of the learning habit and action's contribution of XuetangX and KDDCUP are visualized in Fig. 6(a) and (b), respectively. It illustrates that students tend to study from videos and courseware in XuetangX. In KDDCUP, the category of recorded action is not so fine-grained, there is no significant difference in the proportion of students performing various actions. In contrast, in both XuetangX and KDDCUP, the problem-like and forum-like actions make the most contribution to the course completion according to ESM. The visualization of the learning habit and action's contribution shows that the most frequently appeared actions are different from the most important actions. That means what students tend to do makes less help to the course completion, which can be evidence of the high dropout rate in MOOCs. This finding is useful to MOOC

platforms in optimizing the learning resources in courses, which may guide students to do actions that are more useful to course completion.

D. Ablation Experiment of External Factors

More importantly, in the contribution of two external factors in ESM, the probability of dropout by accident and completion by outside effort, are embedded into parameters \mathbf{g} and \mathbf{h} . In Table V, the comparison with and without these factors is displayed. When these factors are considered, the performance increases by about 5%-6% in AUC and 1%-3% in F1. This improvement verifies the effectiveness of external factors discussed in the Methodology. Moreover, the comparison of time consumption shows that the extra computing time is negligible compared to the time fluctuations when the program is running.

VI. PRE-LEARNING INTERVENTION SERVICE

In this part, a case study about pre-learning intervention services is proposed based on the dropout prediction before enrollment. Since students are not required to enroll in and take any actions in courses, the proposed pre-learning intervention service is not implemented during students' learning process. Technically, the proper courses for students are recommended to improve the average completion rate in the platform. Meanwhile, courses that are predicted to be a high dropout probability will be filtered out, which can avoid the waste of time for students. Different from the normal action-level intervention service like CFIN [16], the proposed strategy of intervention service depends on early dropout prediction.

In this section, the public datasets are used for simulation. This means all data is recorded logs. The datasets are divided into the training set and the test set. Data in the training set is used for training the parameters in the ESM. Since the test set is completely unknown during the training phase, it can be considered as the upcoming data in the simulation experiment.

To evaluate the effectiveness of the strategy, the completion rate before and after the pre-learning intervention service is compared. Specifically, the students and their enrolled courses in the test set are selected, and then statistic the average completion rate, which is regarded as the learning status without any learning intervention service. Assume that the set of all enrolled courses in the test phase is indicated as C_{test} , where the set of completed courses is denoted as $C_{complete}$. The original completion rate R_{before} is computed as

$$R_{before} = \frac{|C_{complete}|}{|C_{test}|}. \quad (23)$$

Pre-Learning intervention service strategy 1 - course recommendation: After the dropout before students' enrollment is predicted, r courses for each student with the r highest probability of completion are recommended. The set of recommended courses is represented as $C_{recommend}$, and then the completion rate after the pre-learning intervention service is

$$R_{after} = \frac{|C_{complete} \cap C_{recommend}|}{|C_{test} \cap C_{recommend}|}. \quad (24)$$

TABLE V
INFLUENCE OF THE EXTERNAL FACTORS

External factors	KDDCUP				XuetangX			
	AUC	F1	Train time/ an epoch	test time/ a student	AUC	F1	Train time/ an epoch	test time/ a student
with	66.00%	87.07%	29s	5.67ms	67.00%	87.99%	34s	4.08ms
without	59.57%	84.18%	30s	5.86ms	62.74%	86.81%	34s	3.93ms

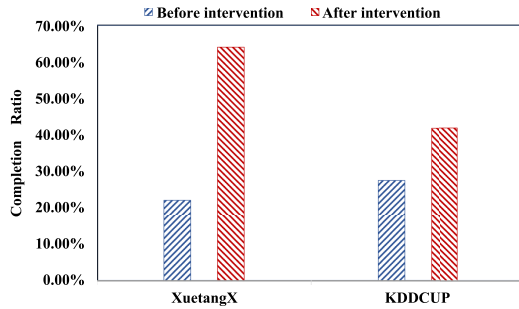


Fig. 7. Comparison of the average completion rate with and without pre-learning intervention service in different datasets.

The improvement of the completion rate with the pre-learning intervention service is shown in Fig. 7, where r is set to 10. From the comparison, it is shown that only 22% and 27% enrolled courses are completed in XuetangX and KDDCUP, respectively. When personalized courses are recommended to different students as a service of optimizing learning content, the completion rate of XuetangX and KDDCUP achieves 60% and 45%, respectively. Therefore, both platforms and students will significantly benefit from the proposed prediction and intervention service. Notice the number of courses r reflects the strength of intervention service in practice. If $r = |C|$, it means there is no intervention service because all courses are available to all students. In contrast, the smaller r is, the greater the level of intervention service. In practice, it can be considered by the platforms. For example, it should give strong intervention services for students who have a low completion rate.

Besides, the number of students who benefit from the pre-learning intervention service is recorded. Here, the percentage of the improved completion rate of each student is calculated, and then a histogram of the number of students with different degrees of improvement is drawn. Since this experiment counts the completion rate for each student, the student who has completed at least one course is considered, where the result is displayed in Fig. 8. It is shown that more than 500 students in XuetangX improve at least 40% completion rate after the pre-learning intervention service. In KDDCUP, more than 1200 students improve their 20%-40% completion rate when they are recommended proper courses. Therefore, most students will benefit from the pre-learning intervention service, which is supported by the proposed early dropout prediction method.

Pre-Learning intervention service strategy 2 - course filtration: When it is intervened by recommending personalized courses for each student, it aims to improve the average completion rate in platforms. In contrast, *it could also remove courses that students have a high probability to give up.* In this way, the benefit to students of filtering the courses with low

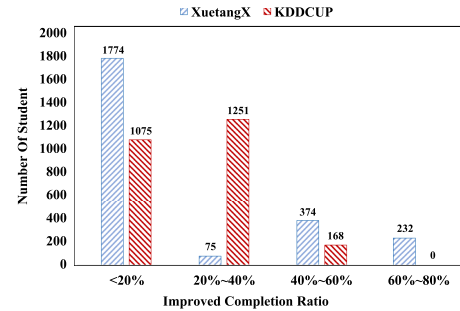


Fig. 8. Histogram of the number of students with different degrees of improvement in the completion rate.

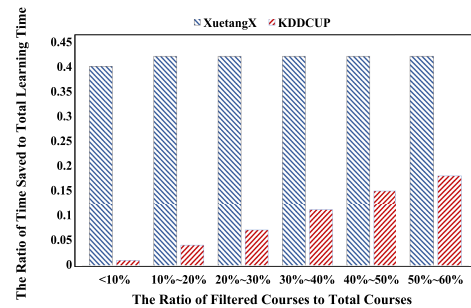


Fig. 9. Average of hours saved with variate number of courses filtered.

completion probability is discussed. Specifically, the courses that are predicted to be easily dropped out are selected, and the student who enrolled in and dropped them is counted. This result reflects how much time a student could save when the pre-learning intervention service is introduced. In the simulation experiment, the duration of students' actions is accumulated, as the time students spent in specific courses. Then, we remove variable rates of the number of courses that are selected by students, and statistics the rate of time students saved when uncompleted courses are filtered. From Fig. 9, it is shown that students studying in XuetangX can save about 40% time when 20% courses predicted high dropout probability are filtered. For students in KDDCUP, when 60% courses are removed, 15% time that students waste in their uncompleted course will be saved.

From the above experimental results, it is concluded that the proposed prediction method can support pre-learning intervention services such as course recommendations and filtration, which can significantly improve the efficiency of both students and platforms.

VII. CONCLUSION

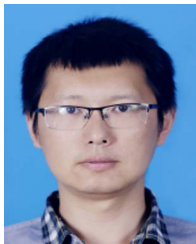
In this paper, an explainable probability method is proposed, which can predict dropouts without using any enrollment or

learning logs. To predict the dropout of a student from a not enrolled course, the actions he or she may take are introduced as the hidden variables. First, the conditional probability of the action taken by this student is predicted by considering his or her previous learning habit. Second, the probability that this course will be completed on the condition that each action appears is estimated. Finally, the completion probability is calculated by aggregating the possible action, where the contribution of learning action on course completion is regarded as the weight. Experimental results show that ESM achieves comparable performance to the action-based method using short-period logs of learning action. More important, pre-learning intervention services based on ESM are designed, which is helpful to improve the completion rate for MOOCs.

REFERENCES

- [1] C. Tekin, S. Elahi, and M. Van DerSchaar, "Feedback adaptive learning for medical and educational application recommendation," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 2144–2157, Jul./Aug. 2022.
- [2] G. Sun, T. Cui, J. Yong, J. Shen, and S. Chen, "MLaaS: A cloud-based system for delivering adaptive micro learning in mobile MOOC learning," *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 292–305, Mar./Apr. 2018.
- [3] J. A. González-Martínez, M. L. Bote-Lorenzo, E. Gómez-Sánchez, and R. Cano-Parra, "Cloud computing and education: A state-of-the-art survey," *Comput. Educ.*, vol. 80, pp. 132–151, 2015.
- [4] W. Xing and D. Du, "Dropout prediction in MOOCs: Using deep learning for personalized intervention," *J. Educ. Comput. Res.*, vol. 57, no. 3, pp. 547–570, 2019.
- [5] J. He, J. Bailey, B. Rubinstein, and R. Zhang, "Identifying at-risk students in massive open online courses," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1749–1755.
- [6] H. Khalil and M. Ebner, "MOOCs completion rates and possible methods to improve retention—A literature review," in *Proc. EdMedia + Innovate Learn.*, Tampere, Finland, Jun. 2014, pp. 1305–1313.
- [7] D. Clow, "MOOCs and the funnel of participation," in *Proc. 3rd Int. Conf. Learn. Analytics Knowl.*, 2013, pp. 185–189.
- [8] K. Jordan, "Massive open online course completion rates revisited: Assessment, length and attrition," *Int. Rev. Res. Open Distrib. Learn.*, vol. 16, no. 3, pp. 341–358, 2015.
- [9] D. F. Onah, J. Sinclair, and R. Boyatt, "Dropout rates of massive open online courses: Behavioural patterns," in *Proc. 6th Int. Conf. Educ. New Learn. Technol.*, 2014, pp. 5825–5834.
- [10] I. Borrella, S. Caballero-Caballero, and E. Ponce-Cueto, "Taking action to reduce dropout in MOOCs: Tested interventions," *Comput. Educ.*, vol. 179, Art. no. 104412, 2022.
- [11] B. Prenkaj et al., "A survey of machine learning approaches for student dropout prediction in online courses," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 57:1–57:34, 2020.
- [12] Z. Xiong and L. Zhi, "Research on optimizing online course construction mode," in *Proc. IEEE 14th Int. Conf. Comput. Sci. Educ.*, 2019, pp. 370–373.
- [13] D. Wei, S. Zhong, Z. He, and J. Tang, "Construction and practice of curriculum pre-warning model based on data analysis," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Educ.*, 2021, pp. 286–291.
- [14] E. Shao, S. Guo, and Z. A. Pardos, "Degree planning with PLAN-BERT: Multi-semester recommendation using future courses of interest," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 14920–14929.
- [15] C. Wang et al., "Personalized and explainable employee training course recommendations: A Bayesian variational approach," *ACM Trans. Inf. Syst.*, vol. 40, no. 4, pp. 70:1–70:32, 2022.
- [16] W. Feng, J. Tang, and T. X. Liu, "Understanding dropouts in MOOCs," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 517–524.
- [17] C. Jin, "Dropout prediction model in MOOC based on clickstream data and student sample weight," *Soft Comput.*, vol. 25, no. 14, pp. 8971–8988, 2021.
- [18] P. Guo, N. Saab, L. S. Post, and W. Admiraal, "A review of project-based learning in higher education: Student outcomes and measures," *Int. J. Educ. Res.*, vol. 102, 2020, Art. no. 101586.
- [19] K. Bartimote-Aufflick, A. Bridgeman, R. Walker, M. Sharma, and L. Smith, "The study, evaluation, and improvement of university student self-efficacy," *Stud. Higher Educ.*, vol. 41, no. 11, pp. 1918–1942, 2016.
- [20] T. Eriksson, T. Adawi, and C. Stöhr, "'Time is the bottleneck': A qualitative study exploring why learners drop out of MOOCs," *J. Comput. Higher Educ.*, vol. 29, no. 1, pp. 133–146, 2017.
- [21] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard, "Who does what in a massive open online course?," *Commun. ACM*, vol. 57, no. 4, pp. 58–65, 2014.
- [22] R. F. Kizilcec, C. Piech, and E. Schneider, "Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses," in *Proc. 3rd Conf. Learn. Analytics Knowl.*, 2013, pp. 170–179.
- [23] Y. Lee et al., "Deep attentive study session dropout prediction in mobile learning environment," in *Proc. 12th Int. Conf. Comput. Supported Educ.*, 2020, pp. 26–35.
- [24] S. Lee, K. S. Kim, J. Shin, and J. Park, "Tracing knowledge for tracing dropouts: Multi-task training for study session dropout prediction," in *Proc. 14th Int. Conf. Educ. Data Mining*, 2021, pp. 641–646.
- [25] B. Prenkaj, G. Stilo, and L. Madeddu, "Challenges and solutions to the student dropout prediction problem in online courses," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 3513–3514.
- [26] S. Nagrecha, J. Z. Dillon, and N. V. Chawla, "MOOC dropout prediction: Lessons learned from making pipelines interpretable," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 351–359.
- [27] W. Wang, H. Yu, and C. Miao, "Deep model for dropout prediction in MOOCs," in *Proc. 2nd Int. Conf. Crowd Sci. Eng.*, 2017, pp. 26–32.
- [28] J. Berens, K. Schneider, S. Gortz, S. Oster, and J. Burghoff, "Early detection of students at risk - predicting student dropouts using administrative student data from German universities and machine learning methods," *J. Educ. Data Mining*, vol. 11, no. 3, pp. 1–41, 2019.
- [29] F. D. Bonifro, M. Gabbriellini, G. Lisanti, and S. P. Zingaro, "Student dropout prediction," in *Proc. Artif. Intell. Educ. 21st Int. Conf.*, 2020, pp. 129–140.
- [30] J. Lin et al., "MOOC student dropout rate prediction via separating and conquering micro and macro information," in *Proc. 28th Int. Conf. Neural Inf. Process.*, 2021, pp. 459–467.
- [31] J. Chen, J. Feng, X. Sun, N. Wu, Z. Yang, and S. Chen, "MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine," *Math. Problems Eng.*, vol. 2019, no. 3, pp. 1–11, 2019.
- [32] L. Qiu, Y. Liu, Q. Hu, and Y. Liu, "Student dropout prediction in massive open online courses by convolutional neural networks," *Soft Comput.*, vol. 23, no. 20, pp. 10287–10301, 2019.
- [33] E. Drousiotis, P. Pentaliotis, L. Shi, and A. I. Cristea, "Capturing fairness and uncertainty in student dropout prediction - A comparison study," in *Proc. 22nd Int. Conf. Artif. Intell. Educ.*, 2021, pp. 139–144.
- [34] N. Wu, L. Zhang, Y. Gao, M. Zhang, X. Sun, and J. Feng, "CLMS-Net: Dropout prediction in MOOCs with deep learning," in *Proc. ACM Turing Celebration Conf.*, 2019, pp. 75:1–75:6.
- [35] R. H. Mogavi, X. Ma, and P. Hui, "Characterizing student engagement moods for dropout prediction in question pool websites," in *Proc. ACM Hum. Comput. Interact.*, vol. 5, pp. 1–22, 2021.
- [36] Y. Cheng, B. P. Nunes, and R. Manrique, "Not another hardcoded solution to the student dropout prediction problem: A novel approach using genetic algorithms for feature selection," in *Proc. 18th Int. Conf. Intell. Tutoring Syst.*, 2022, pp. 238–251.
- [37] G. Conole, "Designing effective MOOCs," *Educ. Media Int.*, vol. 52, no. 4, pp. 239–252, 2015.
- [38] K. A. Douglas, H. E. Merzdorf, N. M. Hicks, M. I. Sarfraz, and P. Bermel, "Challenges to assessing motivation in MOOC learners: An application of an argument-based approach," *Comput. Educ.*, vol. 150, 2020, Art. no. 103829.
- [39] H. M. Dai, T. Teo, N. A. Rappa, and F. Huang, "Explaining chinese university students' continuance learning intention in the MOOC setting: A modified expectation confirmation model perspective," *Comput. Educ.*, vol. 150, 2020, Art. no. 103850.
- [40] C. Hart, "Factors associated with student persistence in an online program of study: A review of the literature," *J. Interactive Online Learn.*, vol. 11, pp. 19–42, 2012.
- [41] L. Parte and L. Mellado, "Motivational Emails in distance university," *J. Educators Online*, vol. 18, no. 3, 2021, Art. no. n3.
- [42] B. C. P. Rodriguez and A. Armellini, "Developing self-efficacy through a massive open online course on study skills," *Open Praxis*, vol. 9, no. 3, pp. 335–343, 2017.
- [43] L. Rai and D. Chunrao, "Influencing factors of success and failure in MOOC and general analysis of learner behavior," *Int. J. Inf. Educ. Technol.*, vol. 6, no. 4, 2016, Art. no. 262.

- [44] S. Zheng, M. B. Rosson, P. C. Shih, and J. M. Carroll, "Understanding student motivation, behaviors and perceptions in MOOCs," in *Proc. 18th ACM Conf. Comput. Supported Cooperative Work Social Comput.*, 2015, pp. 1882–1895.
- [45] I. Bouchrika, N. Harrati, V. Wanick, and G. Wills, "Exploring the impact of gamification on student engagement and involvement with e-learning systems," *Interactive Learn. Environ.*, vol. 29, no. 8, pp. 1244–1257, 2021.
- [46] M. Aparicio, T. Oliveira, F. Bacao, and M. Painho, "Gamification: A key determinant of massive open online course (MOOC) success," *Inf. Manage.*, vol. 56, no. 1, pp. 39–54, 2019.
- [47] D. Zwillinger, *CRC Standard Mathematical Tables and Formulas*. London, U.K.: Chapman and hall/CRC, 2018.
- [48] J. Tus et al., "The learners' study habits and its relation on their academic performance," *Int. J. Res. Writings*, vol. 2, no. 6, pp. 1–19, 2020.
- [49] Y. Shi, Z. Peng, and H. Wang, "Modeling student learning styles in MOOCs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, Art. no. 979–988.
- [50] V. R. Joseph, "Optimal ratio for data splitting," *Statist. Anal. Data Mining: ASA Data Sci. J.*, vol. 15, no. 4, pp. 531–538, 2022.



Jin Li (Member, IEEE) received the PhD degree from Xi'an Jiaotong University, in 2019. He studied with the University of East Anglia, in the U.K. in 2018, as a visiting PhD student. He worked with Tencent (Shanghai) Ltd. from 2019 to 2021. He worked with Durham University, in 2021, as a post-doctoral research associate. He currently works at Shaanxi Normal University as a lecturer. He has published more than 10 academic papers in the top conferences and journals, including IEEE CVPR, IJCAI, *IEEE Transactions on Image Processing*, and *IEEE Transactions on Multimedia*. His research interest contains data retrieval, Zero-shot Learning, and intelligent education.



Shu Li received the BE degree from the School of Computer and Information Engineering, Henan Normal University, Xinxiang, China, in 2020, where she is currently working toward the master's degree in computer technology from the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her current research interests include online dropout prediction and personalized course recommendations based on educational data.



Yuan Zhao received the BE degree from the School of Computer and Information Engineering, Henan Normal University, Xinxiang, China, in 2020, where he is currently working toward the master's degree in the College of Computer Science and Engineering, Northwest Normal University, Lanzhou, China. He current research interests include neural networks and bioinformatics.



Longjiang Guo (Member, IEEE) received the PhD degree from the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. He was a visiting scholar with Georgia State University, Atlanta, GA, USA, in 2009 and 2013. He was awarded the title of new century outstanding talent by the Ministry of Education, China. He is currently a professor with the School of Computer Science, Shaanxi Normal University, Xi'an, China. He has published more than 70 papers such as IEEE INFOCOM, IEEE ICDCS, IEEE GLOBECOM, IEEE IPCCC, *IEEE Transactions on Computational Social Systems (TCSS)*, *IEEE Transactions on Network Science and Engineering (TNSE)*, *Knowledge-based Systems (KBS)*, as well as Expert Systems with Applications (ESWA). In addition, he won the Second Prize in the National Science and Technology Progress Award, China, in 2005 and the First Prize in the Heilongjiang Province Science and Technology Progress Award, Heilongjiang Province, China, in 2000. His current research interests include AI Education, Learning Technologies, Ubiquitous and Mobile Computing, Edge Computing in Education, and Big Data Processing in Education. He is also a member ACM and CCF.



Fei Hao received the PhD degree in computer science and engineering from Soonchunhyang University, South Korea, in 2016. Since 2016, he has been with Shaanxi Normal University, Xi'an, China, where he is an associate professor. From 2020 to 2022, he was a Marie Curie fellow with the University of Exeter, Exeter, United Kingdom. His research interests include social computing, soft computing, Big Data analytics, pervasive computing, and data mining. He holds a world-class research track record of publication in the top international journals and prestigious conferences. He has published more than 150 papers in the leading international journals and conference proceedings, such as *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, *IEEE Transactions on Services Computing (TSC)*, *IEEE Transactions on Network Science and Engineering (TNSE)*, *IEEE Communications Magazine*, *IEEE Internet Computing*, *ACM Transactions on Multimedia Computing, Communications and Applications* as well as *ACM SIGIR*, *IEEE GLOBECOM*. In addition, he was the recipient of the Best Paper Award from IEEE GreenCom 2013. He was also the recipient of the Outstanding Service Award at IEEE DSS 2018, and IEEE SmartData 2017, the IEEE Outstanding Leadership Award at IEEE CPSCOM 2013, and the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad. He is also a member of ACM, CCF, and KIPS.



Meirui Ren (Member, IEEE) received the PhD degree in the School of Electronic Engineering, Heilongjiang University, Harbin, China, under the guidance of Prof. Jianzhong Li. She is currently an associate professor with the School of Computer Science, Shaanxi Normal University. Her current research interests include computing education, data processing, data mining, and artificial intelligence in education.



Keqin Li (Fellow, IEEE) received the BS degree in computer science from Tsinghua University, in 1985 and the PhD degree in computer science from the University of Houston, in 1990. He is currently a SUNY distinguished professor with the State University of New York and also a National distinguished Professor with Hunan University China. He has authored or co-authored more than 910 journal articles, book chapters, and refereed conference papers. He received several best paper awards from international conferences including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCOM-2022. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top five most influential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for more than twenty consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department with the University of Houston, in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE Technical Committee on Cloud Computing, in 2022. He is an AAAS Fellow, an IEEE Fellow, and an AAIA Fellow. He is a Member of the SUNY Distinguished Academy. He is a Member of Academia Europaea (Academician of the Academy of Europe).